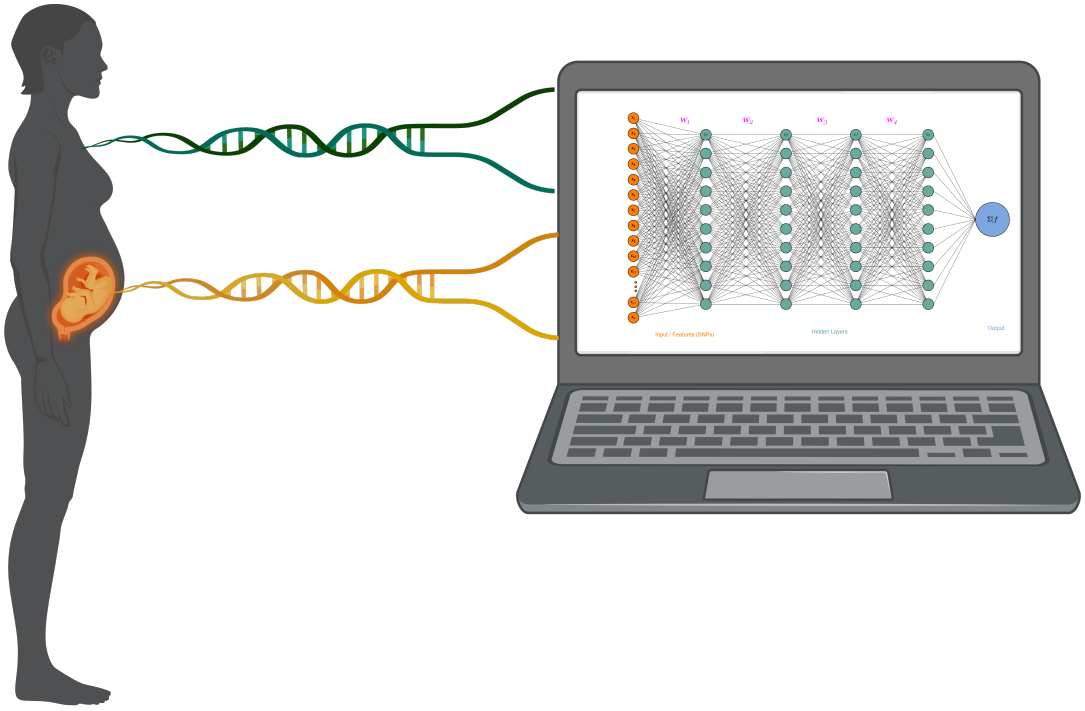
CHALMERS
UNIVERSITY OF TECHNOLOGY



Machine Learning for Genetic Studies

Exploring the Potential of Machine Learning Models for Predicting Preterm Delivery using Genetic Markers

Master's thesis in Biomedical Engineering

HEDVIG SUNDELIN

DEPARTMENT OF ELECTRICAL ENGINEERING
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2023
www.chalmers.se

Master's thesis 2023

Machine Learning for Genetic Studies

Exploring the Potential of Machine Learning Models for Predicting Preterm Delivery using Genetic Markers

Hedvig Sundelin

CHALMERS
UNIVERSITY OF TECHNOLOGY

Department of Electrical Engineering
Division of Biomedical Engineering
Chalmers University of Technology
Gothenburg, Sweden 2023

Machine Learning for Genetic Studies
Exploring the Potential of Machine Learning Models for
Predicting Preterm Delivery using Genetic Markers
Hedvig Sundelin

Supervisor: Julius Juodakis, Department of Obstetrics and Gynecology,
Institute of Clinical Sciences, Sahlgrenska Academy, University of Gothenburg
Examiner: Andreas Fhager, Department of Electrical Engineering,
Chalmers University of Technology

Master's Thesis 2023
Department of Electrical Engineering
Division of Biomedical Engineering
Chalmers University of Technology
SE-412 96 Gothenburg
Telephone +46 31 772 1000

Cover: Pregnant woman and baby with DNA structures stretching over to a computer performing machine learning. Created with BioRender.com

Typeset in LaTeX
Printed by Chalmers Reproservice
Gothenburg, Sweden 2023

Machine Learning for Genetic Studies
Hedvig Sundelin
Department of Electrical Engineering
Chalmers University of Technology

Abstract

Preterm delivery (PTD) is a significant contributor to infant mortality and morbidity worldwide, influenced by environmental and genetic factors. Although previous studies have identified genetic variants associated with PTD and gestational duration, their effect sizes remain relatively small, leaving a substantial portion of the hereditary variation unexplained. This thesis explores the potential of machine learning (ML) techniques to uncover additional insights into PTD and gestational duration using genetic data.

The background section underscores the global impact of preterm birth on child mortality and long-term health outcomes, emphasising the role of genetics with an estimated heritability of around 30%. This project aims to apply ML techniques to improve the prediction of gestational duration and PTD based on genetic data. Research questions address ML model selection, the impact of variables on prediction performance, and a comparison to previous studies. The study is based on the Norwegian Mother, Father and Child Cohort Study (MoBa) and uses data from the Medical Birth Registry of Norway (MBRN). The scope includes the use of genetic data and a focus on the 23 loci previously identified in a related study.

The theory chapter provides an overview of genetics and its application in studying complex conditions like preterm delivery. It also introduces ML and explains the theoretical foundations of different ML models. Subsequently, the methods and materials chapter describes the data acquisition process, preprocessing steps, employed ML classifiers, and model evaluation methods. The chapter highlights the use of neural networks, classic ML algorithms, and libraries for implementation.

Results reveal varying AUC scores among classic models, with logistic regression (LR) performing the best. The choice of variables had a significant impact, with the maternal genome and the Top 23 set, offering the best conditions. Network models achieved comparative scores for binary classification. Additional analyses on the predicted probabilities demonstrated higher AUC scores compared to binary classifications, identifying RMSprop as the best-performing network model. The study reveals a slight improvement in results compared to Polygenic Risk Scores (PRS) but a modest predictive ability overall. The findings in this study suggest that more extensive research is needed to unveil the potential of ML models in improving predictions based on genetic data.

Keywords: MoBa, MBRN, machine learning, SNP, classification, preterm delivery, gestational duration, genetics, prediction, neural network

Acknowledgements

We thank the Norwegian Institute of Public Health (NIPH) for generating high-quality genomic data. This research is part of the HARVEST collaboration, supported by the Research Council of Norway (#229624). We also thank the NORMENT Centre for providing genotype data, funded by the Research Council of Norway (#223273), South East Norway Versjon 6.9 3 Health Authorities and Stiftelsen Kristian Gerhard Jebsen. We further thank the Center for Diabetes Research, the University of Bergen for providing genotype data and performing quality control and imputation of the data funded by the ERC AdG project SELECTionPREDISPOSED, Stiftelsen Kristian Gerhard Jebsen, Trond Mohn Foundation, the Research Council of Norway, the Novo Nordisk Foundation, the University of Bergen, and the Western Norway Health Authorities. We are grateful to all the participating families in Norway who take part in this on-going cohort study.

We are grateful for the funding from Lilla Barnets-foundation, awarded to Julius Juodakis, which helped support the project. The research activities were also made possible by funding from The Swedish Research Council, Stockholm, Sweden (2019-01004), The Research Council of Norway, Oslo, Norway (FRIMEDBIO # 547711), and the Agreement concerning research and education of doctors (ALFGBG-965353).

Finally, I would like to thank everyone who has contributed to this work. A special thanks to the team at Perinatal Laboratory, who made this possible: My supervisor, Julius Juodakis, whose expertise and guidance have been invaluable. My colleagues, Pol Solé Navais and Karin Ytterberg, for their valuable insights and support. Our research leader, Bo Jacobsson, for his leadership and for creating an inspiring and inclusive research environment. Our administrative team, Ylva Folkesson and Kristina Karlsson, for making everything run smoothly.

I am fortunate to have such a remarkable team of individuals around me, and I am truly grateful for their contributions and the positive impact they have had on my professional development.

Hedvig Sundelin, Gothenburg, June 2023

Contents

Acronyms

APGAR Appearance Pulse Grimace Activity Respiration
AUC Area Under the Curve
BNB `BernoulliNB`
DNA deoxyribonucleic acid
DTC `DecisionTreeClassifier`
EI Expected Improvement
FPR False Positive Rate
GWAS Genome-wide association study
IVF In Vitro Fertilisation
k-NC `KNeighborsClassifier`
LDA `LinearDiscriminantAnalysis`
LR `LogisticRegression`
MBRN Medical Birth Registry of Norway
ML Machine Learning
MoBa Norwegian Mother, Father and Child Cohort Study
OAS Oracle Approximating Shrinkage
PRS Polygenic risk score
PTD Preterm delivery
ReLU Rectified Linear Unit
RFC `RandomForestClassifier`
RMSprop Root Mean Square propagation
ROC Receiver Operating Characteristic
SGD Stochastic Gradient Descent
SNP Single nucleotide polymorphism
SNV Single nucleotide variant
SVM Support Vector Machine
TP True Positive
TPR True Positive Rate
UCB Upper Confidence Bound

Abbreviations

AdaDelta Adaptive Delta
AdaGrad Adaptive Gradient
Adam Adaptive moment estimation
AdaMax Adam with infinity norm
AdamW Adam with decoupled Weight decay
sklearn `Scikit-learn`

Special terms

Adaptive Gradient
: abbreviated AdaGrad, an optimisation algorithm that adapts the learning rate for each weight based on the historical sum of squared gradients [1]. A common issue with AdaGrad is diminishing learning rates.

allele
: An allele is one of two or more versions of DNA sequence (a single base or a segment of bases) at a given genomic location. An individual inherits two alleles, one from each parent, for any given genomic location where such variation exists. If the two alleles are the same, the individual is homozygous for that allele. If the alleles are different, the individual is heterozygous[1].

APGAR
: Appearance Pulse Grimace Activity Respiration (APGAR), is an evaluation score (0-10), used by doctors to asses the health of a newborn after 1, 5 and 10 minutes. Higher scores are associated with a healthier baby.

chromosome
: Chromosomes are threadlike structures made of protein and a single molecule of DNA that serve to carry the genomic information from cell to cell. In plants and animals (including humans), chromosomes reside in the nucleus of cells. Humans have 22 pairs of numbered chromosomes (autosomes) and one pair of sex chromosomes (XX or XY), for a total of 46. Each pair contains two chromosomes, one coming from each parent, which means that children inherit half of their chromosomes from their mother and half from their father. Chromosomes can be seen through a microscope when the nucleus dissolves during cell division.[1]

cohort study
: A research study that compares a particular outcome (such as lung cancer) in groups of individuals who are alike in many ways but differ by a certain characteristic (for example, female nurses who smoke compared with those who do not smoke).[2]

complex condition
: A complex disease (or condition), when discussed in the context of genetics, reflects a disorder that results from the contributions of multiple genomic variants and genes in conjunction with significant influences of the physical and social environment. For this reason, complex diseases are also called multifactorial diseases. This stands in contrast to a "simple" genetic disease that is more directly caused by mutations in a single gene. Common examples of complex genetic diseases include heart disease, diabetes, and cancer.[1]

deoxyribonucleic acid
: Deoxyribonucleic acid (abbreviated DNA) is the molecule that carries genetic information for the development and functioning of an organism. DNA is made of two linked strands that wind around each other to resemble a twisted

[1] According to "Talking Glossary of Genomic and Genetic Terms" by leading scientists and professionals at the National Human Genome Research Institute [2]

[2] According to NCI "Dictionary of Cancer Terms" [3].

ladder — a shape known as a double helix. Each strand has a backbone made of alternating sugar (deoxyribose) and phosphate groups. Attached to each sugar is one of four bases: adenine (A), cytosine (C), guanine (G) or thymine (T). The two strands are connected by chemical bonds between the bases: adenine bonds with thymine, and cytosine bonds with guanine. The sequence of the bases along DNA's backbone encodes biological information, such as the instructions for making a protein or RNA molecule.[1]

gene

The gene is considered the basic unit of inheritance. Genes are passed from parents to offspring and contain the information needed to specify physical and biological traits. Most genes code for specific proteins, or segments of proteins, which have differing functions within the body. Humans have approximately 20,000 protein-coding genes.[1]

genetic marker

A sequence of DNA with a known physical location on a chromosome. Genetic markers and genes that are close to each other on a chromosome tend to be inherited together. Genetic markers vary between individuals to the extent that they can be used to help find a nearby gene causing a certain disease or trait within a family. Examples of genetic markers are single polymorphism nucleotides (SNPs), restriction fragment length polymorphisms (RFLPs), variable number of tandem repeats (VNTRs), microsatellites, and copy number variants (CNVs). Genetic markers may or may not have a known function.[3]

genetics

Genetics is the branch of biology concerned with the study of inheritance, including the interplay of genes, DNA variation and their interactions with environmental factors.[1]

genome

The genome is the entire set of DNA instructions found in a cell. In humans, the genome consists of 23 pairs of chromosomes located in the cell's nucleus, as well as a small chromosome in the cell's mitochondria. A genome contains all the information needed for an individual to develop and function.[1]

genome-wide association study

A genome-wide association study (abbreviated GWAS) is a research approach used to identify genomic variants that are statistically associated with a risk for a disease or a particular trait. The method involves surveying the genomes of many people, looking for genomic variants that occur more frequently in those with a specific disease or trait compared to those without the disease or trait. Once such genomic variants are identified, they are typically used to search for nearby variants that contribute directly to the disease or trait.[1]

genomic variation

Genomic variation refers to DNA sequence differences among individuals or populations. Some variants influence biological function (such as a mutation that causes a genetic disease), while others have no biological effects.[1]

[3]According to NCI "Dictionary of Genetics Terms" developed by the PDQ Cancer Genetics Editorial Board [4].

genotype

A genotype is a scoring of the type of variant present at a given location (i.e., a locus) in the genome. It can be represented by symbols. For example, BB, Bb, bb could be used to represent a given variant in a gene. Genotypes can also be represented by the actual DNA sequence at a specific location, such as CC, CT, TT. DNA sequencing and other methods can be used to determine the genotypes at millions of locations in a genome in a single experiment. Some genotypes contribute to an individual's observable traits, called the phenotype.[1]

genotyping

A laboratory process in which an individual's germline DNA is analyzed for specific nucleotides or bases to determine whether certain variants are present. Genotyping differs from sequencing in which all of the nucleotides comprising a specific length of DNA are assessed (e.g., within a gene, exome, or genome).[3]

heritability

The proportion of variation in a population trait that can be attributed to inherited genetic factors. Heritability estimates range from 0 to 1 and are often expressed as a percentage. A number close to 1 may be indicative of a highly heritable trait within a population. It should not be used to estimate risk on an individual basis.[3]

HUNT Cloud

HUNT Cloud was established in 2013 to elevate the collection, accessibility and exploration of large scale information. HUNT Cloud is owned by NTNU and operated by HUNT Research Centre at the Department of Public Health and Nursing at the Faculty of Medicine and Health Sciences.

kilobase

A kilobase (abbreviated kb) is a unit of measurement used to help designate the length of DNA or RNA. One kilobase is equal to 1,000 bases.[1]

locus

A locus, as related to genomics, is a physical site or location within a genome (such as a gene or another DNA segment of interest), somewhat like a street address. The plural of locus is loci.[1]

nucleotide

A nucleotide is the basic building block of nucleic acids (RNA and DNA). A nucleotide consists of a sugar molecule (either ribose in RNA or deoxyribose in DNA) attached to a phosphate group and a nitrogen-containing base. The bases used in DNA are adenine (A), cytosine (C), guanine (G) and thymine (T). In RNA, the base uracil (U) takes the place of thymine. DNA and RNA molecules are polymers made up of long chains of nucleotides.[1]

pedigree

A pedigree, as related to genetics, is a chart that diagrams the inheritance of a trait or health condition through generations of a family. The pedigree particularly shows the relationships among family members and, when the information is available, indicates which individuals have a trait(s) of interest.[1]

phenotype
: Phenotype refers to an individual's observable traits, such as height, eye color and blood type. A person's phenotype is determined by both their genomic makeup (genotype) and environmental factors.[1]

polygenic score
: See Polygenic risk score

polygenic risk score
: A polygenic risk score (abbreviated PRS) uses genomic information alone to assess a person's chances of having or developing a particular medical condition. A person's PRS is a statistical calculation based on the presence or absence of multiple genomic variants, without taking environmental or other factors into account.[1]

single nucleotide variant
: A DNA sequence variation that occurs when a single nucleotide (adenine, thymine, cytosine, or guanine) in the genome sequence is altered. Single nucleotide variants may be rare or common in a population. Sometimes single nucleotide variants are referred to as single nucleotide polymorphisms if they are present in at least 1% of the population. Also called SNV.[3]

single nucleotide polymorphism
: A single nucleotide polymorphism (abbreviated SNP, pronounced snip) is a genomic variant at a single base position in the DNA. Scientists study if and how SNPs in a genome influence health, disease, drug response and other traits.[1]

trait
: A trait, as related to genetics, is a specific characteristic of an individual. Traits can be determined by genes, environmental factors or by a combination of both. Traits can be qualitative (such as eye color) or quantitative (such as height or blood pressure). A given trait is part of an individual's overall phenotype.[1]

variant
: An alteration in the most common DNA nucleotide sequence[3]. The term genomic- or genetic variant can be used to describe e.g. a single nucleotide variant, single nucleotide polymorphism and a mutation [5].

Machine Learning Libraries and Frameworks

BayesianOptimization A constrained global optimization package built upon bayesian inference and gaussian process, that attempts to find the maximum value of an unknown function in as few iterations as possible. This technique is particularly suited for optimization of high cost functions, situations where the balance between exploration and exploitation is important.

Imbalanced-learn Imbalanced-learn (imported as imblearn) is an open source, MIT-licensed library relying on `scikit-learn` and provides tools when dealing

with classification with imbalanced classes.

Keras A high-level application programming interface (API) written in Python with the purpose of enabling fast experimentation. Keras can be run on top of other toolkits such as TensorFlow, Theano, and CNTK [4]

OAS Oracle Approximating Shrinkage (OAS) estimator of the covariance is a formula aimed at choosing a shrinkage coefficient that yields a smaller Mean Squared Error [7]. The OAS estimator of the covariance matrix can be computed on a sample with the oas function of the `sklearn.covariance` package, or it can be otherwise obtained by fitting an OAS object to the same sample.

PLINK PLINK is a free, open-source whole genome association analysis toolset, designed to perform a range of basic, large-scale analyses in a computationally efficient manner.

Python Python is an interpreted, interactive, object-oriented programming language. It incorporates modules, exceptions, dynamic typing, very high level dynamic data types, and classes. It supports multiple programming paradigms beyond object-oriented programming, such as procedural and functional programming [8].

scikit-learn Scikit-learn is an increasingly popular machine learning library. Written in Python, it is designed to be simple and efficient, accessible to non-experts, and reusable in various contexts [5]

TensorFlow TensorFlow: An end-to-end, open-source platform written in Python, C ++ that provides a comprehensive flexible ecosystem of tools and libraries to help developers easily build and deploy machine learning- and deep learning-powered applications [4].

[4] According to Principles and Labs for Deep Learning [6].

[5] According to [9]

List of Figures

List of Tables

1
Introduction

Preterm delivery (PTD) refers to delivery before 37 completed weeks of pregnancy. It is one of the main contributors to infant mortality and morbidity worldwide [10]. Although various environmental factors, such as maternal stress and infection, can contribute to PTD, genetics play a significant role in its occurrence [11]–[14]. Understanding the genetic basis of PTD is essential for developing effective preventive and therapeutic strategies to improve health outcomes for both mothers and babies.

The genetic factors involved in pregnancy duration have not been fully mapped out yet. A recent study identified 25 variants in the mother's genome associated with either PTD, gestational duration or both [15]. However, the effect sizes associated with these variants are relatively small and do not explain the total hereditary proportion of variation in the gestational duration.

This thesis explores the possibility of revealing additional information about PTD and gestational duration using ML on genetic data. Several different methods for classification are trained and tested on individual-level genetic data from the Norwegian Mother, Father and Child Cohort Study (MoBa).

1.1 Background

Every year a large number of children under the age of 5 years die. 2020 this number was estimated to be 5 million [16]. One of the leading causes of these deaths is complications due to being born preterm (before 37 weeks of pregnancy). While most causes of child mortality are rapidly declining, death due to preterm complications only has an estimated reduction rate of 2.1% [1], one of the lowest among the top 17 causes of mortality [10].

The estimated number of PTDs per year is 15 million, and studies show that the global rate is increasing [17]. Additionally, surviving children risk having their long-term health affected in several ways, from respiratory problems to cerebral palsy (CP), bronchopulmonary dysplasia, epilepsy and hyperbilirubinemia [18]–[21]. The risks associated with PTD can be minimised through close monitoring and appropriate medical care but would require better stratification of women at risk.

[1] Annually in 2000–2013.

Despite knowledge of several factors that increase the risk of PTD, including high blood pressure, diabetes, stress, or infections [22], the cause of most PTD remains unknown [23]. However, epidemiological studies using pedigrees have shown that genetic factors play a role, with heritability estimates of around 30% [11]–[13]. While some studies suggest that the maternal genome has the greatest influence on the timing of parturition [24], [25], others propose contributions from fetal genetics that exceed those of the maternal genome [24]. Twin studies and research on heritability suggest that gestational duration has both maternal and fetal genetic components [13]. The most significant known risk factor for PTD is a history of previous PTD, and women who were born preterm or have had sisters who delivered prematurely also have an increased risk of giving birth prematurely [13].

1.1.1 Prior work

Previous work on the genetic effects on gestational duration and PTD has mainly been done using Genome-wide association studies (GWASs). GWAS is a tool for identifying genetic factors contributing to complex conditions, as explained in Section 2.2.

A recent GWAS identified 24 independent variants at 22 loci associated with gestational duration [15]. Apart from this, seven loci associated with PTD were found, six of which were also associated with gestational duration, and one which appeared as only related to PTD, resulting in 23 different loci identified. Only five identified loci have been reported in previous studies, while 18 can be considered new findings.

To explore the potential utility of these findings, polygenic scores for gestational duration and PTD were built using the corresponding GWAS. The polygenic score for the gestational duration accounted for 2.2% of its variance. The effect estimate for the PTD polygenic score was comparable to that of gestational duration, highlighting the genetic similarity between these traits.

Overall, these results suggest that the polygenic scores may be useful in predicting both gestational duration and PTD and shed light on the genetic factors underlying these complex conditions. While the effect sizes of the newly identified variants were limited, ML models could potentially leverage multiple variants to improve predictions further.

A study that utilises ML models for predicting PTD has been conducted on genotyped data from 1 527 mothers of African or Haitian descent [26]. All mothers delivered at the Boston Medical Centre and around 40%, 632, had preterm deliveries.

The study conducted an initial GWAS to determine what genetic variants to use. The variants were divided into six groups, based on different thresholds for the obtained p-values. Thresholds ranged from 5e-3 to 5e-8, resulting in groups containing 4 666 to 3 variants each.

A neural network model, further explained in Section 2.3.8, was created with four hidden layers. Each layer consisted of ten nodes with Rectified Linear Unit (ReLU)-activation functions. One output node is used, while the number of input nodes varies for the different variant groups. Stochastic Gradient Descent (SGD) is used as the optimiser algorithm for training the model on 80% of the data, through back-propagation. The remaining 20% is split equally into a validation set and a testing set. The model is trained for 100 epochs, with early stopping, and performance is evaluated by Sensitivity, Specificity, Gini, AUC, LogLoss and MSE.

In addition to the neural network, a few classic models were tested to provide a comparative measure. The presented results, Table 1.1 suggest that the group containing 4 666 variants can provide an (almost) certain prediction when used to train the network model. While the classic models show slightly lower scores, the results are, in general, incredibly high and revolutionary if reproducible on a more extensive set of data.

Table 1.1: Metrics reported in a previous study on predicting Preterm Delivery (PTD) with different Machine Learning (ML) models [26].

Model	Sens	Spec	Gini	LogLoss	AUC	MSE
Neural Net	1.0	0.99	0.9996	0.0960	0.9998	0.0128
SVM	0.9761	0.9720	NA	NA	0.9741	NA
RF	0.9944	0.4603	NA	NA	0.7274	NA
LDA	0.8826	0.8174	NA	NA	0.8501	NA

1.2 Aim and Purpose

This project aims to apply ML techniques to investigate the genetics of preterm birth. It is possible that intricate interactions, including conflicts between the maternal- and fetal genomes, contribute to pregnancy outcomes and may account for the relatively small genetic effects observed thus far. By employing several different ML models, an attempt to find and quantify the impact of these interactions will be made.

Additionally, which selection of variables offers the most significant improvement in prediction accuracy will be evaluated. The outcome of this research will provide insights for future genetic and experimental investigations of preterm birth and enhance our knowledge of pregnancy control in humans.

Research questions

1. Can ML models improve the ability to predict gestational duration and PTD based on genetic data?
    - What ML model(s) provides the best predictability?
    - How do the results compare to previous studies on the genetic effects on gestational duration and PTD?
2. What impact do the different variable choices have on prediction performance?
    - How does the number of SNPs affect the predictability?
    - What impact does using different genomes (maternal, fetal, or both) have on the prediction accuracy?
    - How does the selection of training samples affect the prediction performance?

1.3 Scope and limitations

ML is a fast-growing area of research, and the number of possible algorithms, programs, and methods is increasing daily. Not all of these can be evaluated with the limited time frame available. The ones chosen, logistic regression, k-nearest neighbours, support vector machine, decision tree, random forest, naive Bayes, linear discriminant analysis and feed-forward neural network, are well-known and commonly used methods for the specific problem aimed to solve. However, it is still possible that other methods would achieve better results.

This project will use genetic data from the Norwegian Mother and Child Study (MoBa [27]). Data acquisition and verification will hence not be part of this study. Furthermore, the project will primarily focus on the 23 loci identified previously in a related study [15].

The programming language used for constructing ML models will be `Python`. This is an effect of prior knowledge and time restrictions. While directly compiled languages, like C or C++, run faster, code development is typically faster in `Python` [28]. `Python` also has an active community with an extensive list of libraries for different purposes.

1.4 Ethics

When conducting research involving medical data, it is essential to consider several ethical principles and guidelines to ensure that the privacy and confidentiality of the individuals involved are protected and that the research is conducted in a responsible and trustworthy manner. For this project, all analyses will be done according to the principles of the Helsinki Declaration [29]. Some of the key ethical considerations relevant for this project include the following:

- **Informed consent:** MoBa is a population-based pregnancy cohort study conducted by the Norwegian Institute of Public Health [27]. Participants were

recruited from all over Norway from 1999–2008. The women consented to participation in 41% of the pregnancies.

- **Data privacy and security:** Any medical data is stored on secure servers designed for biomedical data purposes. Only authorised users can access the data remotely through strict access control and multi-layer encryption. The servers are provided by Norwegian Services for sensitive data (TSD) [30] and HUNT Cloud at the Norwegian University of Science and Technology [31]. Data protection impact assessment is available in Norwegian at DPIA for MoBa [32].
- **Confidentiality and anonymity:** All data is anonymised [2] by MoBa to assure patient confidentiality.
- **Justice:** Participants are selected to provide a relatively homogeneous data set. Therefore, the participants lack diversity, and results may not apply to people genetically different from those with Norwegian ancestry.
- **Transparency and accountability:** Method and results are provided in their respective chapter of this report. All code is available on GitHub, and any questions regarding the project may be directed towards the author of this paper.
- **Ethical oversight:** The establishment of MoBa and initial data collection was based on a licence from the Norwegian Data Protection Agency and approval from The Regional Committees for Medical and Health Research Ethics. The MoBa cohort is currently regulated by the Norwegian Health Registry Act. The current study was approved by Swedish and Norwegian institutional review boards [3] [4].
- **Beneficiary:** This research aims to increase the understanding of how genetics can influence the risk of giving birth preterm.

[2] *"All the data made available to researchers (internal and external) will be stripped of personal identifiers; that is identification will be possible using the code number, but not directly to the person who has provided that information."*[33]

[3] Etikprövningsmyndigheten decision Dnr 2022-03248-01

[4] Regionale komiteer for medisinsk og helsefaglig forskningsetikk sør-øst project 2015/2425

2
Theory

The following chapter provides an overview of essential concepts and theories relevant to this project. To provide the reader with the necessary background knowledge, the chapter begins with an introduction to genetics and the application of genetic data in investigating complex conditions or diseases, such as PTD. This is followed by a brief introduction to ML and an explanation of the theories behind each model used for classification in this project. Finally, the chapter elucidates the approaches employed for data management and model evaluation, ensuring that the reader has a comprehensive understanding of the methods utilised in this study.

2.1 Genetics

The genome refers to the complete set of genetic material present in an organism, including all the nucleotide sequences of deoxyribonucleic acid (DNA), both coding DNA (genes) and non-coding regions [34]. The part of the genetic information that determines inherited traits is referred to as the genotype. An individual's genotype consists of all genetic variants, such as Single nucleotide variant (SNV), insertions, deletions, and other mutations, that can be found in the genome[35].

An Single nucleotide polymorphism (SNP) is a type of genomic variant that occurs when a single nucleotide (adenine (A), thymine (T), cytosine (C) or guanine (G)) in the genome differs between individuals [35], as illustrated in Fig. 2.1. This single change can result in differences in an organism's physical characteristics, or increase the risk of developing certain diseases, among others.

An allele is a variant of a single nucleotide or gene at a specific locus on a chromosome. For each SNP, there are typically two alleles, one from each parent. Each allele pair can be either identical(homozygous) or different (heterozygous).

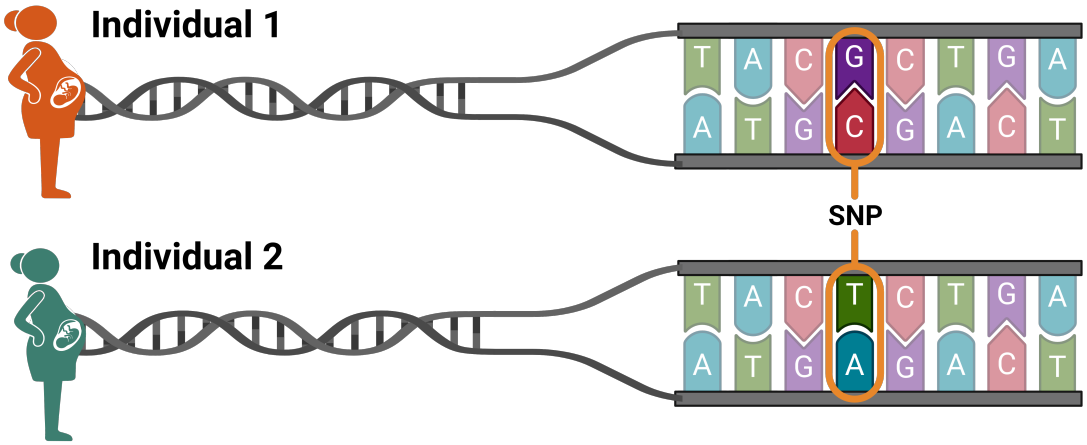


Figure 2.1: Illustration of a Single Nucleotide Polymorphism (SNP)

2.2 Association Studies

Association studies, including GWASs, are powerful tools in genetics research used to investigate the relationship between genetic variants and complex conditions or diseases [36]. These studies are conducted by comparing the frequency of specific genetic variants in individuals with and without the condition or disease of interest. The associated genetic variants identified through these studies can help identify genetic risk factors for diseases, as well as provide insights into the underlying biological mechanisms of complex conditions.

2.2.1 GWAS

In a GWAS, millions of genetic markers are scanned across the genome to find specific variations that are more common in individuals with the condition or disease of interest compared to those without it [37]. The goal of GWASs is to identify genetic loci or regions associated with a particular condition or disease, which can then be studied to better understand the underlying biological mechanisms.

One limitation is that GWAS typically test genetic variants one-by-one for association with a disease, using an additive model (testing the linear additive effect of alleles), and only variants that pass a certain statistical significance threshold are selected for further analysis. This method can overlook specific combinations of

variants or other more complex associations, i.e. gene-gene interactions, that may be important for disease risk.

Moreover, GWAS usually focus on common genetic variants present in a large proportion of the population. Rare genetic variants, which may have a more substantial effect on disease risk but are present in fewer individuals, are often missed by GWAS.

2.2.2 Polygenic Risk Score

Polygenic risk score (PRS), or polygenic score, is a numerical score derived from a combination of genetic variants associated with a condition or disease [38]. The score can be based on any combination of genetic variants, including those with protective effects or those that contribute to the risk. However, interactions between different variants are generally not modelled when creating a PRS.

The data used to generate PRS comes from large-scale GWASs that compare groups with and without the condition or disease to identify genomic variations, meaning only a relative risk can be explained by the score. PRS allow for comparisons of an individual's risk to that of others with different genetic makeup. However, they do not provide a baseline or time frame for disease progression. For example, two individuals with the same PRS may still have lifetime risks that differ if they are of different ages.

2.3 Machine Learning

ML is a subfield of artificial intelligence that involves the development of algorithms and statistical models that can learn patterns and make predictions from data [39]. In recent years, ML has become increasingly popular and applied to various fields, including healthcare, finance, and natural language processing.

Common ML tasks are classification, clustering and regression [40]. In a classification task, the goal is to predict the class label of a given instance based on its features or attributes. ML algorithms and models can be trained on a labelled dataset consisting of a set of instances with known class labels to learn the underlying patterns in the data and make accurate predictions on new, unseen data.

The models and algorithms discussed in this section employ different approaches to classification, ranging from linear models, such as logistic regression, to non-parametric models, such as K-Nearest Neighbours and Decision Trees, to ensemble models, such as Random Forest. Each algorithm has its strengths and weaknesses, and the choice of algorithm depends on the specific problem and data available.

It is worth noting that the performance of a ML model depends not only on the choice of algorithm but also on the quality and quantity of the data used for training and testing [41]. In addition, hyperparameters, which are set before training and

control the model's behaviour, can significantly impact its performance[42]. Therefore, hyperparameter tuning and resampling techniques, such as those discussed in Section 2.4, are often used to improve the performance of ML models.

2.3.1 Logistic Regression

Logistic regression is a statistical method used for binary classification [40]. It uses an equation that models the relationship between a set of independent variables and a dependent binary variable. The equation output (purple dot in Fig. 2.2) is a probability that the binary outcome is true. The probabilities are then transformed into estimated classes using a threshold value, usually 0.5. If the probability exceeds the threshold, the outcome is predicted to be true. Otherwise, the outcome is predicted to be false. The logistic regression equation is fitted during the training process so that the predicted probabilities are as close as possible to the actual binary outcomes in the training data. The fitted equation can then be used to make predictions on new, unseen data.

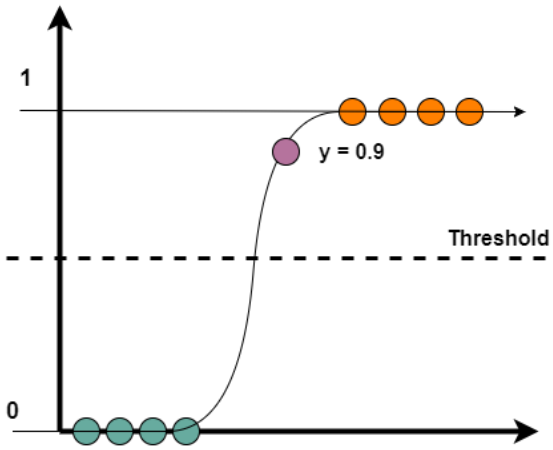


Figure 2.2: Logistic regression

2.3.2 k-Nearest Neighbours

The k-Nearest Neighbours algorithm is an instance-based method used for classification and regression. In classification, the algorithm is used to predict the class of a new data point (purple dot in Fig. 2.3) based on the most common class label among its k-nearest neighbours (marked in blue in Fig. 2.3). The choice of k, the number of nearest neighbours to consider, is an important parameter that affects the performance of the algorithm. Larger values of k tend to smooth out the prediction, while smaller values of k can lead to overfitting and make the prediction more sensitive to individual data points.

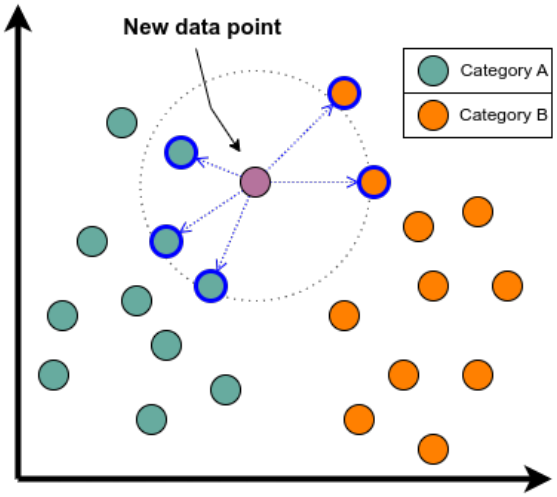


Figure 2.3: k-Nearest neighbour

2.3.3 Support Vector Machines

Support Vector Machine (SVM) is a linear classifier that separates data into different classes by finding the best hyperplane in the feature space[40]. The hyperplane (blue line in Fig. 2.4) is chosen to maximise the margin between the closest data points of different classes, called support vectors.

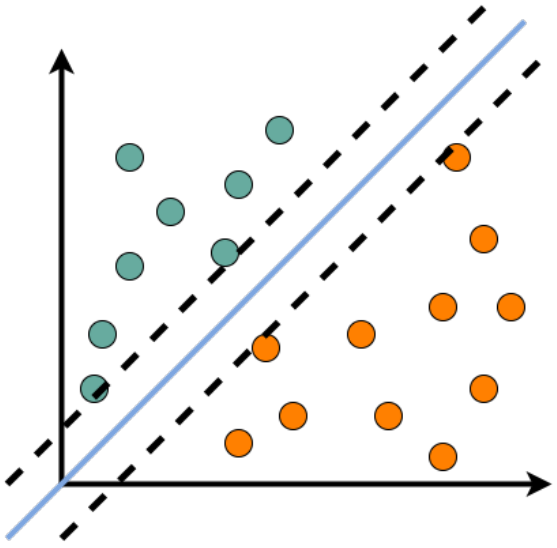


Figure 2.4: Support Vector Machine

2.3.4 Decision Tree

Decision tree classifiers, Fig. 2.5, work by recursively partitioning the data into smaller subsets based on the values of input features, ultimately leading to a set of decision rules or predictions for new data [40]. Each tree node represents a decision point based on a feature, the branches represent the options, and each leaf node

represents a class label. The goal is to create a tree that accurately predicts the target variable while minimising complexity and overfitting.

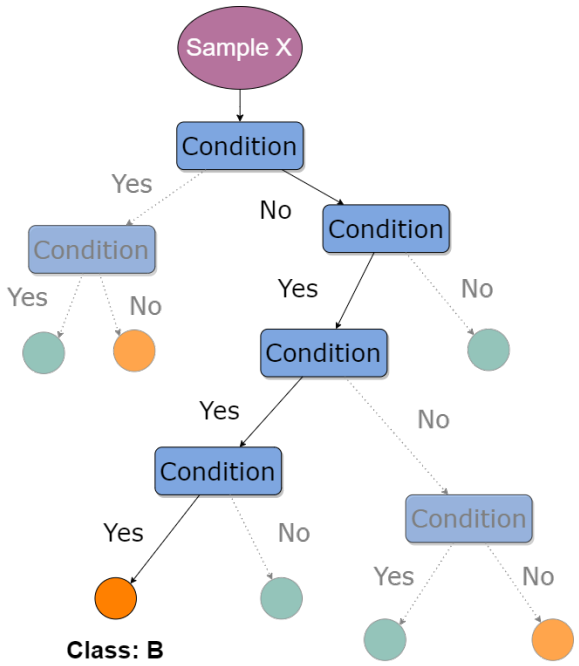


Figure 2.5: Decision Tree

2.3.5 Random Forest

Random Forest, Fig. 2.6, is an ensemble learning method for classification and regression that combines multiple decision trees to make a final prediction. In Random Forest, each tree is trained independently on a random subset of the training data, using a random subset of the features at each split. The final prediction is made by averaging the predictions of all trees, either by a majority vote for classification or by taking the mean for regression.

Random Forest is a flexible and robust method that can handle non-linear relationships in the data, missing values, and irrelevant features. Combining multiple trees also reduces the variance and overfitting of individual trees, leading to improved generalisation performance. However, Random Forests can be slow to make predictions and memory-intensive for large data sets. The number of trees and the size of the random subsets of the data and features are hyperparameters that can be tuned to achieve good performance on the given data.

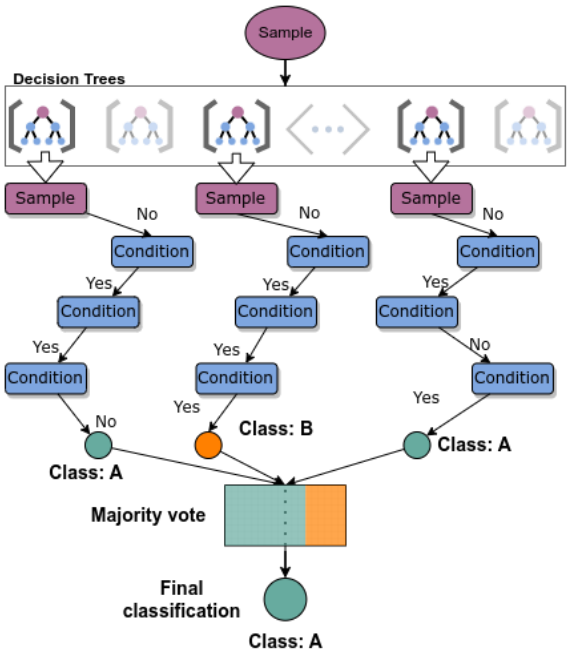


Figure 2.6: Random Forest Classifier

2.3.6 Naive Bayes

Naive Bayes is a probabilistic classifier based on Bayes' theorem, which states that the probability of a class given a set of features is proportional to the probability of the features given the class multiplied by the prior probability of the class. Bayes' theorem:

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

$$\begin{cases} \text{P(A|B)} = & \text{Probability that A happens given B.} \\ \text{P(B|A)} = & \text{Probability of B, given A.} \\ \text{P(A)} = & \text{Probability of A, regardless of B.} \\ \text{P(B)} = & \text{Probability of B, regardless of A.} \end{cases}$$

In Naive Bayes, the assumption is made that the features are conditionally independent given the class, which is why it is called "naive". Based on this assumption, the joint probability of the features and the class can be factored into the product of the individual probabilities of the features given the class.

There are several variants of Naive Bayes, including Gaussian Naive Bayes, Multinomial Naive Bayes, and Bernoulli Naive Bayes, depending on the distributional assumption and the type of features.

The basic idea of Naive Bayes is to estimate the probabilities of the features given the class using training data and then use Bayes' theorem to compute the posterior probability of each class given a new set of features. The class with the highest posterior probability is then chosen as the prediction.

Naive Bayes is simple, fast, and often works well with high-dimensional data. How-

ever, the assumption of feature independence is often not true in real-world data, which can lead to reduced performance compared to other classifiers.

2.3.7 Linear Discriminant Analysis

Linear discriminant analysis, Fig. 2.7, is a dimensionality reduction and classification technique that finds a linear combination of features that maximises the separation between different classes while minimising the separation within classes [43]. It works by modelling the distribution of the features in each class and calculating the posterior probability of belonging to each class. The method is robust against unbalanced data and can provide useful information about the feature distribution of different classes.

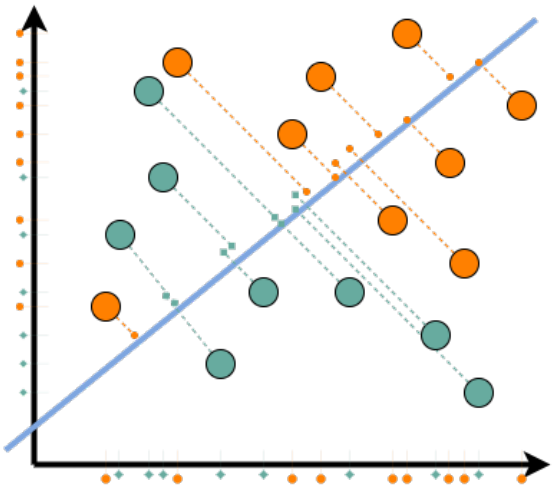


Figure 2.7: Linear Discriminant Analysis

2.3.8 Forward Neural Network

A forward neural network, also known as a feed-forward neural network, is a type of artificial neural network that processes input data in a single direction, from input layer to output layer, without looping back. Forward neural networks are used for both classification and regression tasks, and their architecture can be tailored to the specific problem at hand. They can learn complex non-linear relationships in the data and model complex functions, but can also be prone to over-fitting if the network is too complex for the given data.

A forward neural network consists of an input layer, one or more hidden layers, and an output layer. Each hidden layer consists of interconnected artificial neurons. Each neuron receives input from the previous layer, performs a mathematical operation, and passes its output to the next layer, as visualised in Fig. 2.8. The mathematical operation performed by each neuron is a summation of the product of inputs and weights together with a bias term, typically followed by a non-linear activation function, such as sigmoid (2.1), hyperbolic tangent (2.2), or ReLU (2.3).

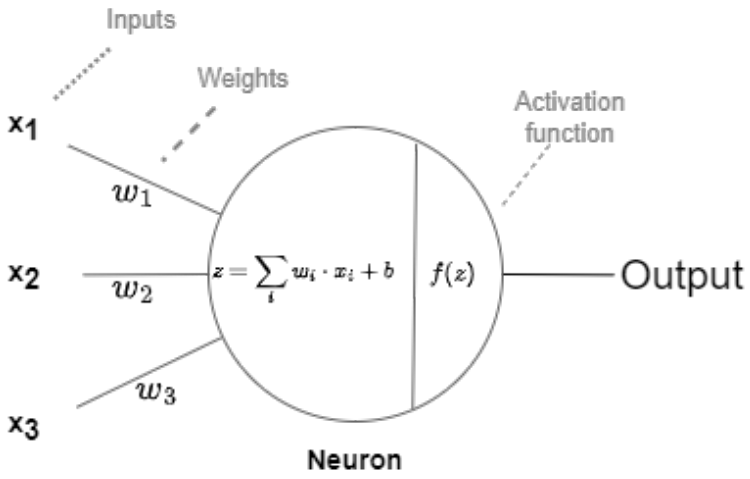


Figure 2.8: A typical setup of the components associated with artificial neurons.

$$f(x) = \frac{1}{1 + e^{-x}} \tag{2.1}$$

$$f(x) = \frac{\sinh x}{\cosh x} = \frac{1 - e^{-2x}}{1 + e^{-2x}} \tag{2.2}$$

$$f(x) = \max(0, x) \tag{2.3}$$

The prediction is determined by the output from the last layer. To determine the performance of the network, this prediction is compared to the true target values by a loss function. There are different loss functions that can be used. For binary classification, the most common loss function is binary cross-entropy (2.4), where y is the observed class and $\hat{y}$ is the network's predicted value.

$$L(\theta) = -\frac{1}{n} \sum_{i=1}^{n} [y_i \log(\hat{y}_i) + (1 - y_i) \log(1 - \hat{y}_i)] \tag{2.4}$$

Training is done by iteratively adjusting the network parameters according to an optimiser. Optimisers are algorithms used to minimise output error. The error, calculated by the loss function, is derived and divided with the derivative of each weight to create a gradient. Part of the gradient is used to update the corresponding weight. The size of this part can range from 0–1 and is called the learning rate. The learning rate, or step size, is what defines how fast the model is changing and can be either fixed or variable according to a learning rate schedule. The choice of learning rate depends highly on the choice of optimisation algorithm. Some commonly used optimisers for feed-forward neural networks are listed in Table 2.1.

Table 2.1: Different optimisers used for feed-forward neural networks.

SGD:	SGD is an iterative optimisation technique that uses randomly selected mini-batches of data to form an expectation of the gradient, rather than the full gradient. Parameters are then adjusted in the direction of the steepest descent. Often employed with momentum. The momentum term accumulates a weighted average of past gradients to determine the direction and magnitude of updates [44]. This helps to smooth out oscillations, accelerate convergence, and improve the ability to escape local optima.
RMSprop:	Root Mean Square propagation (RMSprop) is a variant of SGD that seeks to address the issue of the learning rate not adapting properly to the landscape of the loss function [45]. The basic idea behind RMSprop is to scale the learning rate for each weight based on the estimated variance of the gradient [46]. This is done by keeping a moving average of the squared gradients and dividing the current gradient by the square root of this average. This has the effect of decreasing the learning rate for weights with high variance gradients and increasing it for weights with low variance gradients, allowing the algorithm to take larger steps in flatter directions and smaller steps in steeper directions.
AdaDelta:	AdaDelta is an extension of AdaGrad that addresses the problem of diminishing learning rates by using a rolling average of the past gradients instead of their sum [47]. In AdaDelta, the learning rate is calculated based on the root mean square of the past gradients.
Adam:	Adam is a variant of the SGD algorithm [45]. Adam combines the benefits of momentum and RMSprop. It adapts the learning rate for each weight based on the estimated first and second moments of the gradients.
AdamW:	AdamW is an extension of the Adam optimiser that introduces weight decay directly into the optimisation process [48]. Weight decay is a regularisation technique that encourages the model to have smaller weights by adding a penalty term to the loss function that is proportional to the L2 norm of the weights.
AdaMax:	AdaMax is a variant of the Adam optimiser that replaces the L2 norm of the gradients with the L-infinity norm [45]. Instead of computing the running average of the gradients and its squares as in Adam, AdaMax computes the running average of the gradients and its exponential weighted infinity norm.

2.4 Data management

Two key components of ML, that can greatly impact the performance of a model are resampling and hyperparameters. Resampling is used to balance the distribution of data, while hyperparameters control the behaviour of the model. This section explains the concepts of resampling and hyperparameters, their importance in ML, and techniques for hyperparameter tuning.

2.4.1 Resampling

Resampling is a method used to achieve a more balanced distribution of data. There are several different methods for resampling data, but the main idea is to even out the classes. This is done either by increasing the minority class samples, oversampling, or decreasing the majority class samples, undersampling. Random over- and undersampling are the most basic forms of resampling and are portrayed in Fig. 2.9. Another oversampling method is SMOTE-oversampling, Fig. 2.10, where synthetic minority class samples are created in between the real ones.

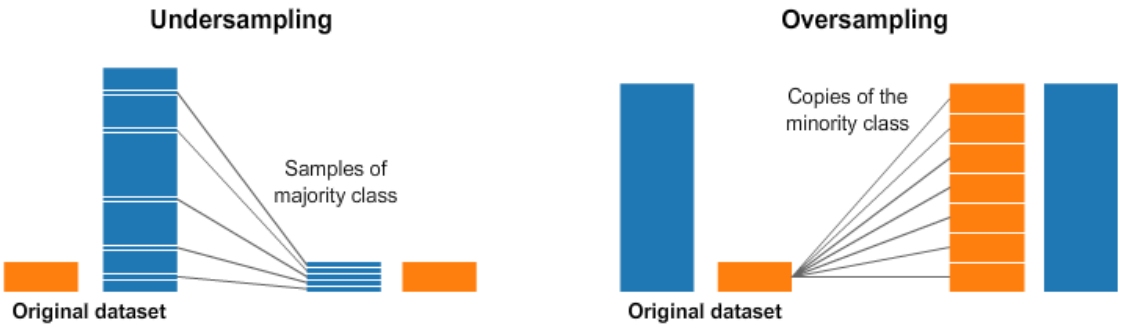


Figure 2.9: Random undersampling and random oversampling [49]

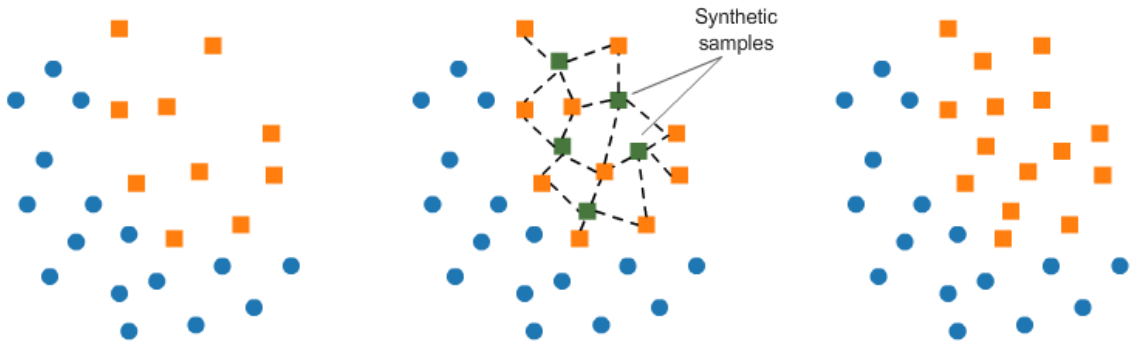


Figure 2.10: SMOTE oversampling [49]

2.4.2 Hyperparameters

Hyperparameters are parameters in ML models that are set before training, as opposed to model parameters, which are learned from the training data. Hyperparameters control the behaviour of the model and the learning process, and include

values such as the learning rate and number of hidden layers in a neural network, the number of trees in a random forest, and the expected distribution in Naive Bayes.

Different hyperparameter values can result in significantly different model behaviour and performance [42]. It is common to perform a hyperparameter tuning process to find the best set of hyperparameters for a given problem. This can involve techniques such as grid search, random search, or Bayesian optimisation, and may involve training multiple models with different hyperparameters and evaluating their performance on a validation set.

Bayesian optimisation involves creating a probability distribution of functions, typically a Gaussian process, to identify the optimal function to be optimised [50]. As more data is collected, the distribution becomes more accurate, enabling the algorithm to determine which areas of parameter space are promising and which are not [51]. This process is repeated, with the algorithm balancing its exploration and exploitation needs based on its understanding of the target function. The known data is used to fit a Gaussian process at each step, and the posterior distribution, in conjunction with an exploration approach such as Upper Confidence Bound (UCB) or Expected Improvement (EI), is used to determine the next point for exploration [50].

Different ML models have different hyperparameters that need to be tuned. In the context of a forward neural network, some key hyperparameters include the batch size, which determines the number of samples used in each iteration, and the choice of optimisation algorithm. Depending on the choice of optimisation algorithm, additional hyperparameters can have an impact on the model performance. Below is an example using the Adam algorithm (2.5) for updating weights[45]. Available hyperparameters here are $\alpha, \epsilon, \beta_1$ and β_2.

$$w_t = w_{t-1} - \alpha \frac{\hat{m}_t}{\sqrt{\hat{\upsilon}_t} + \epsilon} \tag{2.5}$$

Where α refer to the initial step size, ϵ is a small constant which prevents zero-division, while $\hat{m}_t$ and $\hat{\upsilon}_t$ are iteratively updated according to (2.6).

$$\begin{cases} \hat{m}_t = \frac{m_t}{1-\beta_1^t} \\ \hat{\upsilon}_t = \frac{\upsilon_t}{1-\beta_2^t} \end{cases} \tag{2.6}$$

Where m_t is the moving average of the gradient, g_t, according to (2.7) and υ_t is the squared gradient (2.8).

$$m_t = \beta_1 m_{t-1} + g_t(1 - \beta_1) \tag{2.7}$$

$$\upsilon_t = \beta_2 \upsilon_{t-1} + g_t^2(1 - \beta_2) \tag{2.8}$$

β_1 and β_2 controls the exponential decay rate of m_t and υ_t respectively. An additional hyperparameter is offered with the `Keras` implementation, `clipnorm`, which clips each weight gradient to keep the norm below the clipnorm-value.

In summary, hyperparameters are the knobs and dials that control the behaviour of an ML model and can greatly impact its performance, so it is important to tune

them carefully to get the best results for a given problem.

2.5 Evaluation Methods

Evaluation methods are essential for assessing the performance of ML models and for comparing different models. The choice of evaluation method and performance metrics depends on the specific problem and data available. In the following subsections, we will discuss K-fold cross-validation and a series of performance metrics in more detail.

2.5.1 K-fold Cross-validation

K-fold cross-validation is a technique for evaluating and comparing ML models. It is commonly used in the development and tuning of models, as well as in model selection.

The data set is divided into k equally sized "folds", where k is a user-specified number, typically ranging from 5 to 10. The model is then trained k times, each time using a different fold as the validation data set while using the remaining k-1 folds as the training data set [52]. The performance of the model is evaluated by averaging the performance metrics, such as accuracy or F1 score, computed on each of the k validation data sets.

The advantage of k-fold cross-validation is that it provides a more robust estimate of the model performance, compared to a single train-test split. By training the model k times, with each fold used once as the validation set, the technique ensures that the model has seen all the data points in the data set, and provides an estimate of how well the model generalises to new, unseen data.

The choice of k can affect the performance estimate, with higher values of k providing a more robust estimate but also increasing the computational cost of the evaluation [52]. A common choice for k is 10, which provides a good balance between bias and variance in the performance estimate.

2.5.2 Performance Metrics

Performance metrics are used to measure the performance of a ML model. These metrics provide a quantitative assessment of the model's ability to correctly classify instances into their respective classes. It is important to choose the appropriate performance metrics for a given problem, as different metrics may be more relevant depending on the specific task and data [41]. Below are some of the most common metrics for binary classification explained.

Table 2.2: A selection of different performance metrics, often used for binary classification problems.

Accuracy:	Accuracy refers to the ratio of correct predictions to the total number of predictions. It is a simple and intuitive metric but can be misleading if the class distribution is imbalanced. $$Accuracy = \frac{TP + TN}{TP + FP + TN + FN}$$
Specificity:	Specificity, or True Negative Rate, refers to the ratio of true negative predictions to the total number of actual negative examples. It measures the proportion of actual negative examples that are correctly identified. $$Specificity = \frac{TN}{TN + FP}$$
Precision:	Precision refers to the ratio of true positive predictions to the total number of positive predictions. It measures the proportion of positive predictions that are actually true. $$Precision = \frac{TP}{TP + FP}$$
Recall:	Recall, Sensitivity, or True Positive Rate, refers to the ratio between true positive predictions and the total number of actual positive examples. It measures the proportion of actual positive examples that are correctly identified. $$Recall = \frac{TP}{TP + FN}$$
ROC:	Receiver Operating Characteristic (ROC) refers to the combination of Recall or True Positive Rate (TPR) and False Positive Rate (FPR). The two metrics are plotted against each other with the FPR on the x-axis and the TPR on the y-axis. $$TPR = \frac{TP}{TP+FN}, \quad FPR = \frac{FP}{FP+TN}$$
AUC:	Area Under the Curve (AUC) is a metric that summarises the information in the ROC plot to a numeric value. AUC include performance across all possible classification thresholds. It measures the ability of the classifier to distinguish between the positive and negative classes. The score can range between 0–1. however, a score of 0.5 suggests that the classifier can not distinguish one category from the other better than a random guess.

Gini: Gini index is based on AUC but provides a more intuitive score. The score ranges from -1–1, where a perfect model gets 1, a score of 0 suggests that the model can not classify any better than flipping a coin, and a score of -1 suggests that the classifier predicts all positive samples as negative and vice versa.

$$Gini = 2 \cdot AUC - 1$$

3
Methods and Materials

The following chapter presents the approach taken to acquire and pre-process data, the ML classifiers used for prediction, and the evaluation methods employed to assess the performance of the models. The chapter is divided into four sections, with subsections providing details on the various processes involved.

Section 3.1 covers the data acquisition process, which involves gathering genetic data. Section 3.2 focuses on the pre-processing of data, which includes feature reduction, the reading of genetic data, cross-validation, and resampling. Section 3.3 introduces the ML classifiers employed for prediction, including classic models and network models. Finally, Section 3.4 discusses the model evaluation methods used to assess the performance of the ML classifiers. An overview of the different steps in the method is illustrated in Fig. 3.1.

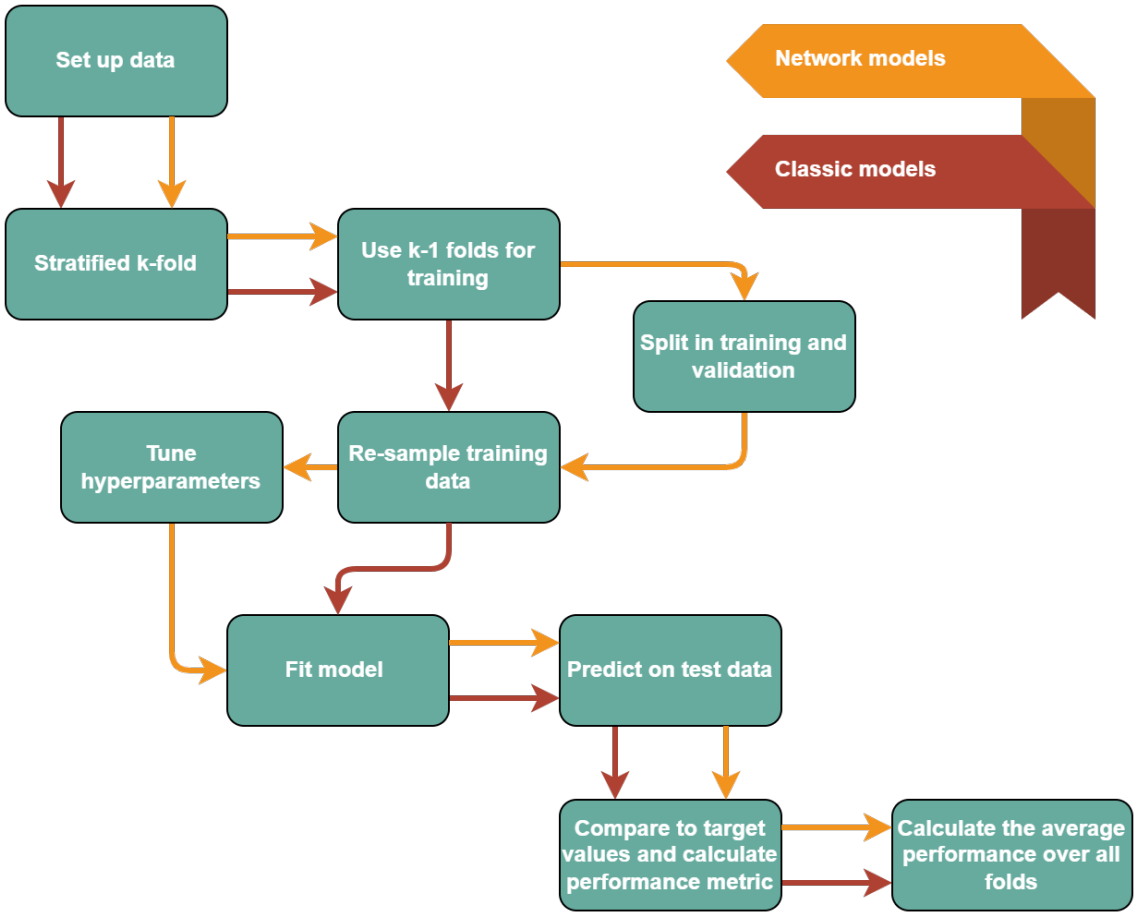


Figure 3.1: Steps for implementing and testing the ML models. Orange arrows show the order of steps for the network models while the red arrows show the path for classic models. 'Set up data' is further explained in Section 3.2, Section 3.4.2 explain the 'Stratified k-fold' step, Section 3.4.1 go through 'Re-sample training data', 'Tune hyperparameters' is covered in Section 3.3.3 and Section 3.4.3 discuss the final steps.

3.1 Data Acquisition

Norwegian Mother, Father and Child Cohort Study (MoBa) is a population-based pregnancy cohort study conducted by the Norwegian Institute of Public Health [27]. Participants were recruited from all over Norway from 1999–2008. The cohort includes approximately 114 500 children, 95 200 mothers, and 75 200 fathers. The data used in this study is version 12 of the quality-assured data files released by MoBa. It includes genotype data, questionnaires filled out by the parents, and linked records from Medical Birth Registry of Norway (MBRN), a national health registry containing information about all births in Norway.

The present study uses a subset of this cohort: pregnancy outcomes, maternal and fetal genotypes from ∼31 000 parent-offspring trios that were genotyped over 2012–2018.

To ensure the quality of the genotyping data and that the study sample was representative of a healthy, non-IVF population, various criteria were used to exclude variants and samples. Genotyping was restricted to singleton, live-birth pregnancies with complete birth registry data, at least one answered MoBa questionnaire, and individuals alive at the time of genotyping. Additionally, pregnancies involving IVF, maternal diabetes, extreme outlier gestational duration, and children with an APGAR score of 0 at both 1 and 5 minutes were excluded. Finally, for mothers with repeated pregnancies in the cohort, one random pregnancy was retained, and genetically non-European populations were excluded. By implementing these exclusion criteria, the study can minimise potential confounding (i.e. when ethnicity affects both genetics and the outcome) and ensure that any associations found between genotypes and outcomes are less likely to be influenced by external factors. These steps are standard in genotyping data acquisition [1].

3.2 Pre-processing of Data

When working with genetic data, it is common to have more features than samples. This makes the data both hard and computationally very heavy to work with. To get around this, the data needs to be filtered. Not all features are relevant for every task. Unfortunately, it is not known in advance which are. To determine which features to include in the model, for classifying whether or not delivery will be preterm, different feature reduction methods can be implemented [54].

3.2.1 Feature reduction

A feature reduction method commonly used in genetic research is GWAS. As explained in Section 2.2, GWAS is a specific kind of association study that involves

[1] Full details specific for this data set can be found in the paper [53].

analysing the entire genome of a large group of individuals to identify genetic variations associated with a particular trait or disease.

Prior to this study, a GWAS was performed, resulting in identifying 23 loci with significant association to PTD or gestational duration [15]. From these regions, two sets of data were constructed. One containing only the 23 top hits in [15], henceforth referred to as the 'Top 23'. One set comprising a subset of 6690 SNPs extracted from regions within 50 kilobase of the top hits and filtered for highly correlated SNPs[2], henceforth referred to as the 'Extended set'.

3.2.2 Reading Genetic Data

The genetic data used in this project is comprised of a phenotype file and a set of genotype files. The genotype files include a list of genetic variants, SNPs, along with information on their genomic location, reference and alternate alleles, a list of individual identifiers, Sentrix ids, and a matrix of numeric values, 0, 1 or 2, corresponding to an individual's genotype at that SNP (either 0, 1 or 2 counts of a particular non-reference allele).

The phenotype file includes a list of pregnancy identifiers, along with information on the individual's phenotypic characteristics, gestational duration in days and whether the birth in question was classified as preterm. In addition to the pregnancy identifiers, "Sentrix ids", used in the genotyping facility as labels on the genotyping devices, are available. These IDs were used to link an individual's trait with their genotypes.

Each pregnancy id is typically linked to one maternal and one fetal sample. However, due to prior filtering of samples, some pregnancy ids were only present in the list of maternal samples and some were only present in the list of fetal samples. Only pregnancies containing both samples were considered, allowing for an accurate comparison of their respective predictability. The presence of both samples was confirmed in 16 301 pregnancies in total, of which 15 882 were controls and 419 were cases.

3.3 Machine Learning Classifiers

In Section 2.3 eight different ML methods are explained. For this project, the models created can be divided into Classic models and Network models. Seven different 'Classic models' were created, each with one of the methods described in Sections 2.3.1 to 2.3.7. Seven different 'Network models' were also created, all with the method 'Forward Neural Network', Section 2.3.8, and one of the optimisers listed in Table 2.1.

[2]Using `PLINK`'s `--indep-pairwise 100 5 0.3` command.

3.3.1 Classic models

All classic models were implemented using the sklearn ML library [55]. A total of seven different models were created, listed in Table 3.1.

Table 3.1: List of implemented classic models, including references to the corresponding section in Chapter 2 and sklearn website.

Method	Sec.	Classification model
Logistic regression	2.3.1	`LogisticRegression` (LR)
k-Nearest neighbours	2.3.2	`KNeighborsClassifier` (k-NC)
Support vector machine	2.3.3	`SVC` (SVC)
Decision tree	2.3.4	`DecisionTreeClassifier` (DTC)
Random forest	2.3.5	`RandomForestClassifier` (RFC)
Naive Bayes	2.3.6	`BernoulliNB` (BNB)
Linear discriminant analysis	2.3.7	`LinearDiscriminantAnalysis` (LDA)

Two hyperparameters were changed for the LDA model. The covariance estimator was selected as `OAS`, as proposed in [56]. Additionally, the solver was exchanged to accommodate the choice of covariance estimator. For the LR model, one hyperparameter was altered. The maximum number of iterations run by the model was increased from 100 to $1e^9$ to allow for convergence. The BNB was selected as the Naive Bayes model since the data is binary and thus have two, mutually exclusive, outcomes [57]. Sklearns default hyperparameters were used without alterations for k-NC, SVC, DTC and RFC. The exact settings are available at the sklearn website for each model, linked in Table 3.1.

3.3.2 Network

The neural network, with the fully-connected setup shown in Fig. 3.2, was created with `TensorFlow` and `Keras`. The sequential model, inspired by the study [26] mentioned in Section 1.1.1, has four hidden layers, each with 10 nodes. Each hidden node is followed by a ReLU activation function (2.3), and the output node is followed by a sigmoid function (2.1). Binary cross-entropy (2.4) was used as the loss function and each optimiser in Table 2.1 was implemented to set up the seven network models.

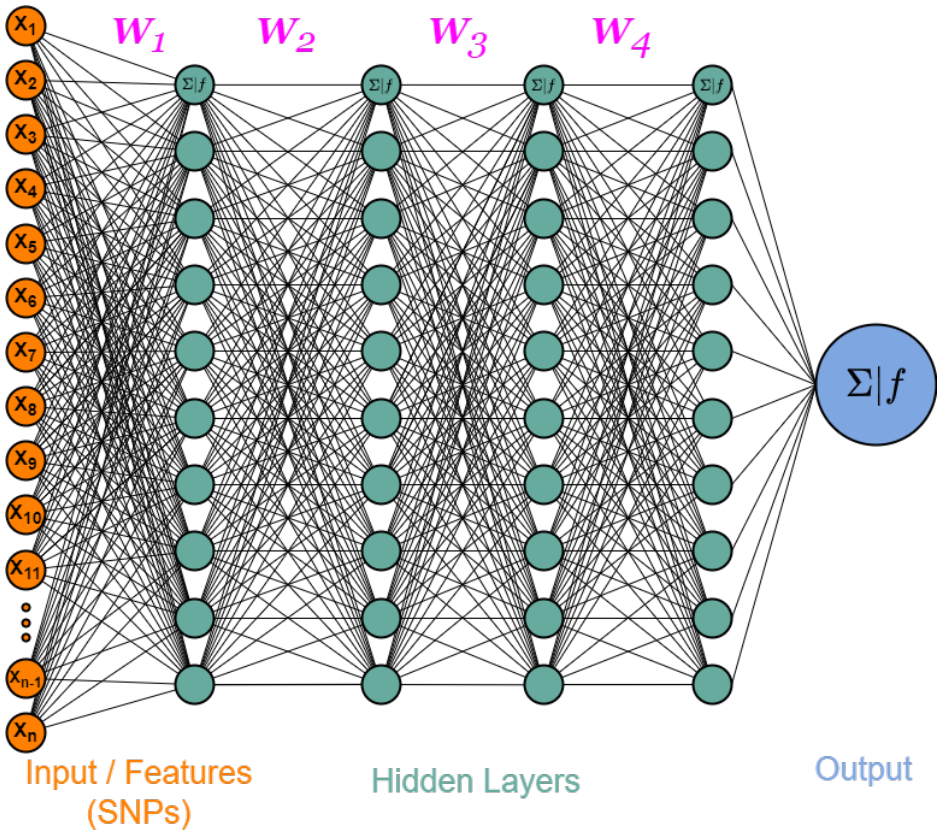


Figure 3.2: Feed Forward Neural Network

To avoid overfitting the model and limit the time needed for training, two different callbacks are implemented. 'Early stopping' will terminate the training if the monitored value does not improve for the number of epochs set in 'Patience'. 'Checkpoint' saves the model from the epoch that provided the best performance according to the monitored metric values. The monitored metric value was selected as the AUC for the validation set.

The saved models are thereafter used to predict the outcome for the samples in the test set. The predicted values are compared to the true target values to provide a set of evaluation metrics for each model.

3.3.3 Hyperparameter Selection

Seven different optimisers, Table 2.1, were tested in this study. Each optimiser has an individual set of hyperparameters that need tuning to suit the problem at hand. The tuning was done by implementing the hyperparameter optimisation tool `BayesianOptimization` [51]. As the name suggests, the tool uses a Bayesian optimisation algorithm, Section 2.4.2, which is run a number of times for each optimiser to provide 2–4 tuned sets of hyperparameter values respectively. Before tuning, boundary values for the hyperparameters of each optimiser were set. The bounds, Table A.2, were selected by taking values from both sides of the default options

provided by `Keras`. Due to time restrictions, hyperparameter tuning was only performed with the oversampled Top 23 SNPs from the maternal genome.

3.4 Evaluation

The unbalanced distribution of cases and controls influences the selection of metrics, appropriate to evaluate the models. Sensitivity and specificity on their own can be very misleading. Predicting that all samples are positive, for example, would yield a perfect sensitivity score but the model would not be very helpful. AUC provides a more balanced score where all classifications, both correct and incorrect ones, are taken into consideration. The AUC is therefore used as the initial evaluation metric for all models.

3.4.1 Resampling

As previously mentioned, the distribution between case and control samples is uneven. Around 2.6% of the samples are cases and 97.4% are controls. Most ML classifiers respond quite poorly to very unbalanced data. Classifying all samples as the majority class would be both easy to learn and actually correct 97% of the time, however, it would not achieve the goals of this work. By resampling the data in a more balanced matter, the model is forced to distinguish between the classes in order to classify the samples correctly 97% of the time.

Three different methods for resampling were tested, random oversampling, random undersampling (Fig. 2.9) and SMOTE (Fig. 2.10). Implementation of the resampling methods was done with `Imbalanced-learn`.

3.4.2 Cross-validation

To provide a robust estimate of each ML model, k-fold cross-validation was used. The selection of k is based on two things, the relatively small number of cases available and the computational cost. To ensure an adequate number of cases in each test set and limit the computational cost, k was set to 5 and a stratified folding was implemented.

To be able to divide the samples into five folds of equal size, one control pregnancy was removed. Each fold contains a test set of 3260 pregnancies with 83–84 cases and 3176–3177 controls. The gestational duration for samples classified as controls span from 259–307 days, with an average of 282 days. For cases, the span goes from 182–258 days with an average of 248 days. Some distributional properties for the test sets in each fold are available in Table A.1.

As visualised in Fig. 3.3, each iteration comprises one fold of test data and four folds of training data. When used for the classic models, all training data with their corresponding true target values, were resampled and used to fit each model. After fitting the models, they were given the test fold and responded with a set of predicted target values. The predicted target values were then compared to the true values and the AUC for that iteration were calculated. After five iterations the average AUC was calculated and used for comparing different models.

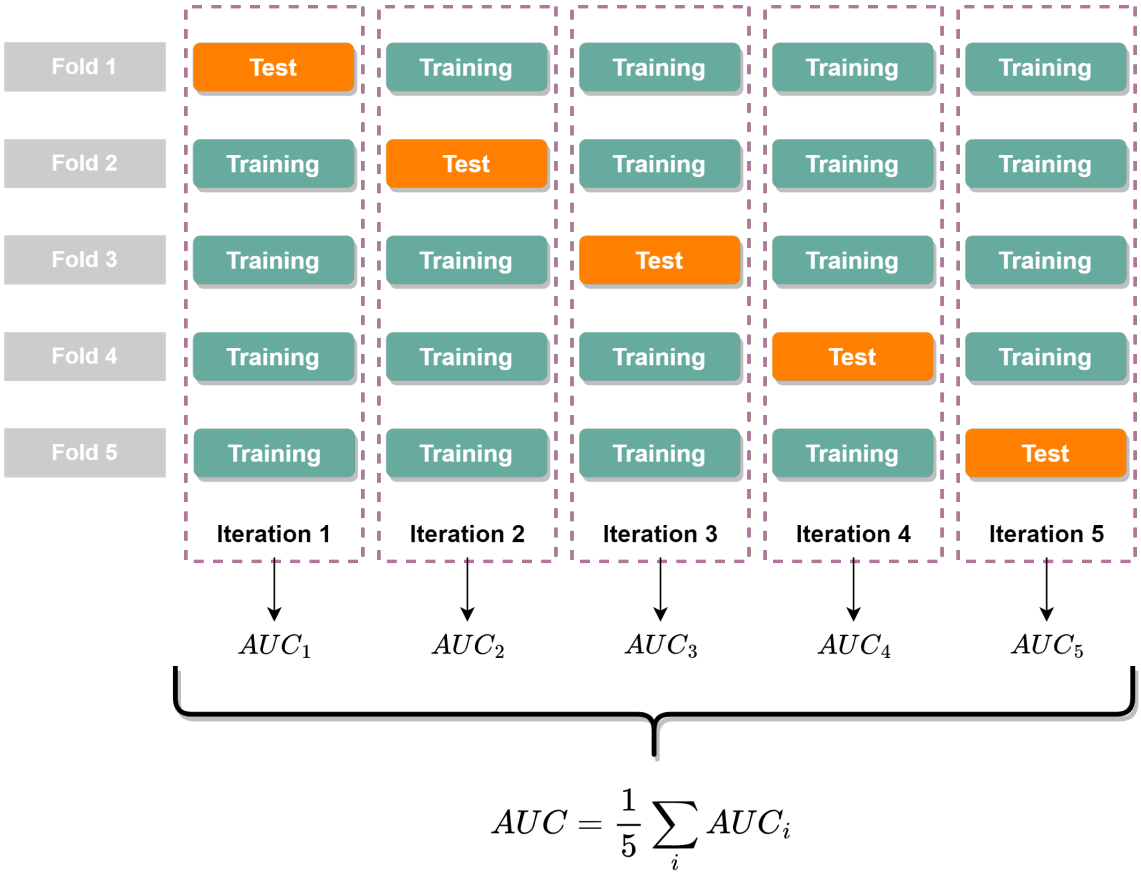


Figure 3.3: K-fold cross-validation with k=5. Each iteration k-1 folds are used for training while the remaining fold is used for testing. An AUC is calculated after each iteration, contributing to the average AUC used to compare different models.

Training the network models requires an additional division of the training data to create a validation set. This set is used during training of the network to keep track of performance progression on unseen data. The validation data was extracted before resampling to keep a distribution similar to that of the test set.

3.4.3 Model Assessment

Classification of samples is, in the models used for this project, based on calculating the probabilities of belonging to class 0 or 1. The class is then either determined as the class with the highest probability score, if probabilities for both classes are

provided, or compared to a threshold value to determine their class if only one probability score is provided. Initially, all models provide the predicted probabilities for each class. When classifying the samples, a threshold is used to determine which class the sample belongs to. Note that any sample on the 'wrong' side of the threshold will be considered an error, regardless of the distance to the threshold. Thus, a model providing the prediction [0.50, 0.51, 0.52] for samples with target values [1, 1, 1] would get perfect performance metrics while a model predicting [0.49, 0.45, 0.48] would get terrible ones, even though their predictions are quite similar. Nonetheless, using the binary classification is faster and provides a good first filter for selecting models to analyse further.

Each model was fit and tested on the different SNP sets and conditions to provide an indicative AUC. The set(s) which provided the highest AUCs for classic models were thereafter used again to calculate metrics from the classic models' predicted probabilities. Likewise, the set(s) that provided the highest AUCs for network models were used to obtain metrics from the network models' predicted probabilities.

As mentioned in Section 1.1.1, a previous study [15] built PRSs to explore the potential utility of their identified SNPs. The score is continuous and reversed with regard to the risk of delivering preterm, meaning negative scores indicate higher risk. A threshold for classifying samples as preterm, based on their PRS, was determined based on the obtained AUCs after iteratively testing different values.

Different models may detect different patterns in the data. Hoping to further improve predictions multiple models were combined. Predicted probabilities from two or more models, either different models or the same model fit on different genomes, were added together and divided by the number of models.

4
Results

The following chapter will present the results obtained while testing the different models described in Chapter 3. Each model was tested on a set of different variables, listed in Table 4.1.

Table 4.1: The variable categories including the different options used for training and testing the ML models.

Feature (SNP) set
`Top 23`
`Extended set`
Resampling method
`Random undersampling`
`Random oversampling`
`SMOTE oversampling`
`No resampling`
Genome
`Maternal`
`Fetal`

4.1 Classic Models

Classic models were fitted and tested on different sets of data. The AUC for these can be seen in Table 4.2. (The AUCs calculated here are based on the binary classifications made by each model.) The highest AUCs were obtained with the use of LR and LDA, both scoring 0.56 when fitted on either the under- or oversampled Top 23-set. Most classic models responded best to the undersampled data and all obtained higher results when run on the maternal genome compared to the fetal genome.

The set of conditions which provided the highest AUC scores was selected for further testing. For the classic models, the selected conditions were: 'Top 23' features, 'Random undersampling' and 'Maternal genome'. Using these variables, additional metrics and a new AUC were calculated on the predicted probabilities from each

Table 4.2: AUC for classic models trained in different conditions. Divided firstly by the number of SNPs, the top 23 hits versus the Extended set. Further divided by method for sampling, random undersampling, random oversampling, SMOTE oversampling or no sampling. Lastly divided by genome, either fetal (F) or maternal (M). The highest AUC(s) for each model are marked in bold. All related plots are available in Appendix B.

	Top 23								Extended Set	
	Under		Over		SMOTE		No		Under	Over
Model	M	F	M	F	M	F	M	F	M	M
LR	**0.56**	0.51	**0.56**	0.53	0.54	0.52	0.50	0.50	0.53	0.51
k-NC	0.49	0.49	0.50	0.50	**0.52**	0.50	0.50	0.50	0.49	0.49
SVC	**0.54**	0.51	0.50	0.50	0.51	0.49	0.50	0.50	**0.54**	0.50
DTC	0.50	0.51	0.51	0.50	0.50	0.49	0.52	0.50	**0.53**	0.51
RFC	**0.54**	0.51	0.50	0.50	0.50	0.50	0.50	0.50	**0.54**	0.50
BNB	0.51	0.51	**0.52**	0.51	0.49	0.51	0.50	0.50	0.51	**0.52**
LDA	**0.56**	0.51	**0.56**	0.53	0.54	0.51	0.50	0.50	0.55	0.50

model. As seen in Table 4.3, the AUCs calculated on predicted probabilities are in general higher than those calculated on binary classifications.

Table 4.3: Additional metrics for the predicted probabilities by classic models, fitted on the maternal, undersampled, Top 23-set. Highest obtained AUC is marked in bold.

Model	AUC	Sens.	Spec.	Gini
LR	**0.58**	0.57	0.58	0.15
k-NC	0.51	0.34	0.67	0.014
SVC	0.56	0.56	0.56	0.11
DTC	0.53	0.52	0.50	0.064
RFC	0.53	0.51	0.52	0.063
BNB	0.53	0.51	0.51	0.067
LDA	0.57	0.57	0.57	0.14

After evaluating the predicted probabilities provided by the classic models, the LR model was still producing the highest AUC. The mean ROC curve for the probabilities predicted by the LR model is presented next to the highest scoring network model in Fig. 4.1.

4.2 Network Models

Before training the network models, the hyperparameters for each optimiser were tuned. The tuning resulted in the values presented in Table A.3. Training and testing were thereafter done on the same data sets as the classic models. The resulting AUCs for the binary classifications made by each network model are available in Table 4.4.

The table shows that SGD2, with an AUC of 0.55, provided the highest score among the network models. The score was obtained with the maternal, oversampled, Top 23-set. Most network models, with the exception of AdamW, responded best to this set and it was hence selected for further analysis on the predicted probabilities. Table 4.5 presents similar metrics for all network models and like the classic models, higher AUC scores compared to the ones obtained with binary classifications.

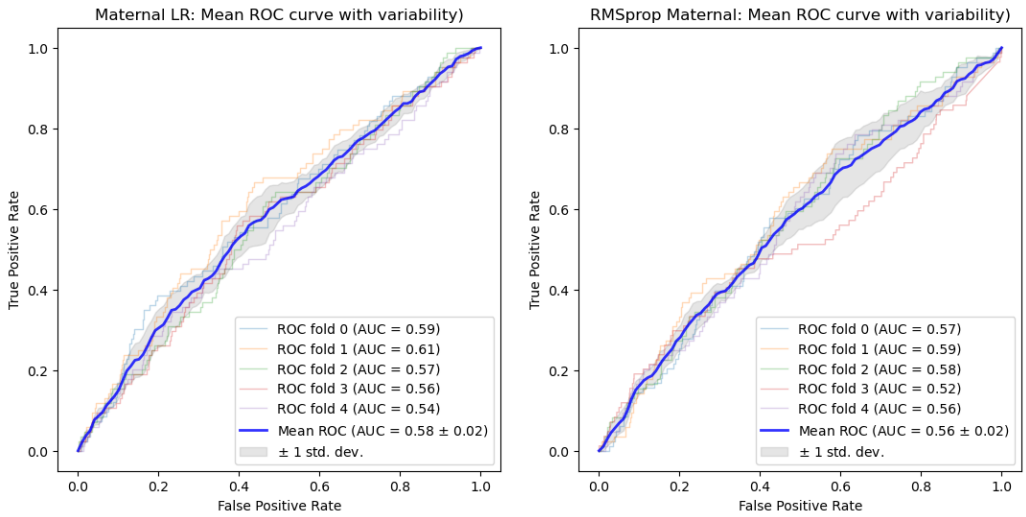
Table 4.4: AUC for network models trained with different optimisers on different feature sets. Divided firstly by the set of SNPs, the Top 23-set versus the Extended set. Further divided by method for sampling, random undersampling, random oversampling, SMOTE oversampling and no sampling. Lastly divided by genome, either maternal (M) or fetal (F). The highest AUC(s) for each model are marked in bold. All related plots are available in Appendix C.

	Top 23								Extended Set	
	Under		Over		SMOTE		No		Under	Over
Model	M	F	M	F	M	F	M	F	M	M
SGD	**0.53**	0.51	**0.53**	0.50	0.52	0.51	0.50	0.50	0.50	0.50
SGD2	0.53	0.51	**0.55**	0.50	0.52	0.48	0.50	0.50	0.53	0.51
AdaMax	**0.53**	0.51	**0.53**	0.52	0.51	0.50	0.50	0.50	0.52	**0.53**
AdaDelta	0.52	0.50	**0.54**	0.49	0.52	0.50	0.50	0.50	0.51	0.53
Adam	0.53	0.51	**0.53**	0.50	0.52	0.51	0.52	0.50	0.51	0.51
AdamW	**0.53**	0.51	0.52	**0.53**	**0.53**	0.50	0.50	0.50	0.51	**0.53**
RMSprop	0.52	0.49	**0.54**	0.50	0.52	0.50	0.50	0.50	0.51	0.50

Table 4.5: Metrics for the predicted probabilities by network models trained on the maternal, oversampled, Top 23-set. Highest obtained AUC(s) are marked in bold.

Model	AUC	Sens.	Spec.	Gini
SGD	0.55	0.55	0.55	0.099
SGD2	**0.56**	0.56	0.56	0.12
AdaMax	**0.56**	0.56	0.56	0.12
AdaDelta	**0.56**	0.56	0.56	0.12
Adam	0.54	0.51	0.56	0.082
AdamW	0.53	0.52	0.52	0.05
RMSprop	**0.56**	0.55	0.57	0.13

While four of the models provided equal AUC scores for predicted probabilities, RMSprop obtained a higher Gini index and was hence considered the best-performing network model. Presented in Fig. 4.1, is thus, the mean ROC curve for the RMSprop model to the right of the corresponding plot for the LR model.



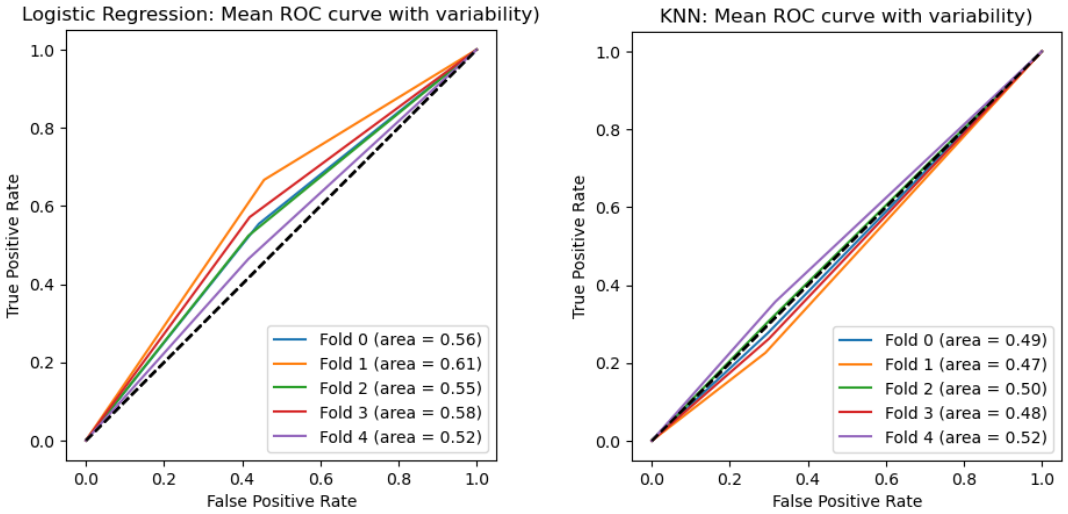
(a) Best classic model: LR **(b)** Best network model: RMSprop

Figure 4.1: Mean ROC for probability predictions by best performing classic- and network model respectively.

4.3 Folds

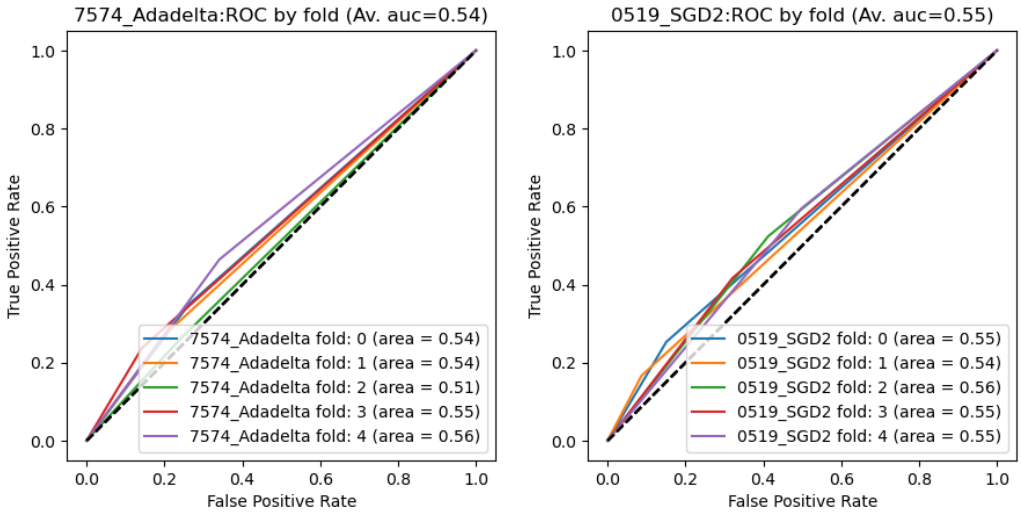
The average results from classic models are quite consistent. They tend to differentiate at most ± 0.01 between different runs. The difference between folds, however, is, in general, larger. Since folds are divided and numbered beforehand, fold 1 always contains the same samples etc. Most classic models perform particularly well on fold one, with AUCs up to 0.61 Fig. 4.2a. This is not true for two models, k-NC, and DTC, which both instead perform particularly badly on this fold (Fig. 4.2b). K-NC and DTC provide their highest AUCs on fold four, which for LDA, BNB, and LR is the fold with the lowest AUCs.

Network models did not show the same pattern as the classic models. AdaDelta for instance, presented in Fig. 4.3, obtained its highest AUC on fold four. Overall, the performance between folds is much more even among the network models. Among the folds providing the best performance, all but fold three are represented and each fold provides the worst performance for at least one model.



(a) AUC by fold for LR **(b)** AUC by fold for k-NC

Figure 4.2: AUC by fold for two different classic models, plots of the remaining models are available in Fig. B.6.

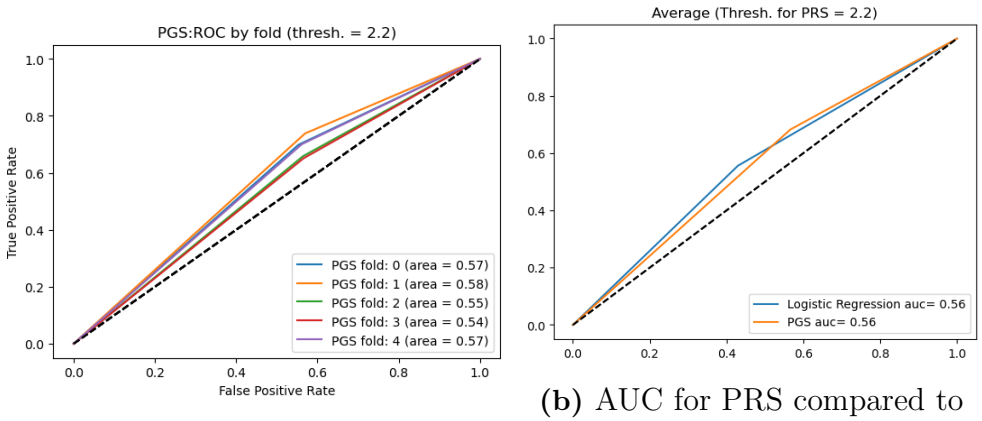


(a) AUC by fold for AdaDelta **(b)** AUC by fold for SGD2

Figure 4.3: AUC by fold for different network models, plots of the remaining models are available in Fig. C.6.

4.4 PRS

Classifying samples with a PRS < 2.2 as preterm provided the highest AUC. This threshold for PRS provided an average AUC equal to the LR model. The comparison, along with the PRS results by fold, are presented in Fig. 4.4.



(a) AUC for PRS by fold

(b) AUC for PRS compared to LR

Figure 4.4: Predictions made with the PRS from [15], by fold and in comparison with the LR model.

4.5 Combined Predictions

As presented in Table 4.6, combining the models for maternal and fetal genomes did not improve predictions overall. Only one model, BNB, obtained a higher AUC with the combined prediction compared to the individual ones.

Combining the predictions of different models, including combining PRS with model predictions, did in some cases improve results, as presented in Table 4.7 and Fig. 4.5.

Table 4.6: AUC for classifications based on combining the predicted probabilities from a model fit on the maternal and fetal genome separately. Trained and tested on the undersampled and oversampled Top 23-set. Corresponding AUC for classifications on the maternal genome alone showed in parenthesis. Models (including the corresponding AUC) that offered an improved AUC by combining predictions are marked in bold.

	Under	Over
Model	C(M)	C(M)
LR	0.55 (0.56)	0.55 (0.56)
k-NC	0.49 (0.49)	0.50 (0.50)
SVC	0.53 (0.54)	0.50 (0.50)
DTC	0.51 (0.51)	0.50 (0.51)
RFC	0.51 (0.54)	0.50 (0.50)
BNB	0.51 (0.51)	**0.53** (0.52)
LDA	0.55 (0.56)	0.55 (0.56)
SGD	0.53 (0.53)	0.52 (0.53)
SGD2	0.52 (0.53)	0.53 (0.55)
AdaMax	0.52 (0.53)	0.52 (0.53)
AdaDelta	0.52 (0.52)	0.51 (0.54)
Adam	0.53 (0.53)	0.51 (0.53)
AdamW	0.52 (0.53)	0.52 (0.52)
RMSprop	0.50 (0.52)	0.53 (0.54)

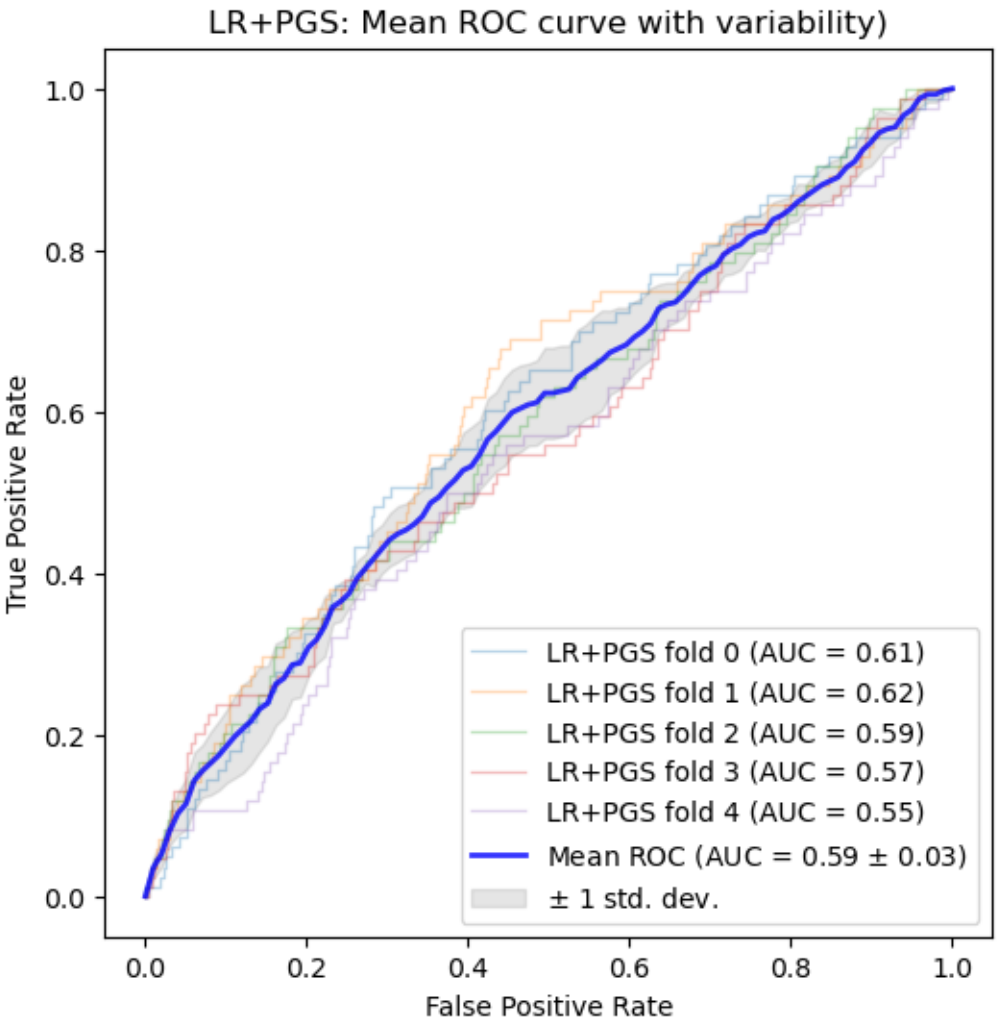


Figure 4.5: Combined probability prediction with PRS and LR

Table 4.7: Metrics for predictions made with the PRS and the combined, PRS + model, predictions with the highest AUC.

Model(s)	AUC	Sens.	Spec.	Gini
PRS	0.56	0.56	0.55	0.13
PRS + LR	0.59	0.58	0.59	0.17
PRS + SGD	0.59	0.59	0.59	0.18
PRS + RMSprop	0.59	0.59	0.59	0.18
PRS + AdaDelta	0.59	0.58	0.59	0.17
PRS + SGD2	0.59	0.59	0.59	0.18
LR + RMSprop	0.58	0.58	0.57	0.15

5
Discussion

This chapter delves into key aspects of the study, starting with variable selection-Section 5.1. This section explores the impact of different variables on prediction performance, including the significance of the maternal genome and the selected Top 23 set. Additionally, model selection is discussed (Section 5.4), highlighting the varying performance of classic ML algorithms and network models. Section 5.2 examines the influence of folds on model performance, showcasing consistency and variations across different folds. Furthermore, the chapter explores the potential of combining models, Section 5.3, to further enhance prediction accuracy. A short review of previous work is included, comparing findings and methodologies with another study in the field, Section 5.4.1. Lastly, the challenges encountered throughout this study and outlined avenues for future work are presented in Section 5.5, suggesting areas for improvement and exploration, such as incorporating additional genetic markers and considering other environmental factors in the prediction models.

5.1 Variable Selection

What impact do the different variable choices have on prediction performance?

In a majority of tests, the Top 23 set provided better results than the Extended set, which uses more SNPs from the same genetic locations. This could mean that no new information is obtained from the additional SNPs. It is, however, also possible that some SNPs hold additional information but that a large number of uninformative SNPs makes it difficult to derive any benefit from these.

Tables 4.2 and 4.4 both show that the selection of genome has a big impact. The maternal genome seems to provide more useful information than the fetal, which is in line with previous research [25], [58]. One instance where the fetal genome provides better results than the maternal is seen for AdamW on the oversampled Top 23 set. The difference is however quite small, and considering the variance in results obtained for the network models, this is most likely not a consequence of the genome choice.

The network models in general respond better to the oversampled data compared to the undersampled. This has two possible explanations. Either information is lost when disregarding several control samples, or this is a result of the hyperparameters being tuned on oversampled data. The classic models have the opposite behaviour

and provide higher scores when comparing the top results from both categories. Information is thus most likely not lost in the undersampling process, even though it disregards a large part of the control samples, suggesting that this group is rather homogeneous.

5.2 Folds

As mentioned earlier, the control samples seem relatively homogeneous and should not be the cause for differences between folds. Table A.1 shows that the difference in average gestational duration for cases does not vary much between the different folds. While the minimum value for the cases in fold one (which provided the highest AUC) is only 182 days (which is the shortest duration), the minimum value in fold four (which provided the lowest AUC) is the second shortest duration with 193 days. Fold zero, for instance, has a minimum value of 219 days, a much larger difference, but still performs almost as well as fold one. Thus, the differences do not seem to be caused by differences in gestational age within the cases; the folds may, however, differ in some other properties of the cases that are not directly observed and recorded in the medical data, but affect the performance.

5.3 Combined Predictions

Since the models work in different ways, they are expected to interpret the data differently and detect different kinds of patterns. This could, in theory, be quite beneficial since models can be combined to produce ensemble predictions.

5.3.1 Maternal and Fetal

No improvement was seen by combining the predictions made on the maternal and fetal genomes. Due to time restrictions, only additive effects could be explored in this study. Including both genomes as input data would allow detection of interactions between different SNPs from the maternal and fetal genomes.

5.3.2 Combining Models

Combining classic models with each other did not provide any higher AUCs than the value provided by LR alone. Unfortunately, the lowest score obtained with the LR model, the score for fold four, is still higher than the score obtained by k-NC and DTC, which had their best results on fold four. Combining either k-NC or DTC with LR does not improve the outcome.

Combining the LR model with the PRSs presented in [15], Fig. 4.5, did slightly improve results. However, the difference in average over the folds is only 0.01, which

lies within the standard deviation.

5.4 Predictive ability

Can ML models improve the ability to predict gestational duration and PTD based on genetic data?

The model that performed best in this study, as seen in Fig. 5.1, was also one of the most simple ones, LR. This model is quite similar to linear regression (in that effects of SNPs are additive in both), the model used for finding the 23 loci selected for the project. Whether this has affected the outcome is hard to say since no comparison to other methods for feature reduction has been possible to conduct during the time of this project. The simplicity of the LR model does, however, contribute to the conclusion that no complex, nonlinear effects with high significance are present within the feature sets used in this study.

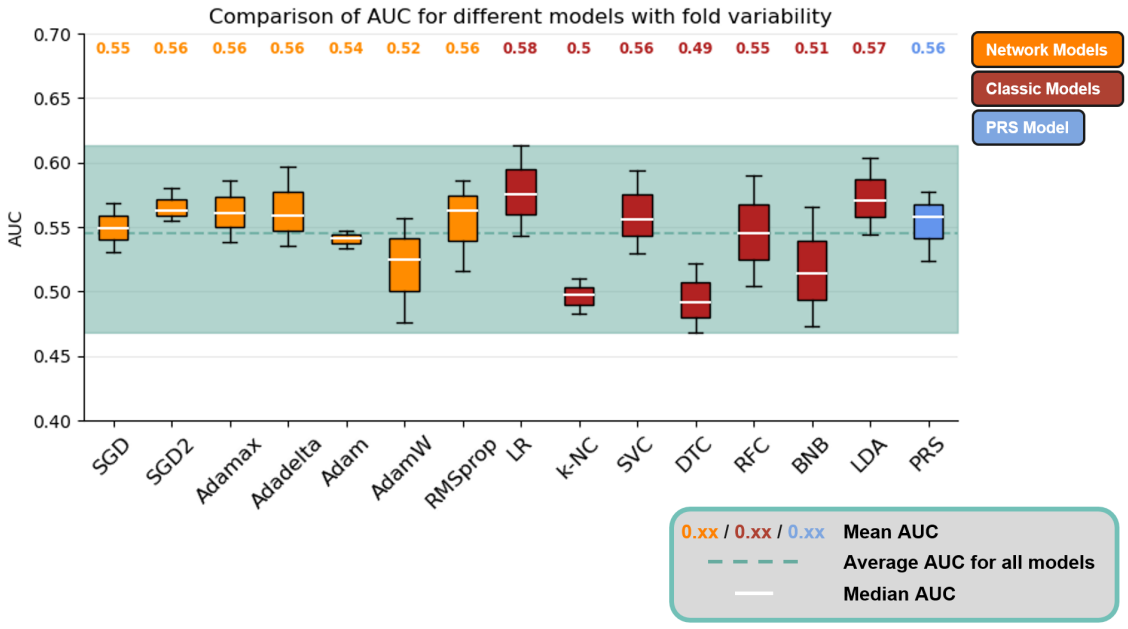


Figure 5.1: Box plot over AUCs calculated on the predicted probabilities for each model. The range for each box represents the variability between different folds.

5.4.1 Previous Work

As mentioned in Section 1.1.1, two previous studies have been examined in this project. Fig. 5.1 shows how the models in this project compare to the previously created PRS. Two models, LR and LDA, provide a higher average AUC. The improvements are, however, not very big.

The difference compared to the previous ML study [26] is significantly larger, as seen in Table 1.1. They present an AUC of 0.9998 based solely on the mothers' genetic data. A literally, unbelievable result. Since PTD is a complex condition it relies on more than genetics. Hence, the base for their predictions does not include everything needed for such distinction.

One possibility for their incredible results could be that there are other genetically distinct differences between cases and controls, for instance, that all cases are of Haitian descent while the controls have African descent.

5.5 Challenges and Future Work

The previously mentioned paper used data based on the p-value of each SNP. While their results are unbelievable, it would be interesting to use their selection criteria on our data to see if that would have any impact on the results.

SNPs located near each other on the genome are often affecting the same gene. If gestational duration or PTD were, genetically, solely driven by the 23 genes found in [15], most information to be found in the genome would be included in the Extended set. However, if we believe that there are more genes involved in this complex condition, including SNPs from additional areas would be an appropriate way to extend this research.

Hyperparameter selection for different optimisers had a large impact on the network models. A challenge has been to determine what conditions to use for the tuning, since different conditions produce different results. Tuning hyperparameters are a quite computationally heavy process and to tune each optimiser for each condition is a very time consuming process and hence not a viable option. To include all options (both general conditions and hyperparameters for the optimiser) in the tuning is not an option either since the working memory required for such operation extends the working memory available on the server. The solution to this issue has been to try a few different conditions when tuning one of the optimisers and simply use the best performing selection for tuning the remaining optimiser. It is hence possible that better results could have been obtained by tuning with different conditions.

The initial hypothesis was that network models would be able to extract more information than the PRS and the classic models. This assumption led to a disproportionate amount of time spent on tuning and testing these models. As a drawback, no tuning was done for the classic models. Knowing the outcome, this time might have been better spent on grid search for the classic models hyperparameter selection. Another time consuming part of this study was the number of models explored. In hindsight, it would have been better to focus on a smaller number of models. However, the limited number of previous studies within the area made the decision of selecting models hard to validate without testing several options.

The defining rule for a sample being considered a case is a gestational duration of less than 259 days. However, an individual delivering after 258 respectively 259 days of gestation will most likely not have distinct genetic difference. One way to handle this difficulty is to make continuous predictions on the gestational duration instead of binary classifications on PTD. While the limited time for this study did not allow for this, it is a strong suggestion for continued research.

6
Conclusion

Multiple factors contribute to the risk of delivering preterm, both genetic and environmental. The associated genetic variants explored in this study only show minor effects individually, and no indications of complex interactions have been revealed.

Furthermore, the maternal genome has shown the best predictability based on the examined SNPs. However, it is possible that interactions between SNPs in the maternal and fetal genomes have been missed since only additive effects were examined.

It is clear that resampling of the training data is needed, but whether it is done by random under or random oversampling seems to be less important.

The areas around each of the 'Top 23' SNPs have not shown any indication of harbouring essential information for PTD prediction. And overall, no significant improvements in predictability have been shown.

Genetic testing does, however, provide some information about an individual's risk. And I believe that further research, including more areas of the genome and maybe focusing on gestational duration instead of PTD, may enhance genetic predictability. Possibly even enough to, in the future, be used as a reliable method for early detection of at-risk pregnancies. A measure that would be beneficial for determining possible preventive interactions and the need for close monitoring.

References

[1] J. Duchi, E. Hazan, and Y. Singer, "Adaptive Subgradient Methods for Online Learning and Stochastic Optimization," *Journal of Machine Learning Research*, vol. 12, no. 61, pp. 2121–2159, 2011. [Online]. Available: http://jmlr.org/papers/v12/duchi11a.html.

[2] *Talking Glossary of Genetic Terms | NHGRI*. [Online]. Available: https://www.genome.gov/genetics-glossary.

[3] *NCI Dictionary of Cancer Terms - NCI*, en, nciAppModulePage, Feb. 2011. [Online]. Available: https://www.cancer.gov/publications/dictionaries/cancer-terms/ (visited on 05/09/2023).

[4] *NCI Dictionary of Genetics Terms - NCI*. [Online]. Available: https://www.cancer.gov/publications/dictionaries/genetics-dictionary.

[5] National Human Research Institute, *Human Genomic Variation*, Feb. 2023.

[6] S.-C. Huang and T.-H. Le, "Chapter 2 - Neural networks," en, in *Principles and Labs for Deep Learning*, S.-C. Huang and T.-H. Le, Eds., Academic Press, Jan. 2021, pp. 27–55, ISBN: 9780323901987. DOI: 10.1016/B978-0-323-90198-7.00006-9. [Online]. Available: https://www.sciencedirect.com/science/article/pii/B9780323901987000069 (visited on 05/11/2023).

[7] Y. Chen, A. Wiesel, Y. C. Eldar, and A. O. Hero III, "Shrinkage Algorithms for MMSE Covariance Estimation," Jul. 2009. DOI: 10.1109/tsp.2010.2053029. [Online]. Available: https://arxiv.org/abs/0907.4698.

[8] *General Python FAQ*. [Online]. Available: https://docs.python.org/3/faq/general.html (visited on 05/21/2023).

[9] L. Buitinck, G. Louppe, M. Blondel, *et al.*, *API design for machine learning software: Experiences from the scikit-learn project*, arXiv:1309.0238 [cs], Sep. 2013. DOI: 10.48550/arXiv.1309.0238. [Online]. Available: http://arxiv.org/abs/1309.0238 (visited on 05/08/2023).

[10] L. Liu, S. Oza, D. Hogan, *et al.*, "Global, regional, and national causes of child mortality in 200013, with projections to inform post-2015 priorities: an updated systematic analysis," *The Lancet*, vol. 385, no. 9966, pp. 430–440, Jan. 2015. DOI: 10.1016/s0140-6736(14)61698-6.

[11] D. Modzelewska, P. Sole-Navais, G. Zhang, L. J. Muglia, S. Nilsson, and B. Jacobsson, "Importance of the environment for gestational duration variability and correlation between relatives results from the Medical Swedish Birth Registry, 1973-2012," *PLOS ONE*, vol. 15, no. 7, N. G. Than, Ed., e0236494, Jul. 2020. DOI: 10.1371/journal.pone.0236494.

[12] B. Clausson, P. Lichtenstein, and S. Cnattingius, "Genetic influence on birthweight and gestational length determined by studies in offspring of twins,"

BJOG: An International Journal of Obstetrics and Gynaecology, vol. 107, no. 3, pp. 375–381, Jan. 2000. DOI: 10.1111/j.1471-0528.2000.tb13234.x.

[13] H. A. Boyd, G. Poulsen, J. Wohlfahrt, J. C. Murray, B. Feenstra, and M. Melbye, "Maternal Contributions to Preterm Delivery," *American Journal of Epidemiology*, vol. 170, no. 11, pp. 1358–1364, Jan. 2009. DOI: 10.1093/aje/kwp324.

[14] M. Wadon, N. Modi, H. S. Wong, A. Thapar, and M. C. O'Donovan, "Recent advances in the genetics of preterm birth," *Annals of Human Genetics*, vol. 84, no. 3, p. 205, May 2020, ISSN: 14691809. DOI: 10.1111/AHG.12373. [Online]. Available: /pmc/articles/PMC7187167/%20/pmc/articles/PMC7187167/?report=abstract%20https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7187167/.

[15] P. Solé-Navais, C. Flatley, V. Steinthorsdottir, *et al.*, "Genetic effects on the timing of parturition and links to fetal birth weight," *Nature Genetics 2023 55:4*, vol. 55, no. 4, pp. 559–567, Apr. 2023, ISSN: 1546-1718. DOI: 10.1038/s41588-023-01343-9. [Online]. Available: https://www.nature.com/articles/s41588-023-01343-9.

[16] *Child mortality (under 5 years)*. [Online]. Available: https://www.who.int/news-room/fact-sheets/detail/levels-and-trends-in-child-under-5-mortality-in-2020.

[17] S. R. Walani, "Global burden of preterm birth," *International Journal of Gynecology and Obstetrics*, vol. 150, no. 1, pp. 31–33, Jan. 2020. DOI: 10.1002/ijgo.13195.

[18] D. H. Adamkin, "Late preterm infants: severe hyperbilirubinemia and postnatal glucose homeostasis," *Journal of Perinatology*, vol. 29, no. S2, S12–S17, Apr. 2009. DOI: 10.1038/jp.2009.41.

[19] B. Thébaud, K. N. Goss, M. Laughon, *et al.*, "Bronchopulmonary dysplasia," *Nature Reviews Disease Primers*, vol. 5, no. 1, Nov. 2019. DOI: 10.1038/s41572-019-0127-7.

[20] C. Crump, K. Sundquist, M. A. Winkleby, and J. Sundquist, "Preterm birth and risk of epilepsy in Swedish adults," *Neurology*, vol. 77, no. 14, pp. 1376–1382, Jan. 2011. DOI: 10.1212/wnl.0b013e318231528f.

[21] H. Trønnes, A. J. Wilcox, R. T. Lie, T. Markestad, and D. Moster, "Risk of cerebral palsy in relation to pregnancy disorders and preterm birth: a national cohort study," *Developmental Medicine & Child Neurology*, vol. 56, no. 8, pp. 779–785, Mar. 2014. DOI: 10.1111/dmcn.12430.

[22] S. Dauengauer-Kirlienė, I. Domarkienė, I. Pilypienė, G. G. Gabrielėžukauskaitė, V. Kučinskas, and A. Matulevičienė, "Causes of preterm birth: Genetic factors in preterm birth and preterm infant phenotypes," 2022. DOI: 10.1111/jog.15516. [Online]. Available: https://obgyn.onlinelibrary.wiley.com/doi/10.1111/jog.15516.

[23] J. P. Vogel, S. Chawanpaiboon, A. B. Moller, K. Watananirun, M. Bonet, and P. Lumbiganon, "The global epidemiology of preterm birth," *Best Practice & Research Clinical Obstetrics & Gynaecology*, vol. 52, pp. 3–12, Oct. 2018, ISSN: 1521-6934. DOI: 10.1016/J.BPOBGYN.2018.04.003.

[24] T. P. York, J. F. Strauss, M. C. Neale, and L. J. Eaves, "Racial Differences in Genetic and Environmental Risk to Preterm Birth," *PLoS ONE*, vol. 5, no. 8, A. Lucia, Ed., e12391, Jan. 2010. DOI: 10.1371/journal.pone.0012391.

[25] G. Zhang, B. Feenstra, J. Bacelis, *et al.*, "Genetic Associations with Gestational Duration and Spontaneous Preterm Birth," *New England Journal of Medicine*, vol. 377, no. 12, pp. 1156–1167, Jan. 2017. DOI: 10.1056/nejmoa1612665.

[26] P. Fergus, C. C. Montañez, B. Abdulaimma, P. Lisboa, C. Chalmers, and B. Pineles, "Utilizing Deep Learning and Genome Wide Association Studies for Epistatic-Driven Preterm Birth Classification in African-American Women," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 17, no. 2, pp. 668–678, 2020. DOI: 10.1109/TCBB.2018.2868667.

[27] P. Magnus, C. Birke, K. Vejrup, *et al.*, "Cohort Profile Update: The Norwegian Mother and Child Cohort Study (MoBa)," *International Journal of Epidemiology*, pp. 382–388, 2016. DOI: 10.1093/ije/dyw029. [Online]. Available: www.illumina.com/products/humancore_exome_beadchip_.

[28] A. Subasi, "Chapter 1 - Introduction," en, in *Practical Machine Learning for Data Analysis Using Python*, A. Subasi, Ed., Academic Press, Jan. 2020, pp. 1–26, ISBN: 978-0-12-821379-7. DOI: 10.1016/B978-0-12-821379-7.00001-1. [Online]. Available: https://www.sciencedirect.com/science/article/pii/B9780128213797000011 (visited on 05/11/2023).

[29] *WMA - The World Medical Association-WMA Declaration of Helsinki – Ethical Principles for Medical Research Involving Human Subjects*, en-US. [Online]. Available: https://www.wma.net/policies-post/wma-declaration-of-helsinki-ethical-principles-for-medical-research-involving-human-subjects/ (visited on 05/23/2023).

[30] *Services for sensitive data (TSD) - University of Oslo*, en. [Online]. Available: https://www.uio.no/english/services/it/research/sensitive-data/index.html (visited on 05/23/2023).

[31] *HUNT Cloud - NTNU*. [Online]. Available: https://www.ntnu.edu/mh/huntcloud (visited on 05/23/2023).

[32] *DPIA for MoBa*, en, Mar. 2019. [Online]. Available: https://www.fhi.no/en/publ/forms/price-list-for-moba2/ (visited on 05/23/2023).

[33] *Protocols for the Norwegian Mother, Father and Child Cohort Study (MoBa)*, en, May 2019. [Online]. Available: https://www.fhi.no/en/publ/2012/protocols-for-moba/ (visited on 05/23/2023).

[34] M. Wold, A. Rich, D. Weeks, and L. Lutter, "Deoxyribonucleic acid (DNA),"

[35] *Human Genomic Variation*, en, Sep. 2022. [Online]. Available: https://www.genome.gov/about-genomics/educational-resources/fact-sheets/human-genomic-variation (visited on 05/08/2023).

[36] A. T. Marees, H. de Kluiver, S. Stringer, *et al.*, "A tutorial on conducting genome-wide association studies: Quality control and statistical analysis," *International Journal of Methods in Psychiatric Research*, vol. 27, no. 2, Jun. 2018, ISSN: 15570657. DOI: 10.1002/MPR.1608. [Online]. Available: /pmc/articles/PMC6001694/%20/pmc/articles/PMC6001694/?report=abstract%20https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6001694/.

[37] *Genome-Wide Association Studies Fact Sheet*, en, Sep. 2022. [Online]. Available: https://www.genome.gov/about-genomics/fact-sheets/Genome-Wide-Association-Studies-Fact-Sheet (visited on 05/04/2023).

[38] *Polygenic Risk Scores*, en, Sep. 2022. [Online]. Available: https://www.genome.gov/Health/Genomics-and-Medicine/Polygenic-risk-scores (visited on 05/04/2023).

[39] G. Rebala, A. Ravi, and S. Churiwala, "Machine Learning Definition and Basics," *An Introduction to Machine Learning*, pp. 1–17, 2019. DOI: 10.1007/978-3-030-15729-6{_}1. [Online]. Available: https://link.springer.com/chapter/10.1007/978-3-030-15729-6_1.

[40] A. Subasi, "Chapter 2 - Data preprocessing," en, in *Practical Machine Learning for Data Analysis Using Python*, A. Subasi, Ed., Academic Press, Jan. 2020, pp. 27–89, ISBN: 978-0-12-821379-7. DOI: 10.1016/B978-0-12-821379-7.00002-3. [Online]. Available: https://www.sciencedirect.com/science/article/pii/B9780128213797000023 (visited on 05/11/2023).

[41] P.-H. C. Chen, Y. Liu, and L. Peng, "How to develop machine learning models for healthcare," en, *Nature Materials*, vol. 18, no. 5, pp. 410–414, May 2019, ISSN: 1476-4660. DOI: 10.1038/s41563-019-0345-0. [Online]. Available: https://www.nature.com/articles/s41563-019-0345-0 (visited on 05/10/2023).

[42] J. Bergstra, J. B. Ca, and Y. B. Ca, "Random Search for Hyper-Parameter Optimization Yoshua Bengio," *Journal of Machine Learning Research*, vol. 13, pp. 281–305, 2012. [Online]. Available: http://scikit-learn.sourceforge.net..

[43] S. Balakrishnama and A. Ganapathiraju, "Linear Discriminant Analysis—A Brief Tutorial," vol. 11, Jan. 1998.

[44] *Papers with Code - SGD with Momentum Explained*, en. [Online]. Available: https://paperswithcode.com/method/sgd-with-momentum (visited on 06/12/2023).

[45] D. P. Kingma and J. L. Ba, "Adam: A Method for Stochastic Optimization," *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*, Dec. 2014. [Online]. Available: https://arxiv.org/abs/1412.6980v9.

[46] *RMSProp - Cornell University Computational Optimization Open Textbook - Optimization Wiki*. [Online]. Available: https://optimization.cbe.cornell.edu/index.php?title=RMSProp.

[47] M. D. Zeiler, "ADADELTA: An Adaptive Learning Rate Method," Dec. 2012. [Online]. Available: https://arxiv.org/abs/1212.5701v1.

[48] I. Loshchilov and F. Hutter, "DECOUPLED WEIGHT DECAY REGULARIZATION," [Online]. Available: https://github.com/loshchil/AdamW-and-SGDW.

[49] *Resampling strategies for imbalanced datasets | Kaggle*. [Online]. Available: https://www.kaggle.com/code/rafjaa/resampling-strategies-for-imbalanced-datasets/notebook.

[50] J. Snoek, H. Larochelle, and R. P. Adams, "Practical Bayesian Optimization of Machine Learning Algorithms," in *Advances in Neural Information Pro-*

cessing Systems, vol. 25, Curran Associates, Inc., 2012. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2012/hash/05311655a15b75fab86956663e1819cd-Abstract.html (visited on 05/10/2023).

[51] Fernando Nogueira, *BayesianOptimization: Open source constrained global optimization tool for Python*, 2014. [Online]. Available: https://github.com/fmfn/BayesianOptimization.

[52] P. Refaeilzadeh, L. Tang, and H. Liu, "Cross-Validation," en, in *Encyclopedia of Database Systems*, L. LIU and M. T. ÖZSU, Eds., Boston, MA: Springer US, 2009, pp. 532–538, ISBN: 9780387399409. DOI: 10.1007/978-0-387-39940-9_565. [Online]. Available: https://doi.org/10.1007/978-0-387-39940-9_565 (visited on 05/10/2023).

[53] J. Juodakis, K. Ytterberg, C. Flatley, P. Sole-Navais, and B. Jacobsson, *Time-varying effects are common in genetic control of gestational duration*, en, Feb. 2023. DOI: 10.1101/2023.02.07.23285609. [Online]. Available: https://www.medrxiv.org/content/10.1101/2023.02.07.23285609v1 (visited on 05/12/2023).

[54] U. M. Khaire and R. Dhanalakshmi, "Stability of feature selection algorithm: A review," *Journal of King Saud University - Computer and Information Sciences*, vol. 34, no. 4, pp. 1060–1073, Apr. 2022, ISSN: 1319-1578. DOI: 10.1016/J.JKSUCI.2019.06.012.

[55] F. Pedregosa FABIANPEDREGOSA, V. Michel, O. Grisel OLIVIERGRISEL, *et al.*, "Scikit-learn: Machine Learning in Python," *Journal of Machine Learning Research*, vol. 12, no. 85, pp. 2825–2830, 2011, ISSN: 1533-7928. [Online]. Available: http://jmlr.org/papers/v12/pedregosa11a.html.

[56] *Normal, Ledoit-Wolf and OAS Linear Discriminant Analysis for classification*, en. [Online]. Available: https://scikit-learn/stable/auto_examples/classification/plot_lda.html (visited on 05/23/2023).

[57] V. Michel, F. Pedregosa, A. Aides, *et al.*, *scikit-learn/naive_bayes.py*, Dec. 2022. [Online]. Available: https://github.com/scikit-learn/scikit-learn/blob/364c77e04/sklearn/naive_bayes.py#L1069.

[58] T. P. York, L. J. Eaves, P. Lichtenstein, *et al.*, "Fetal and Maternal Genes' Influence on Gestational Age in a Quantitative Genetic Analysis of 244,000 Swedish Births," *American Journal of Epidemiology*, vol. 178, no. 4, pp. 543–550, Jan. 2013. DOI: 10.1093/aje/kwt005.

A
Data

A.1 Folds

Table A.1: Distribution of gestational duration within the different batches.

Fold	Class	Samples	Min (days)	Max (days)	Average (days)
0	Case	83	219	258	249
	Control	3177	259	304	282
1	Case	84	182	258	248
	Control	3176	259	306	282
2	Case	84	211	258	247
	Control	3176	259	307	282
3	Case	84	213	258	249
	Control	3176	259	306	282
4	Case	84	193	258	248
	Control	3176	259	307	282
All	Case	419	182	258	248
	Control	15881	259	307	282

A.2 Hyperparameter values

Table A.2: Boundaries for hyperparameter tuning

SGD			
Batch size	$(10, 80)$	Nesterov	$(0/1)$
Momentum	$(0.0, 0.99)$	Use Ema	$(0/1)$
Initial learning rate	$(1e^{-12}, 0.1)$		
SGD2			
Initial learning rate	$(0.0, 0.1)$	*(Learning rate schedule)*	
Momentum	$(0.0, 0.9)$	Decay steps	$(1e^{3}, 1e^{5})$
Decay rate	$(0.5, 0.9)$	Batch size	$(30, 100)$
Clipnorm	$(0.5, 2)$		
AdaMax			
Initial learning rate	$(1e^{-07}, 1)$	Use Ema	$(0/1)$
b1	$(0.1, 0.99)$	b2	$(0.5, 0.99)$
Eps	$(1e^{-07}, 0.1)$	Clipnorm	$(1, 4)$
Batch size	$(10, 60)$	Ema momentum	$(0, 0.9)$
AdaDelta			
Batch size	$(30, 50)$	Use Ema	$(0/1)$
Clipnorm	$(1.5, 3)$	Decay rate	$(0.2, 0.9)$
Eps	$(1e^{-04}, 0.8)$	Initial learning rate	$(1e^{-07}, 1)$
Adam			
Initial learning rate	$(1e^{-07}, 0.1)$	b1	$(0.8, 1.0)$
b2	$(0.8, 1.0)$	Eps	$(1e^{-10}, 0.01)$
Clipnorm	$(0.3, 5)$	Batch size	$(20, 50)$
AdamW			
Initial learning rate	$(1e^{-07}, 0.5)$	Use Ema	$(0/1)$
b1	$(0.8, 0.99)$	b2	$(0.9, 1.0)$
Eps	$(1e^{-07}, 0.05)$	Clipnorm	$(1, 2.5)$
Batch size	$(30, 60)$	Ema momentum	$(0, 0.9)$
Decay rate	$(0, 0.8)$		
RMSprop			
Initial learning rate	$(1e^{-06}, 0.1)$	Use Ema	$(0/1)$
Eps	$(1e^{-08}, 0.01)$	Clipnorm	$(1, 3)$
Batch size	$(10, 50)$	Ema momentum	$(0, 0.9)$
Decay rate	$(0.1, 0.99)$		

Table A.3: Tuned hyperparameter values

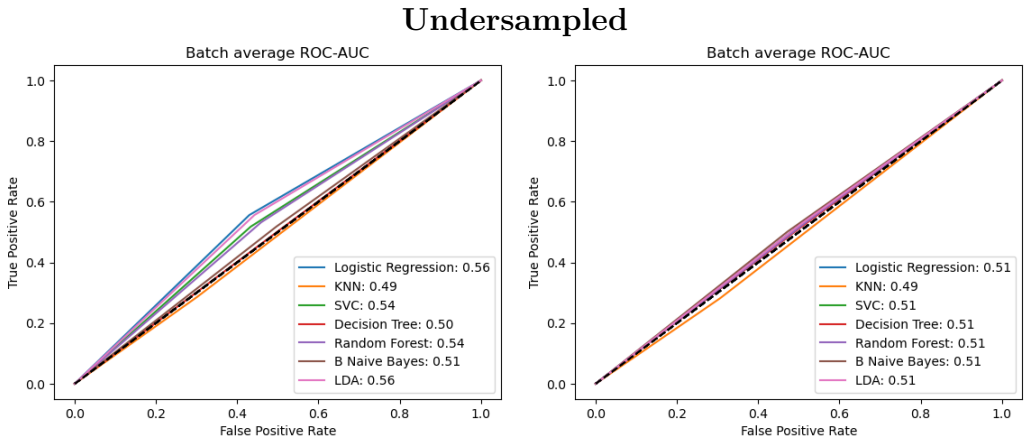
SGD			
Batch size	24	Nesterov	*Yes*
Momentum	0.0271	Use Ema	*Yes*
Initial learning rate	0.0878		
SGD2			
Batch size	32		
Initial learning rate	0.0140	*(Learning rate schedule)*	
Momentum	0.1783	Decay steps	56310
Decay rate	0.6669	Clipnorm	1.5057
AdaMax			
Initial learning rate	0.0011	Use Ema	*No*
b1	0.6990	b2	0.8730
Eps	0.0902	Clipnorm	3.9935
Batch size	50	Ema momentum	*NA*
AdaDelta			
Batch size	34	Use Ema	*Yes*
Clipnorm	1.6343	Decay rate	0.3598
Eps	0.4983	Initial learning rate	0.0068
Adam			
Initial learning rate	0.0204	b1	0.8691
b2	0.8794	Eps	0.0069
Clipnorm	0.8794	Batch size	36
AdamW			
Initial learning rate	0.0021	Use Ema	*No*
b1	0.8724	b2	0.9954
Eps	0.0068	Clipnorm	1.2878
Batch size	36	Ema momentum	*NA*
Decay rate	0.0260		
RMSprop			
Initial learning rate	0.0042	Use Ema	*Yes*
Eps	0.0029	Clipnorm	1.5942
Batch size	47	Ema momentum	0.6049
Decay rate	0.8054		

B

Classic Models

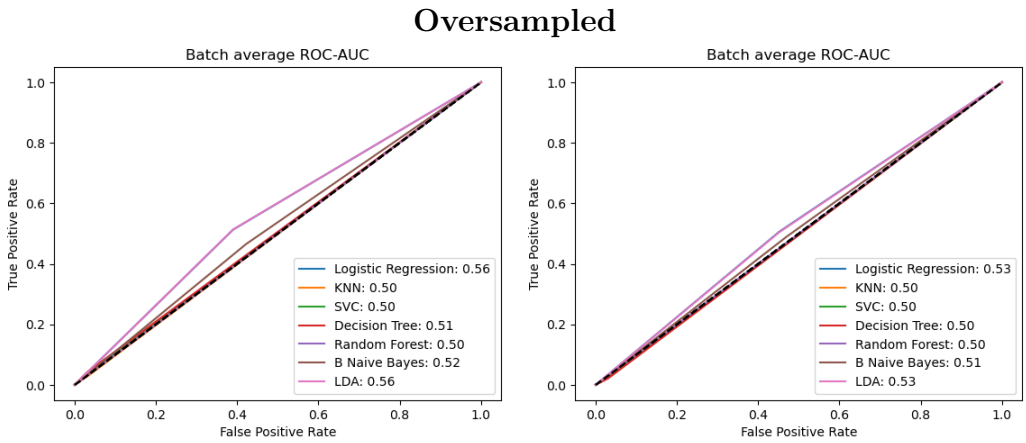
ROC-AUC plots of the fold average for classic models with different selection of SNPs, sampling methods and origin of genome.

B.1 Top 23



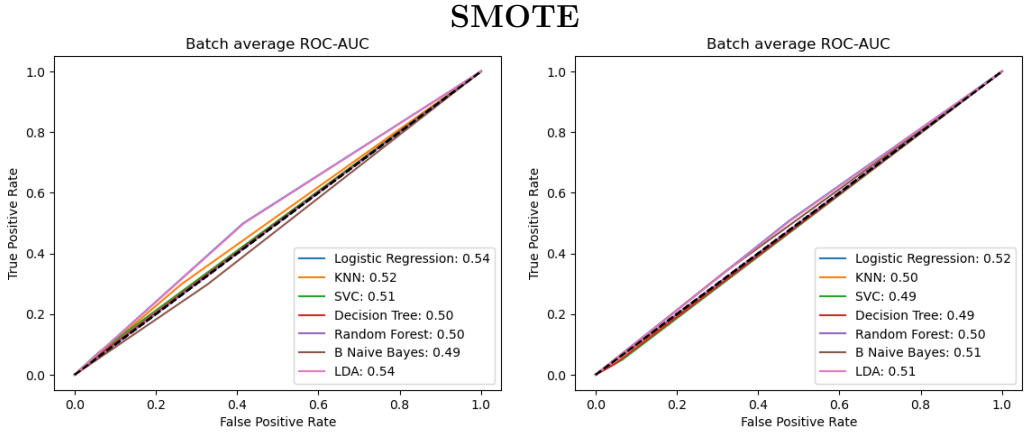
(a) Maternal genome **(b)** Fetal genome

Figure B.1: AUC for different classic models trained and tested on the top 23 SNPs, with basic undersampling of training data



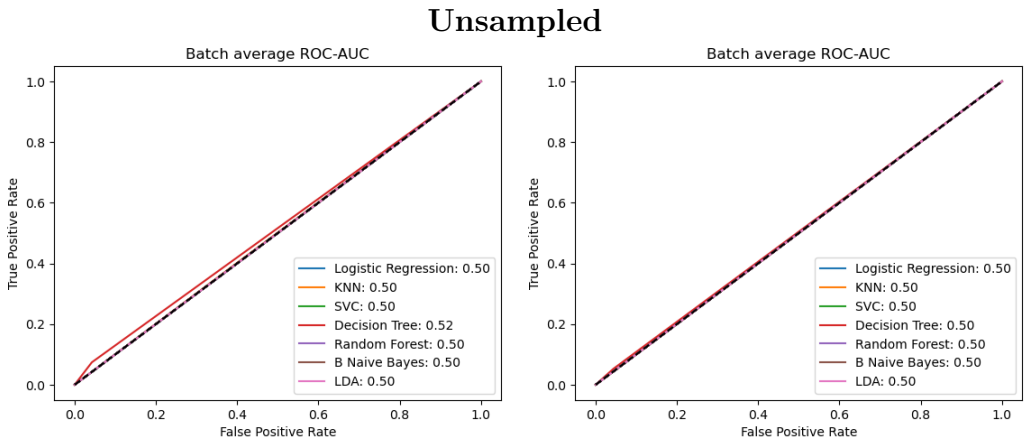
(a) Maternal genome **(b)** Fetal genome

Figure B.2: AUC for different classic models trained and tested on the top 23 SNPs, with basic oversampling of training data



(a) Maternal genome **(b)** Fetal genome

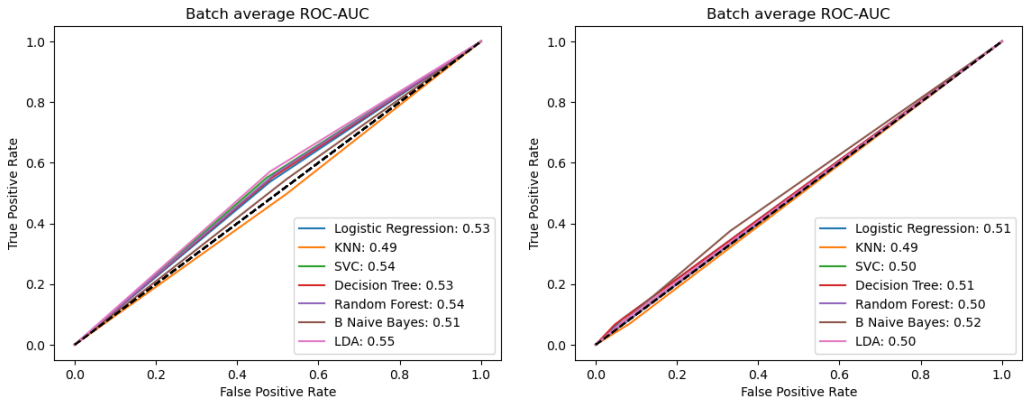
Figure B.3: AUC for different classic models trained and tested on the top 23 SNPs, with SMOTE oversampling of training data



(a) Maternal genome **(b)** Fetal genome

Figure B.4: AUC for different classic models trained and tested on the top 23 SNPs, without re-sampling of training data

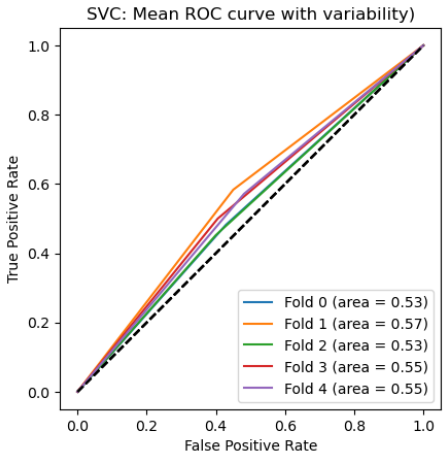
B.2 Extended Set



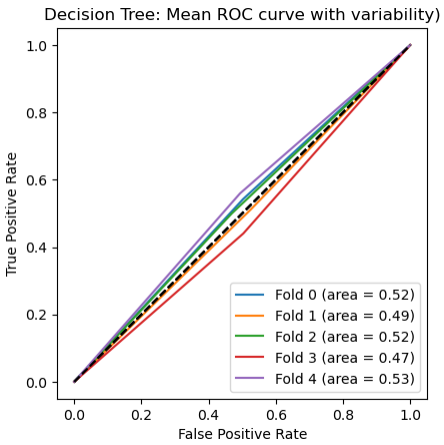
(a) Random undersampling **(b)** Random oversampling

Figure B.5: AUC for different classic models trained and tested on the extended selection of SNPs from the maternal genome

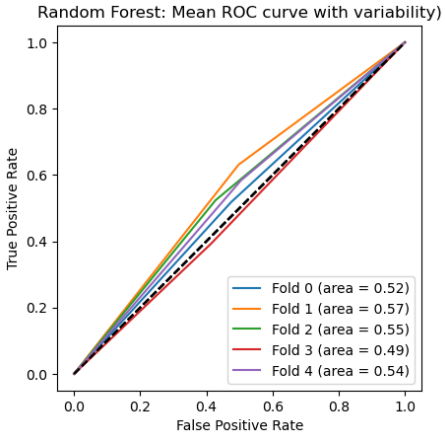
B.3 Folds: Top 23, Maternal, Undersampled



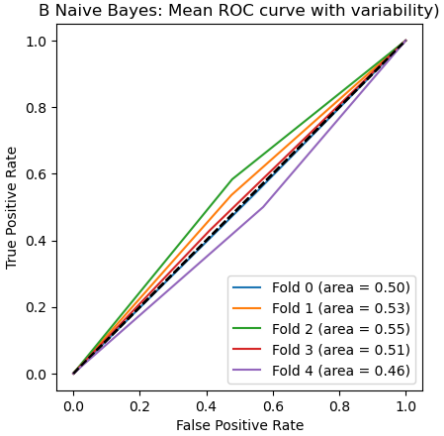
(a) SVC



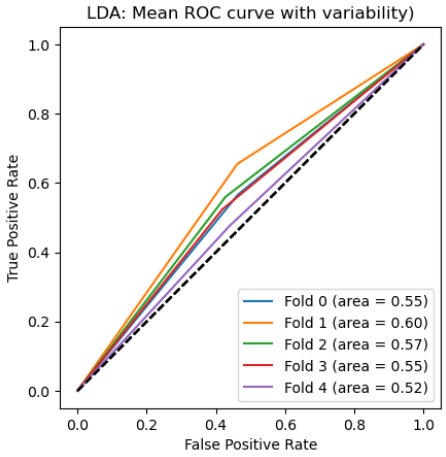
(b) DT



(c) RF



(d) BNB



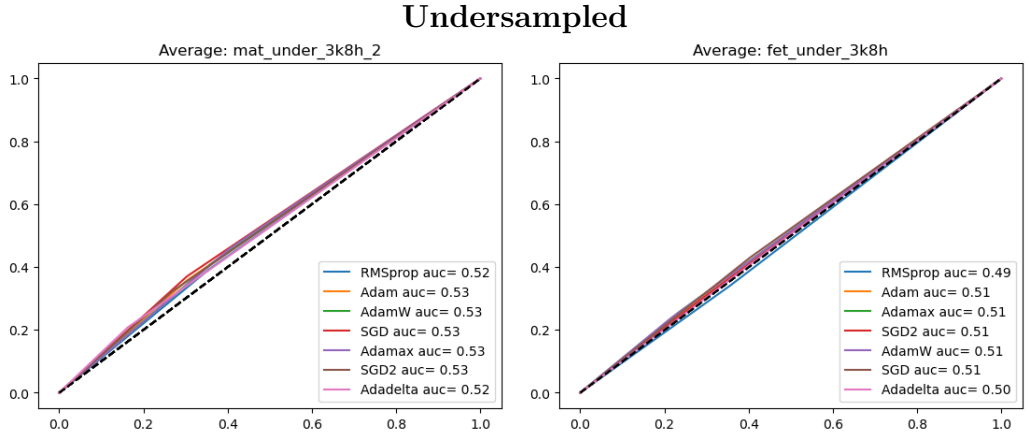
(e) LDA

Figure B.6: AUC for classic models by fold.

C
Network Models

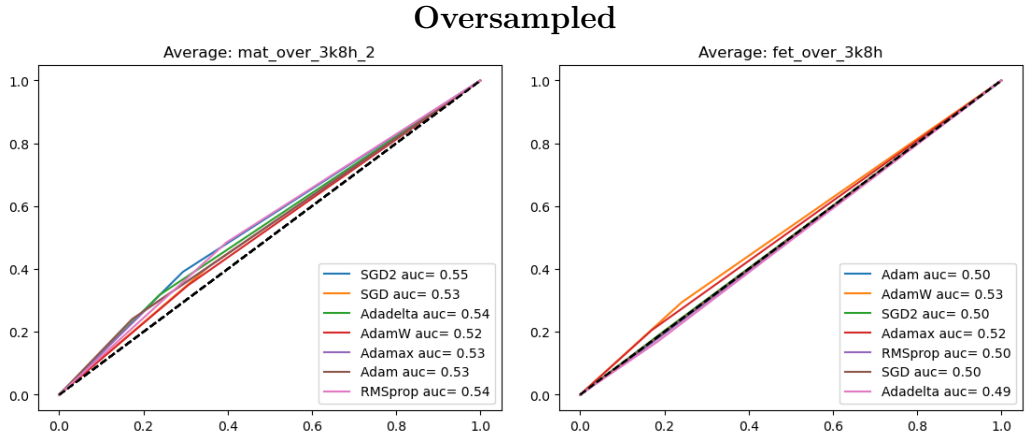
ROC-AUC plots of batch averages for network models with different selection of SNPs, sampling methods and origin of genome.

C.1 Top 23: Maternal / Fetal



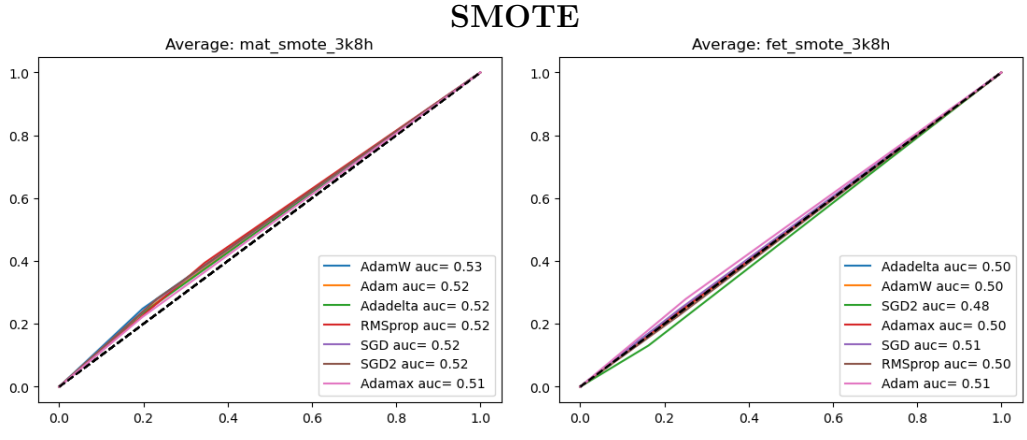
(a) Maternal genome **(b)** Fetal genome

Figure C.1: AUC for different network models trained and tested on the top 23 SNPs, with basic undersampling of training data



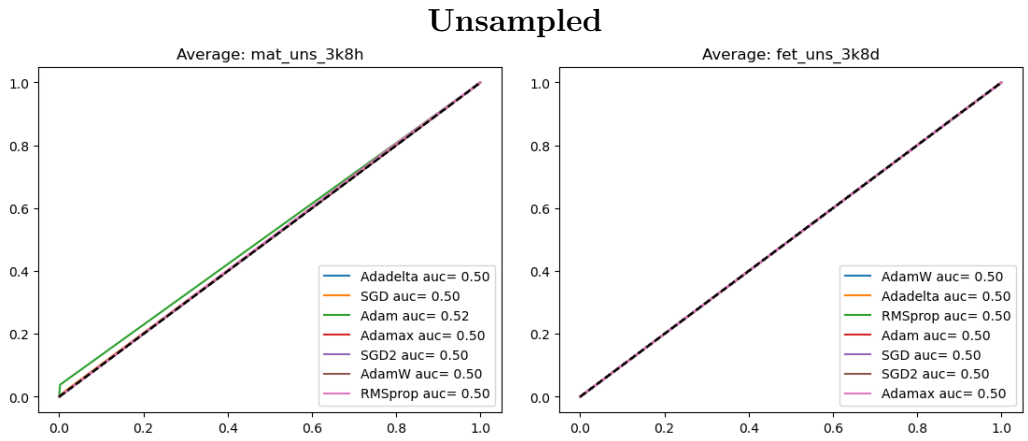
(a) Maternal genome **(b)** Fetal genome

Figure C.2: AUC for different network models trained and tested on the top 23 SNPs, with basic oversampling of training data



(a) Maternal genome **(b)** Fetal genome

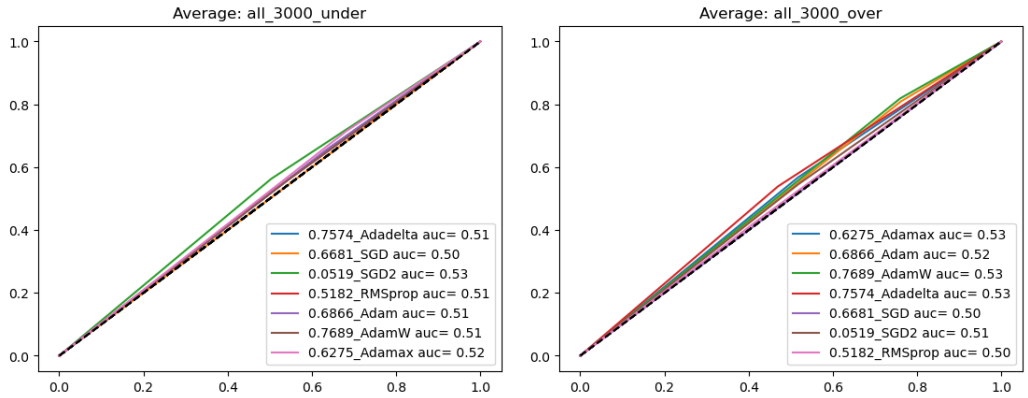
Figure C.3: AUC for different network models trained and tested on the top 23 SNPs, with SMOTE oversampling of training data



(a) Maternal genome **(b)** Fetal genome

Figure C.4: AUC for different network models trained and tested on the top 23 SNPs, without re-sampling of training data

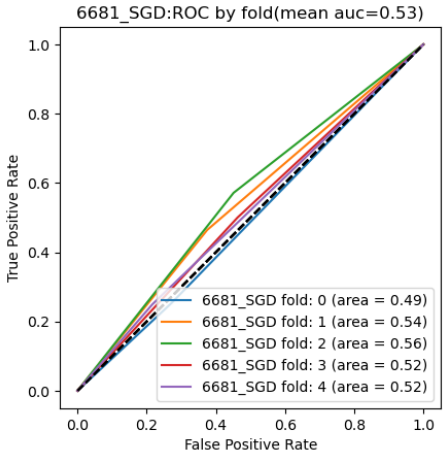
C.2 Extended Set: Maternal



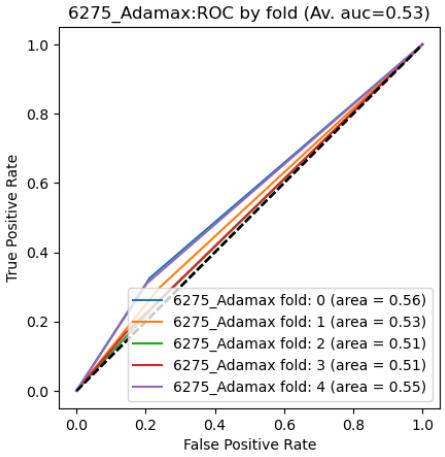
(a) Random undersampling **(b)** Random oversampling

Figure C.5: AUC for different classic models trained and tested on the extended selection of SNPs from the maternal genome

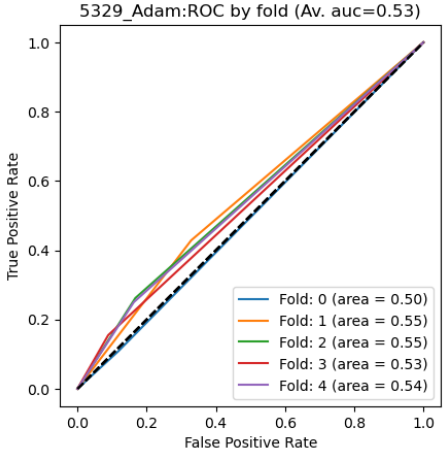
C.3 Folds: Top 23, Maternal, Oversampled



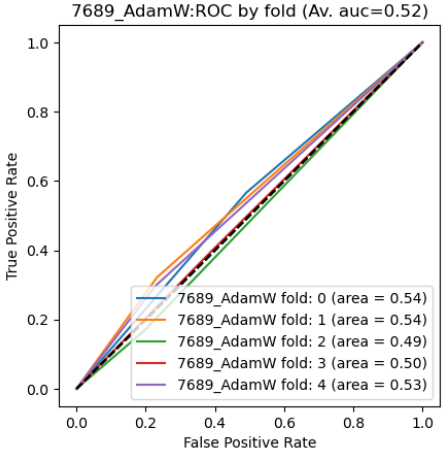
(a) SGD



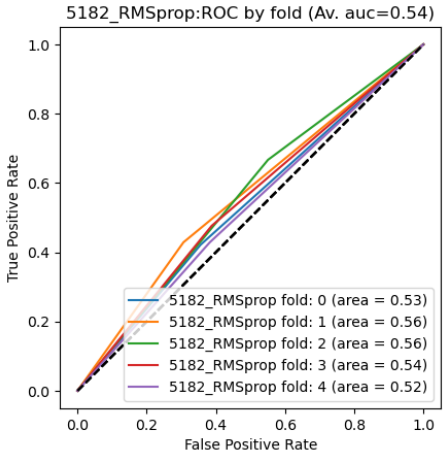
(b) Adamax



(c) Adam



(d) AdamW

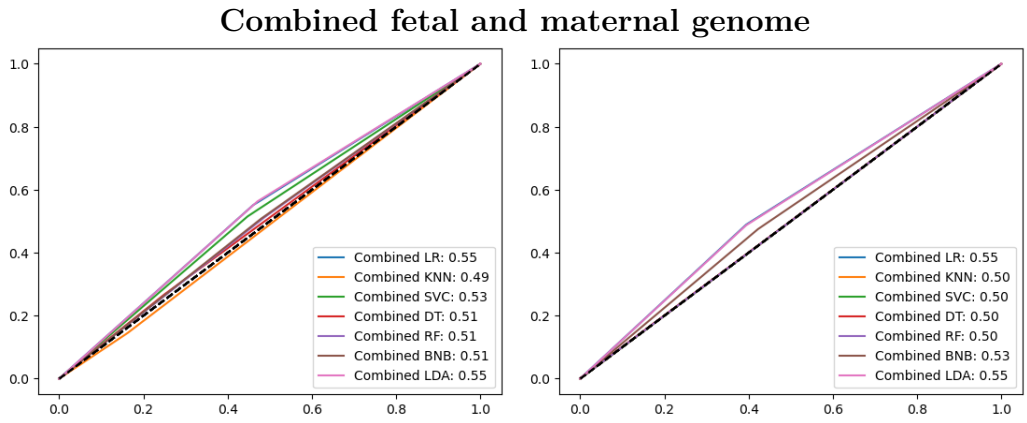


(e) RMSprop

Figure C.6: AUC for network models by fold.

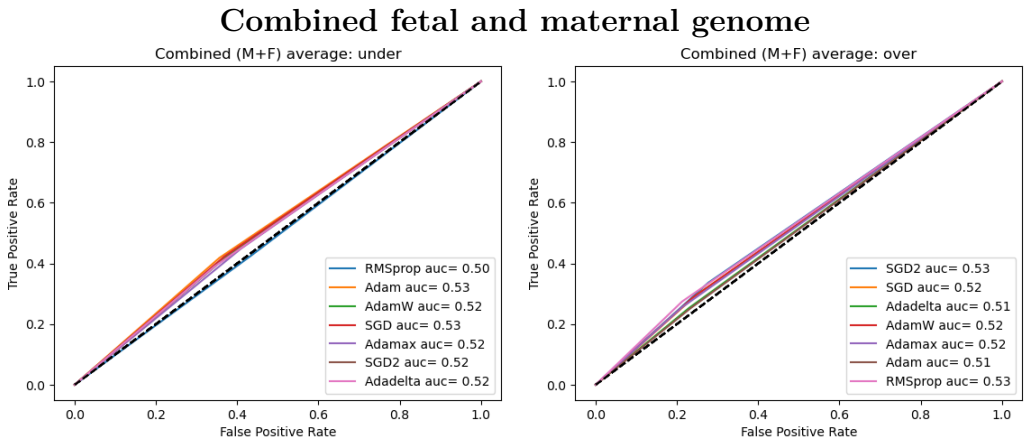
D

Combined Predictions



(a) Undersampled **(b)** Oversampled

Figure D.1: AUC for predictions made by combining the predictions from classic models fitted on the maternal and fetal genome respectively.



(a) Undersampled **(b)** Oversampled

Figure D.2: AUC for predictions made by combining the predictions from network models fitted on the maternal and fetal genome respectively.

DEPARTMENT OF SOME SUBJECT OR TECHNOLOGY
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden
www.chalmers.se

CHALMERS
UNIVERSITY OF TECHNOLOGY