Czech Technical University in Prague
Faculty of Nuclear Sciences and Physical Engineering

Department of Mathematics
Programme: Applied Mathematical Stochastic Methods

Detection of Relationships Between Titles of the Old Kingdom of Egypt

MASTER THESIS

Author:	Bc. Pavel Stojaspal
Supervisor:	Ing. Marek Bukáček, Ph.D.
Advisor:	Mgr. et Mgr. Veronika Dulíková, Ph.D.
Year:	2025

ČVUT
ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE

ZADÁNÍ DIPLOMOVÉ PRÁCE

I. OSOBNÍ A STUDIJNÍ ÚDAJE

Příjmení: **Stojaspal** Jméno: **Pavel** Osobní číslo: **486476**

Fakulta/ústav: **Fakulta jaderná a fyzikálně inženýrská**

Zadávající katedra/ústav: **Katedra matematiky**

Studijní program: **Aplikované matematicko-stochastické metody**

II. ÚDAJE K DIPLOMOVÉ PRÁCI

Název diplomové práce:

Detekce vztahů mezi tituly Starého Egyptského království

Název diplomové práce anglicky:

Detection of relationships between titles of the Old Kingdom of Egypt

Pokyny pro vypracování:

1. Proveďte základní analýzu dat poskytnutých Českým Egyptologickým ústavem FF UK.
2. Naimplementujte logistický regresní model včetně metody pro zúžení parametrického prostoru (například PCA).
3. Naimplementujte vhodné varianty neuronové sítě.
4. Vybrané algoritmy zdokumentujte, popřípadě ilustrujte jejich vlastnosti na umělých datech.
5. Detekujte vazby mezi tituly a dalšími příznaky jedinců z datového vzorku.

Seznam doporučené literatury:

[1] J. Anděl, Statistické metody. MatfyzPress, 2019.
[2] I. J. Goodfellow, Y. Bengio, A. Courville, Deep Learning. MIT Press, 2016.
[3] V. Dulíková, M. Bárta, Addressing the dynamics of change in ancient Egypt: Complex network analysis. Charles University, 2020.

Jméno a pracoviště vedoucí(ho) diplomové práce:

Ing. Marek Bukáček, Ph.D. katedra matematiky FJFI

Jméno a pracoviště druhé(ho) vedoucí(ho) nebo konzultanta(ky) diplomové práce:

Mgr. et Mgr. Veronika Dulíková, Ph.D. Filozofická fakulta, Univerzita Karlova

Datum zadání diplomové práce: **15.03.2024** Termín odevzdání diplomové práce: **06.01.2025**

Platnost zadání diplomové práce: **15.03.2026**

Ing. Marek Bukáček, Ph.D.
podpis vedoucí(ho) práce

prof. Ing. Zuzana Masáková, Ph.D.
podpis vedoucí(ho) ústavu/katedry

doc. Ing. Václav Čuba, Ph.D.
podpis děkana(ky)

III. PŘEVZETÍ ZADÁNÍ

Diplomant bere na vědomí, že je povinen vypracovat diplomovou práci samostatně, bez cizí pomoci, s výjimkou poskytnutých konzultací. Seznam použité literatury, jiných pramenů a jmen konzultantů je třeba uvést v diplomové práci.

25.3.2024
Datum převzetí zadání

Podpis studenta

Declaration

I hereby declare that I have prepared the submitted work on my own and that I have listed all the literature used.

Further, I herby declare that I used artificial intelligence (AI) according to the framework rules for the use of artificial intelligence at CTU (Methodical guideline No. 5/2023). I declare that AI was used mainly for grammar checking and improving graphs.

In Prague ....

Bc. Pavel Stojaspal

Acknowledgements

I would like to thank my supervisor Ing. Marek Bukáček, Ph.D. for all advices and willingness he was providing me during writing this thesis. I would also like to thank to Mgr. et Mgr. Veronika Dulková, Ph.D. for her support during the finishing of this thesis.

Special thanks also go to my entire family for all support they gave me during my studies. It would not be possible without them.

This thesis was created in connection with the project *Titles and bones of ancient Egyptian officials: New mathematical approach to analysing Old Kingdom data, Czech Science Foundation (GAČR), Junior Star 2024–2028, Project No. 24-10275M.*

Bc. Pavel Stojaspal

Název práce:
Detekce vztahů mezi tituly Starého Egyptského království

Autor: Bc. Pavel Stojaspal

Obor: Aplikované matematicko-stochastické metody
Druh práce: Diplomová práce

Vedoucí práce: Ing. Marek Bukáček, Ph.D.
Fakulta jaderná a fyzikálně inženýrská,
České vysoké učení technické v Praze
Konzultant: Mgr. et Mgr. Veronika Dulíková, Ph.D.
Filozofická fakulta Univerzity Karlovy

Abstrakt: Tato diplomová práce si klade za cíl prozkoumat vazby mezi titulem vezíra a dalšími tituly období Staré říše. Jako zdroj dat využíváme jedinečnou databázi Maatbase, vytvořenou na Českém egyptologickém ústavu. Na základě analýzy vztahů mezi jednotlivci a jejich tituly jsme vytvořili dva datové soubory, které jsme dále využili při modelování. Za účelem prozkoumání výše zmíněných vazeb jsme pomocí logistické regrese a vícevrstvého perceptronu vytvořili dvanáct modelů. Tyto modely jsme následně analyzovali z několika úhlů pohledu. Nejprve jsme vyhodnotili jejich adekvátnost, jak dobře popisují data. Poté jsme identifikovali jednotlivce, kterým model přiřadil vysokou pravděpodobnost titulu vezíra, ačkoliv jím ve skutečnosti nebyli. Nakonec jsme s využitím SHAP hodnot zkoumali vliv jednotlivých vstupních proměnných na výslednou predikci. Naše výsledky jsme porovnali s již publikovanými pracemi. Jako vedlejší produkt modelování prezentujeme výsledek PCA transformace, aplikované na kategorie titulů. Ve výsledné transformaci lze jasně odlišit kategorii rodina, která se promítla do odlišných komponent než ostatní kategorie. Práce je zakončena přílohou obsahující množství doprovodných tabulek a grafů, které doplňují výsledky uvedené v hlavních částech práce.

Klíčová slova: Stará říše, vezír, strojové učení, logistická regrese, vícevrstvý perceptron, SHAP hodnoty.

Title:
Detection of relationships between titles of the Old Kingdom of Egypt

Author: Bc. Pavel Stojaspal

Programme: Applied Mathematical Stochastic Methods
Type of thesis: Master thesis

Supervisor: Ing. Marek Bukáček, Ph.D.
Faculty of Nuclear Sciences and Physical Engineering,
Czech Technical University in Prague
Advisor: Mgr. et Mgr. Veronika Dulíková, Ph.D.
Faculty of Arts, Charles University

Abstract: This master thesis aims to analyze the patterns behind the vizier and other titles of the Old Kingdom of Egypt. As a data source we will use unique Maatbase database which was created under the Czech Institute of Egyptology. By inspecting persons together with their titulary, we create two datasets that we will use in our modelling. Later, by means of logistic regression and multilayer perceptron, we implement twelve models to analyze the patterns. The model analysis is performed from multiple perspectives. Firstly, we evaluate the performance of the models. Secondly, we highlight persons with high predictions although these persons were not viziers. Finally, we use SHAP values to analyze the importance of the features in each model. The results are compared with other publications. As a by-product we show a projection of titles categories onto the first two principal component, where family category is easily distinguished from other categories. This thesis is concluded with Attachment that present multiple tables and figures that complement our results in more details.

Key words: Old Kingdom of Egypt, vizier, machine learning, logistic regression, multilayer perceptron, SHAP values.

Contents

Notation

NN	Artificial Neural Network
MLP	Multilayer Perceptron
PCA	Principal Component Analysis
LR	Logistic Regression
VIF	Variance Inflation Factor

Numbers, Arrays, Algebra, Statistics

a	A scalar (integer)
$\mathbf{x}$	A vector
x_i	A *i-th* element of the vector $\mathbf{x}$
$\mathbf{A}$	A matrix
$\mathbf{A}^T$	A transpose of the the matrix $\mathbf{A}$
$\mathbf{A_{i,j}}$	A element of the matrix $\mathbf{A}$ at *i-th* row and *j-th* column
$\mathbf{A}_{i,:}$	A *i-th* row of the matrix $\mathbf{A}$
$\mathbf{A}_{:,j}$	A *j-th* column of the matrix $\mathbf{A}$
$\mathbb{N}$	The set of natural numbers
$\mathbb{R}$	The set of real numbers
$\mathbb{R}^n$	The n-dimensional space of real numbers
$\mathbb{E}(X)$	The mean value of random variable X
$\mathrm{Var}(X)$	The variance of random variable X

Introduction

Ancient Egypt has fascinated historians and the general public for many decades. It is studied from multiple perspectives, such as the evolution of its centralized institutions or the roles of exceptional individuals. One category of such individuals is viziers.

This master's thesis aims to explore patterns behind the vizier title in the Old Kingdom of Egypt. For this analysis, we will create a dataset derived from the Maatbase database. This dataset will consist of individuals and their titles, which will be analyzed using various machine learning approaches. Patterns between the vizier title and other titles will be examined and discussed.

This thesis consists of four main chapters. The first chapter is dedicated to an introduction to the Old Kingdom of Egypt. The author reviews some important behaviors of the society and its structure. Later, he will introduce the Maatbase database, which will serve as the only source of our data. This chapter is concluded with an overview of research not only in the field of the Old Kingdom of Egypt but also from other societies.

The second chapter summarizes the machine learning methods used in this thesis. It describes principal component analysis together with an example on the real Egyptological data. The result of this example will be used later in our models. In addition, this chapter summarized logistic regression and neural network models. Both sections describe regularization techniques that will be applied in later models. The following section is dedicated to SHAP values, which will play a crucial role in both model fitting and post-model analysis. This chapter ends with an overview of the scores that will be used to measure the performance of fitted models.

The third chapter describes how the models were created. The first section defines the objectives of our modeling. Later, it introduces our target feature, followed by feature selection. In the latter case, we will describe how we derive our own dataset. The end of this chapter comments on how we derived our final logistic regression and multilayer perceptron models. In this section, we create in total twelve different models, which will be studied and summarized in the next chapter.

The fourth chapter summarizes our findings. It is divided into three sections: model, person, and title summary. In each section, we will analyze our models from different points of view. In the model summary, we discuss the performance of the models. In the person summary, we will highlight persons that put the most difficulties into model predictions. The last analysis will be on the importance of input features. By means of SHAP values, we will make a list of the most important features in our modeling.

These chapters are followed by conclusion, where we summarize our findings. This thesis is concluded with the reference section and the attachment. The attachment includes several tables and figures that complement our findings.

Chapter 1

Introduction into Egyptological Research Project

This section aims to briefly introduce research in the field of Ancient Egypt.

This chapter is divided into three sections. First, we will take a quick historical look at Ancient Egypt to recap some of the main hierarchy-related aspects. This review will improve our understanding of the variables in the later models. In the second section, we will introduce the Maatbase dataset, which plays a crucial role in our modeling. Finally, in the last section, we will mention some similar studies that focus on social research, although not strictly in the same space and time.

1.1 Egyptological Research

Ancient Egypt is a source of many illustrative examples of both the growth and decline of centralized institutions, as well as the role of exceptional individuals and kings [1]. Monumental pyramids that exceed a height of 100 meters [2], mystically shrouded cults, pharaohs representing the top of the hierarchy and, at the same time, a connection to the gods, fascinating paintings and art, unmistakable sarcophagi covered in mystery - for these and many other reasons, Ancient Egypt has attracted both historians and the general public for many decades. This thesis will focus on the period known as the Old Kingdom, which spans from 2592 to 2120 BC [2]. Although the term Old Kingdom refers to the period from the Third Dynasty to the Sixth Dynasty[1], the following text primarily reviews the social, economic, and cultic aspects of the Fifth Dynasty. This will provide a sufficient background for the thesis.

Let us begin this review by introducing the vizier, the central figure of this work. The vizier is one of many titles in the Old Kingdom that could be attained by a living person. However, this title stands out compared to others. The vizier held a role that could be viewed as equivalent to a prime minister in modern governments [2].

[1] According to [2], [3], the dynasties are approximately dated as follows: the Third Dynasty (2592–2544 BC), Fourth Dynasty (2543–2436 BC), Fifth Dynasty (2435–2306 BC), and Sixth Dynasty (2305–2150 BC).

The individuals who held this title played a crucial role in the administrative hierarchy. They were responsible for tax collection, jurisdiction, grain storage, other goods, and state-sponsored construction projects [2]. Similarly to the pharaoh or a modern-day prime minister, there could only be one vizier at a time. Although knowledge about the vizier in the Third Dynasty is sparse, we know that in the Fourth Dynasty this office was held exclusively by royal family members [2]. Due to the growing complexity of the state at the end of the Fourth Dynasty, more non-royal members began to be employed in the state administration during the Fifth and Sixth Dynasties. From this period onward, non-royal individuals could also reach the vizierial office [2]. Since this prestigious title was associated with extraordinary individuals of significant influence, knowing the exact person who held this title at any given time might be as important as knowing the name of the ruling pharaoh. Unfortunately, this knowledge is incomplete, as there are periods with unknown viziers. Therefore, this work aims to analyze these gaps using machine learning approaches applied to the unique Maatbase database from Czech egyptologist Mgr. et Mgr. Veronika Dulíková, Ph.D.

So far, we have introduced the title of vizier, but individuals in the Old Kingdom could earn thousands of other titles. To be more specific, the current Maatbase contains more than 2,200 unique title names. Egyptologists have categorized these titles into 160 types. To avoid an exhaustive list of all possibilities, let us focus on the fields to which these titles are related. There are titles connected to fields such as administration, epithets, family, priesthood, the privy, rank, and labor. It is worth mentioning that the religious and official titles were not strictly independent, some individuals held both priestly and administrative roles [1]. These roles overlapped much more than we might intuitively expect based on the current world.

Furthermore, titles did not follow a fixed hierarchical progression like in modern bureaucracies or religious institutions, allowing individuals to hold high-status roles without formally advancing through lower ranks.

Extensive information in the Maatbase database has been acquired from tombs. Therefore, it is worth mentioning the types, design, and decoration of such burial grounds. During the Fifth Dynasty, two dominant tomb types can be distinguished [1]: on one side, the monumental, multi-roomed mastaba with significant decoration, and on the other, relatively modest tombs where numerous family members were buried together in a single tomb. The decoration of these tombs was highly specific to social status, projecting status, rank, and kingship [1][4]. As the state underwent many changes, so too did the construction of non-royal tombs. Suddenly, fully decorated tomb complexes began to appear for non-royal individuals, something that was uncommon before the reign of Nyuserre [1]. This shift suggests that for machine learning modeling, features such as timestamps might be crucial. Otherwise, we risk confusing royal and non-royal individuals.

Returning to the tombs themselves, they may differ in size, decorative richness, and number of decorative themes. There may be porticos and walls with thousands of figures depicting the owner of the tomb, his family, and other individuals who may not necessarily be buried in the tomb. These figures illustrate various scenes, such as agricultural activities, fishing, feeding animals, butchering, sailing and rowing boats, boat and statue production, carpentry, hippopotamus fighting, bread baking, and even beer making. These scenes often feature officials and servants with their titles and names. For example, in the tomb of the non-royal individual Ty, about 1,800 depictions of different people were found [1], with Ty himself appearing about a hundred times in the tomb's decorations.

Beyond the famous figures, each tomb offers other aspects to study. For example, cult chambers contain false doors, which were viewed as connections to the world of the dead and served as places for offerings. Since the size, decoration, and even number of false doors could vary from tomb to tomb, inspecting them could reveal the status of the owner. Another important chapel is the burial one, where the actual sarcophagus was placed. Furthermore, the entrance to the burial chapels was through a shaft. Once again, studying the burial chamber and its shaft can reveal much about the owner.

As mentioned before, starting from the reign of Pharaoh Nyuserre, the tombs of non-royal members began to demonstrate their wealth. However, this was not the only change that the state underwent during his reign. Nepotism also increased during this time. Nepotism refers to the practice of a certain family trying to maintain or even increase its influence in a given sphere (such as administration or priesthood) by reserving corresponding functions exclusively for its own members. In such nepotistic families, titles were inherited and passed within the family, regardless of qualifications. Nepotism elevated the social status of these families, but it also had many negative effects on society as a whole, and according to [1], it notably contributed to the decline of the Old Kingdom. To counterbalance the rising power of non-royal members, pharaohs adopted the practice of marrying their daughters to these influential families. These blood marriages secured loyalty and support for the pharaoh.

While the existence of nepotism in the Old Kingdom is a well-known fact, there remains an open question: to whom were these princesses married? Although princesses are relatively well documented, information about the backgrounds of their husbands is, in many cases, unknown. However, the Maatbase database might provide insight in this area, as it has already helped solve a mystery. It is well known that the aforementioned Ty had five children [1], but his ancestors were unknown until recently. Through an analysis of title transfers from father to son, a man of the same name and similar titulary, including a rare ones, was identified [1]. This match suggests that this man could potentially be Ty's father. As demonstrated, Maatbase has the potential to reveal unknowns about the Old Kingdom, thus improving our understanding of its society. For that reason, this thesis also aims to discover potential missing connections, which may not be apparent through traditional Egyptological research methods.

1.2 Overview of Maatbase Database

The goal of this section is to introduce the Maatbase database. Although we will discuss Maatbase in more detail in a later chapter dedicated to feature selection, it is already an appropriate time to become familiar with the sole data source for this thesis.

The Maatbase database was launched in 2006 by Mgr. et Mgr. Veronika Dulíková, Ph.D. [5]. This database consists of records from both published documentation and unpublished sources [5], including the Giza Archives (Boston), Project-Giza (Leipzig), and the archive of the Czech Institute of Egyptology (Prague) [5][2]. It is continuously updated, so its size is increasing.

Currently, the database consists of 44 tables, specifically tables that contain information about person names, a list of all titles, person-title relations, and family relations. Furthermore, individuals are linked to the tombs where they were buried. We can retrieve details related to the dimensions of the tomb and their owners. In addition, information about cult chambers and the dimensions of false doors is available, as well as the dimensions of the shafts and the equipment within the burial chambers. The schema of the database is shown in 3. The content of the database is summarized in Table 1.1. There are records of over 4,900 individuals, 2,200 titles, and 13,600 person-title relations.

Feature type	**N. of records**
Persons	4,962
· Men	3,930
Titles	2,219
· Administration	427
· Epithet	289
· Family	163
· Priest	348
· Privy	117
· Rank	94
· Worker	125
· Unclassified	656
Person-title relations	13,695
Family relations	10,335
Tombs	903
Cult chambers	347
False doors	397
Shafts	5,305

Table 1.1: Overview of the size of Maatbase database. The left rows represent the feature type, and the right column number of the corresponding records. The numbers correspond to the database version from October 11, 2024.

With this overview, we have introduced the available information that can be used in our later modeling. It is evident that the database contains a substantial number of potential features. However, it is clear that not all of these features can be used in a single model, otherwise we would not be able to train the model properly due to overfitting. Furthermore, certain features had to be omitted due to their sparsity, being available only for a limited number of persons. During feature selection, we should incorporate as much prior knowledge as possible. Hence, for the remainder of this chapter, we will focus on similar studies.

1.3 Similar Studies

At this point, we have become familiar with the Old Kingdom environment and the Maatbase database. Since we aim to analyze its society from a mathematical perspective, let us also introduce similar studies related to this task. First, we will focus again on the Old Kingdom and the research that has already been conducted using the Maatbase database. Later, we will review statistical analyses of societies in different times and places.

The following research will outline methods and results that have been used in the analysis of multiple social networks. As we will see, there is no standardized technique for conducting such analyses. Methods are chosen and sometimes even modified according to the specific task and dataset. We will mention approaches such as linear regression, various graphical tools, Hidden Markov Models, information theory, and self-defined scores that measure specific phenomena.

Through this research, the author became familiar with methods used in the study of social networks and developed the ability to interpret and discuss the resulting findings. It also became apparent that the applied methods must be understandable to Egyptologists. Therefore, this thesis emphasizes more standard and interpretable approaches.

1.3.1 The Old Kingdom Society

The Old Kingdom has been analyzed using the Maatbase database several times. Many articles have been produced through the collaboration of Mgr. et Mgr. Veronika Dulíková, Ph.D.[2] and Ing. Radek Mařík, CSc.[3]. This subsection summarizes their publications. To the best knowledge of the author of this thesis and the aforementioned scholars [6], there are no other similar studies in the field of Ancient Egypt or in any other historical field.

Through this research, the author became familiar with previously discovered patterns within the Maatbase database, specifically the level of nepotism in society, periods of significant changes in vizierial titulary, and titles closely related to the vizier's office. Since many of these publications are accompanied by commentary from Egyptologists, the author also deepened his understanding of Ancient Egypt.

In particular, this research provides insight into which features are potentially crucial for the later models and how different methods have already been applied. This knowledge will be incorporated into the later modeling process.

Although the publication by [6] provides a quick overview of many applied approaches, the literature mentioned below offers a much more detailed explanation of the methods and a discussion of the results.

[2] Czech Institute of Egyptology, Faculty of Arts, Charles University

[3] Department of Telecommunications Engineering, Faculty of Electrical Engineering, CTU in Prague

Identifying Nepotism Among the Old Kingdom Society

This article [7][4] studies both nepotism among non-royal families and the pharaoh's strategy of marrying his daughters to non-royal members.

To measure the rate of nepotism within each family, the authors define their own mathematical approach based on the count of title occurrences within a particular family and the overall population. Later, they highlight certain families exhibiting nepotistic behavior.

In addition, they support their findings by presenting several family trees along with the titulary of each family member. Specifically, they show families that served as scribes, singers, the king's hairdressers, or were involved in weaving or the treasury department. These trees reveal an obvious trend of inheriting titles from parents to their descendants.

What is more interesting for this thesis is that the authors point out two vizier families with a tendency toward nepotism. In one of these families, the vizier title was inherited not only from father to son, but also from son to grandson.

This article reveals that nepotism was indeed very common across all offices. Therefore, in our later models, we should include a feature that captures family relations to particular titles, such as whether an ancestor held the vizier title.

Clustering of Titles with Help of Information Theory

The main goal of the article [8][5] is the clustering of titles. Using directed graphs and information theory, the author searched for patterns in titulary that might reveal hierarchies, clans, and societal groups.

First, the author defines an undirected graph in which each node represents a particular title, and all nodes are connected. The strength of the connection between any two nodes is determined by mutual information.

Subsequently, he creates a directed graph by preserving only the strongest connections for each node. Additionally, he enhances this graph with edges with mutual information above a given threshold.

Since the final graph connects titles with the highest mutual information, that is, strong similarities, it is used for cluster detection. Using this approach, four major clusters are identified. The largest cluster consists of the highest-ranked dignitaries, including viziers.

The article concludes by identifying a community of 34 individuals associated with

[4] V. Dulíková, R. Mařík, *Complex network analysis in Old Kingdom society: a nepotism case.* In 'Abusir and Saqqara in the Year 2015'. Charles University Faculty of Arts, 2017, 63–83.

[5] R. Marik, *Feature Space Decomposition using Information Theory.* In 'Proceedings of the 2018 International Conference on Algorithms, Computing and Artificial Intelligence'. Association for Computing Machinery, 2018, 1–10.

the title ‘keeper of the king’s headdress’. This community is analyzed using information variance and conditional entropy. Through these measures, the author identifies three sub-communities. The first consists of highly ranked individuals, such as overseers and noblemen, while the third is primarily a cluster of hairdressers.

Graphical Methods and Vizier-Scores as Tools for Dignitary Detection

One of the most important publications in terms of relevance to this thesis is [4][6], where the authors investigate the patterns behind the vizier title.

In the first part, the authors use a bipartite graph to visualize patterns among individuals and titles. A bipartite graph is a graph with two distinct sets of nodes. In this study, one set of nodes represents individuals, and the other set represents titles. In this graph, each individual is connected by edges to the titles he has achieved according to the Maatbase database.

Using the Fruchterman-Reingold method, this graph is visualized in two dimensions, with the cluster of viziers highlighted. To briefly review this method, it is based on attractive and repulsive forces between nodes according to their distance. The method iterates until it finds an equilibrium in which all nodes remain stationary.

In the second part, the authors define the titles that most often accompany the title of vizier. Using an uncommon but intuitive set of measures, they identify non-vizier individuals who exhibit a similar titulary to that of actual viziers.

All these measures are based on the counts of given titles within the vizier and non-vizier groups. Non-vizier individuals with the highest scores represent those who achieved significant influence in the Old Kingdom, even though there is no evidence that they were viziers.

Subsequently, these individuals are highlighted in the aforementioned bipartite graph, and all are located near the vizier cluster. Additionally, several of these individuals are discussed along with their family trees, where once again strong nepotistic behavior is evident.

At the end of the article, there are tables listing many such influential individuals.

Hidden Markov Models in the Services of the Old Kingdom

This publication [2][7] covers several data-based analyses of the Old Kingdom, together with many historical reviews. Similarly to the previously mentioned literature, this publication also examines relationships in title occurrences using graphi-

[6] V. Dulikova, R. Marik, *Uncovering Old Kingdom society arrangement: Detection of powerful dignitaries using complex network analysis.* In ’Handbook of Digital Egyptology: Texts’. Editorial Universidad de Alcalá, 2021, 69–102.

[7] M. Bárta, V. Dulíková, R. Mařík, *et al.*, *Modelling the Dynamics of Ancient Egyptian State During the Old Kingdom Period: Hidden Markov Models and Social Network Analysis.* Zeitschrift für Ägyptische Sprache und Altertumskunde 149, 2022, 1–16.

cal methods and explores the influence of nepotism. What has not been mentioned in this text so far is the use of Hidden Markov Models (HMM), probably the most sophisticated method applied to the Maatbase database.

HMM is an appropriate tool for sequence modeling. To create such a sequence, the authors chronologically ordered 74 viziers. These chronologically ordered individuals represented time steps in the sequence. Additionally, they selected 48 of the most significant titles related to the vizier's office. Each title represented one sequence, which was filled with zeros and ones, indicating whether a given individual had a particular title.

Since these sequences are only 74 observations long, it is not sufficient to use HMM in its standard form. The authors proposed their own method to overcome this issue. A more detailed description of this model was published in [9]. In summary, they initialized 1,000 HMM models. Then, using a Gaussian Mixture Model, they selected the 687 best ones. Finally, by analyzing the average changes in hidden states, they identified periods when certain titles either began to appear or, conversely, started to disappear.

Using this approach, the authors determined periods of significant changes in administrative titles. In addition, they presented plots showing the epochs of changes detected by HMM alongside the epochs identified by Egyptologists. Briefly, these epochs correspond well with each other, with a detailed discussion provided in [10]. Additionally, for each period, they highlighted titles that had a high probability of either disappearing or beginning to appear in the vizierial titulary.

In summary, the articles [2], [9], and [10] present an HMM model that reveals that the evolution of titles closely related to the vizier's office was dynamic, rather than characterized by a single significant trend. Some titles disappeared while others emerged, all happening in multiple different periods. This leads to the conclusion that this evolution is strongly non-linear. Therefore, when attempting to capture this non-linear pattern, a neural network should have a notable advantage over more classical approaches like linear regression.

1.3.2 Studies from Different Temporal and Spatial Contexts

We will conclude this chapter with a review of similar studies on societies from different temporal and spatial contexts. Although the author did not find any study directly related to his task, he mentions a few articles below that examine particular social phenomena using mathematically-based approaches.

In the following text, we will introduce specific phenomena, along with a description of the applied methods and the found results.

Social Mobility in Florence from 1282 to 1494

Medieval Florence was an important trade center, although it underwent many changes, including revolts, takeovers, and the Black Death. This turmoil impacted its social structure, which exhibited endogamous behavior. This means that members of high-status families could only marry within their own class. This practice, along with wealth and political power, was analyzed in the article [11][8].

The author collected data from various sources, such as tax censuses, political offices, and marriage records. First, he analyzed the evolution and stability of families over time based on wealth and political power.

Later, he prepared two statistical models. Using classical regression, he identified the main factors that influenced the wealth growth of Florentine families. In addition, he demonstrated that families with strong connections to other families were more likely to grow rather than decline.

The second model was based on Zero-Inflated Poisson regression[9]. Using this approach, he modeled the counts of marriages among different social classes. The model showed that members of the wealthiest families tended to marry each other, which confirms the expectation of endogamy.

Alliance Network in Mafia Families

A more recent study comes from the authors of [13][10]. The authors collected data on the 'Ndrangheta mafia clan, which originates from Calabria, Italy. Similarly to previous cases, this society also exhibits unusual marriage behavior, though for different reasons.

Since these illegal organizations cannot rely on conventional rules and institutions, they exploit marriages as a tool to generate and preserve trust. They use exogamous marriages, that is, unions between members of distinct mafia families, to seal alliances, enter new businesses, expand influence, and resolve conflicts. On the other hand, they use endogamous marriages to prevent business fragmentation and strengthen internal cohesion.

To examine these strategies, the authors collected data from two sources. Namely, family trees of the most important families and judicial documents related to various criminal investigations against the 'Ndrangheta clan between 2006 and 2016.

Using graphical tools and various metrics, the authors analyzed the structure of marriages among families. They were able to identify cohesive subgroups and examine the strategies within them.

[8] J. F. Padgett, *Open Elite? Social Mobility, Marriage, and Family in Florence, 1282–1494*. Renaissance Quarterly 63, 2010, 357–411.

[9] Zero-Inflated Poisson regression is used to model counts with a high number of zeros [12].

[10] M. Catino, S. Rocchi, G. Vittucci Marzetti, *The network of interfamily marriages in 'Ndrangheta*. Social Networks 68, 2022, 318–329.

In summary, they found that different families tend to adopt different matrimonial strategies. Some families prefer to keep their men for endogamous marriages while allowing their women for exogamous unions. On the other hand, there are families that do not exhibit endogamous behavior at all.

Chapter 2

Essentials of Machine Learning Theory

This section summarized several machine learning methods that will be used in later modeling. This chapter is ordered as follows. First, a theoretical background of principal component analysis is introduced. This section is concluded with an emphasis on the importance of feature scaling. The second section is about logistic regression, its properties and hypothesis testing. At the end of this section we mention regularizations and information criteria which will be used in later models.

The third section introduces neural network models. It summarizes several activation functions and its properties together with loss functions and common optimizers. Again, this section is concluded with regularization methods which will be crucial in later modeling. The fourth section introduces SHAP values that will allow us to interpret neural networks. Finally, the last section describes several scores that will be used later in the model study.

2.1 Principal Component Analysis

The Principal Component Analysis (PCA) is the well-known and described algorithm to reduce the dimension of a dataset. The emergency of dimensionality reduction can arise in many tasks such as linear regression and learning an artificial neural network. One of the main reasons for using PCA is to overcome multicollinearity. Multicollinearity can affect the uniqueness and numerical stability of many models [14], [15]. Furthermore, by reducing dimensionality of the data one can also achieve a significant reduction of parameters in a model. Hence, the model is much easier to train.

In the following section, the principle of PCA is briefly reviewed. Subsequently, the theoretical part is concluded by several examples on real Egyptologist data. The theoretical derivation is mainly based on the literature [16], [17], whereas the given examples represent the author's own view on the algorithm.

Theoretical Background of PCA

Let's have a dataset, which can be represented by matrix $\mathbf{D} \in \mathbb{R}^{n \times p}$, where n, resp. p is number of observation, resp. number of features. Imagine that p is too high and our goal is to find a matrix $\mathbf{R} \in \mathbb{R}^{n \times k}$, which carries as much information as matrix $\mathbf{D}$, but the dimension k is much smaller than p.

PCA solves this problem by looking at which directions the data have the highest variance. Since most of the data-variance is hidden in these directions, they are called principal components. To be more precise, the first principal component is defined as the unit-length vector $\mathbf{w}_1$ in the feature space $\mathbb{R}^p$ that scores the highest variance among all other unit-length vectors. Mathematically, if we imagine that rows of matrix $\mathbf{D}$ are samples from some unknown random variable $\mathbf{X}$, we are looking for vector $\mathbf{w}_1$ such fulfills

$$\max_{\substack{\mathbf{w}_1 \\ \mathbf{w}_1^T \mathbf{w}_1 = 1}} \operatorname{Var}(\mathbf{w}_1^T \mathbf{X}), \tag{2.1}$$

where $\operatorname{Var}(\mathbf{w}_1^T \mathbf{x})$ symbolizes empirical variance among projections of $\mathbf{X}$ onto vector $\mathbf{w}_1$ and the maximum is taken among all unit-length vectors in feature space $\mathbb{R}^p$.

The above optimization problem can be solved by Lagrangian function, which is in this case given as

$$\sum_{i=1}^{n} \left[\mathbf{w}_1^T \mathbf{x}_i - \mathbb{E}\left(\mathbf{w}_1 \mathbf{x}\right)\right]^2 - \lambda \left(\mathbf{w}_1^T \mathbf{w}_1 - 1\right), \tag{2.2}$$

where $\mathbf{x}_i$ is i-th row of matrix $\mathbf{D}$, $\mathbb{E}$ is empirical expected value and λ is Lagrange multiplier. By simple algebraic operation one can rewrite the Lagrangian function into very convenient form

$$\mathbf{w}_1^T \mathbf{\Sigma} \mathbf{w}_1 - \lambda(\mathbf{w}_1^T \mathbf{w}_1 - 1), \tag{2.3}$$

where $\mathbf{\Sigma}$ is empirical covariance matrix among columns (features) of matrix $\mathbf{D}$. Finally, by differentiation one can find that the desired $\mathbf{w}_1$ must satisfy

$$\left(\mathbf{\Sigma} - \lambda \mathbf{I}_p\right) \mathbf{w}_1 = \mathbf{0} \tag{2.4}$$

Therefore, looking for the first principal component $\mathbf{w}_1$ of PCA is equivalent to searching for the eigenvector of matrix $\mathbf{\Sigma}$ corresponding to the highest eigenvalue λ_1. Furthermore, this eigenvalue is equal to the maximum given in Equation (2.1).

The second and further principal components $\mathbf{w}_2, \mathbf{w}_3, \ldots, \mathbf{w}_k$ can be found in a similar way. For $l \in [2, 3, \ldots, p]$ vector $\mathbf{v}_l$ must fulfill equation (2.1) with $l-1$ additional restrictions, it is $\mathbf{w}_l$ must be orthonormal to $\mathbf{w}_{l-1}, \mathbf{w}_{l-2}, \ldots, \mathbf{w}_1$. This task is again equivalent to searching for eigenvectors corresponding to the highest eigenvalues $\lambda_2, \lambda_3, \ldots, \lambda_k$.

Once we have found first k components, we can squeeze the p dimensional feature space into k dimensions. The corresponding transformation is given as

$$\mathbf{R} = \widetilde{\mathbf{D}} \mathbf{W}, \tag{2.5}$$

where $\widetilde{\mathbf{D}}$ is matrix $\mathbf{D}$ with columns transformed to have zero mean. $\mathbf{W} \in \mathbb{R}^{p \times k}$ is matrix whose columns are vectors $\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_k$. In this notation, we can see that the element $\mathbf{R}_{i,j}$ is nothing more than a projection of the i-th row of $\mathbf{D}$ onto the eigenvector $\mathbf{w_j}$.

Demonstration of PCA on Egyptologist Data

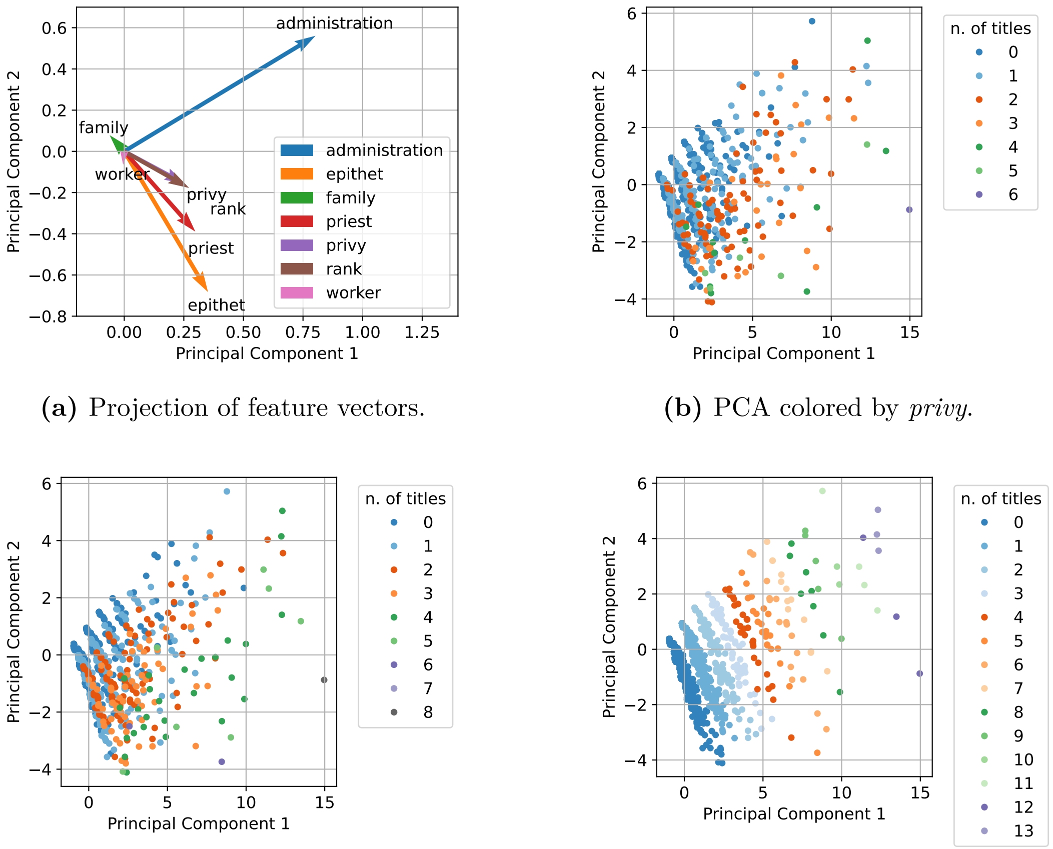
With above described theory we can deep into the PCA algorithm from a practical point of view. In addition, the used notation will give us more insight into the phenomena shown in the following.

First of all, let us introduce our data matrix $\mathbf{D} \in \mathbb{R}^{3977 \times 7}$, whose few rows are shown in Table 2.1. These rows present a number of titles in selected categories for each individual. In addition to the elements of the matrix, the table also contains an estimate of the mean value and standard deviation for each feature (column). Since PCA looks for the direction with the highest variance, looking at Table 2.1 one could guess that columns with a higher standard deviation will be preferred in PCA. Indeed, from Figure 2.1d column *administration* is clearly much more easily recognizable within the first two components of PCA than any other feature owing to its high variance, see once more 2.1.

On the other hand, PCA is often part of a data reprocessing before some modeling, and since high variance is not necessarily connected to the model outcome, the model could potentially lose some crucial information in the prepossessing part. To present this phenomenon, let us see Table 2.1 again. Feature *privy* has quite low standard deviation (see Table 2.1) and therefore is not significantly projected on the main components. Therefore, by blindly applying PCA to the matrix, the model could lose a significant portion of the information. To overcome this situation, the author will propose the *The Scaling Effect of PCA* to increase the projection of important features into the first components of the PCA algorithm.

ID	**administration**	**epithet**	**family**	**priest**	**privy**	**rank**	**worker**
1	7	3	0	4	4	4	1
2	1	1	1	1	1	2	0
4	0	2	0	4	4	1	2
...	...	...	...	...	...	...	...
4738	1	0	0	0	0	0	0
4739	1	0	0	2	0	0	0
mean	0.528	0.412	0.317	0.564	0.161	0.475	0.122
std	1.292	0.858	0.465	0.784	0.563	0.692	0.346

Table 2.1: Table showing the dataset of titles (aggregated into seven groups). Titles are labeled in the top row together with person *ID*, which works as a unique identifier for each person. The elements in the table represent the number of titles that each person has gained. For example, a person with *ID* = 1 gained in total 7 administration titles whereas a person *ID* = 4 zero. At the bottom there are calculated means and standard deviations (*std*) for each title (group).



(a) Projection of feature vectors.

(b) PCA colored by *privy*.

(c) PCA colored by *priest*.

(d) PCA colored by *administration*.

Figure 2.1: Figures show dimension reduction via PCA. Figure (a) shows the projection of basis vectors onto the first two principal components. Figures (b), (c) and (d) show the same data transformation into two dimensions. They only differ in color-coding of titles (*privy*, *priest* and *administration*). See the corresponding legend for more details.

The Scaling Effect of PCA

To force the PCA to encode the given feature in its main components, one can appropriately scale the matrix. Since PCA is looking for direction in feature space with highest variance, by scaling a given feature one can increase variance in this direction and consequently increase projection of this feature into main components. This phenomenon is shown in Figure 2.2. In the Figures 2.2a original matrix is shown. In this case, group *administration* is much more recognizable than title *privy*. However, by scaling group *privy* by integers 2 and 4 (i.e., columns *privy* times given number), PCA starts to prioritize *privy* over all other features. See Figure 2.2 in full details to explore this effect.

From a mathematical point of view, by scaling *privy* column, both column and row of matrix $\mathbf{\Sigma}$ corresponding to this feature are also scaled with this number. Hence, the solution of the eigenvalue problem given by equation (2.4) can change dramatically. To depict this change, Table 2.2 shows all eigenvalues of original matrix $\mathbf{\Sigma}$ and its modifications with scaled *privy*. One can see that the first principal component is acquiring much more variance, whereas the smallest principal components are almost unchanged. To emphasize this fact, one can also look into Table 2.3 which is the same as above table but the eigenvalues are now scaled to the highest one.

scaled	λ_1	λ_2	λ_3	λ_4	λ_5	λ_6	λ_7
by 1	86.6	47.1	38.5	32.1	26.1	23.9	18.5
by 2	96.6	50.4	40.1	38.1	32.1	25.8	18.9
by 3	116.5	57.2	43.6	38.4	32.1	25.8	19.0
by 4	142.9	61.1	44.3	38.4	32.1	25.8	19.0

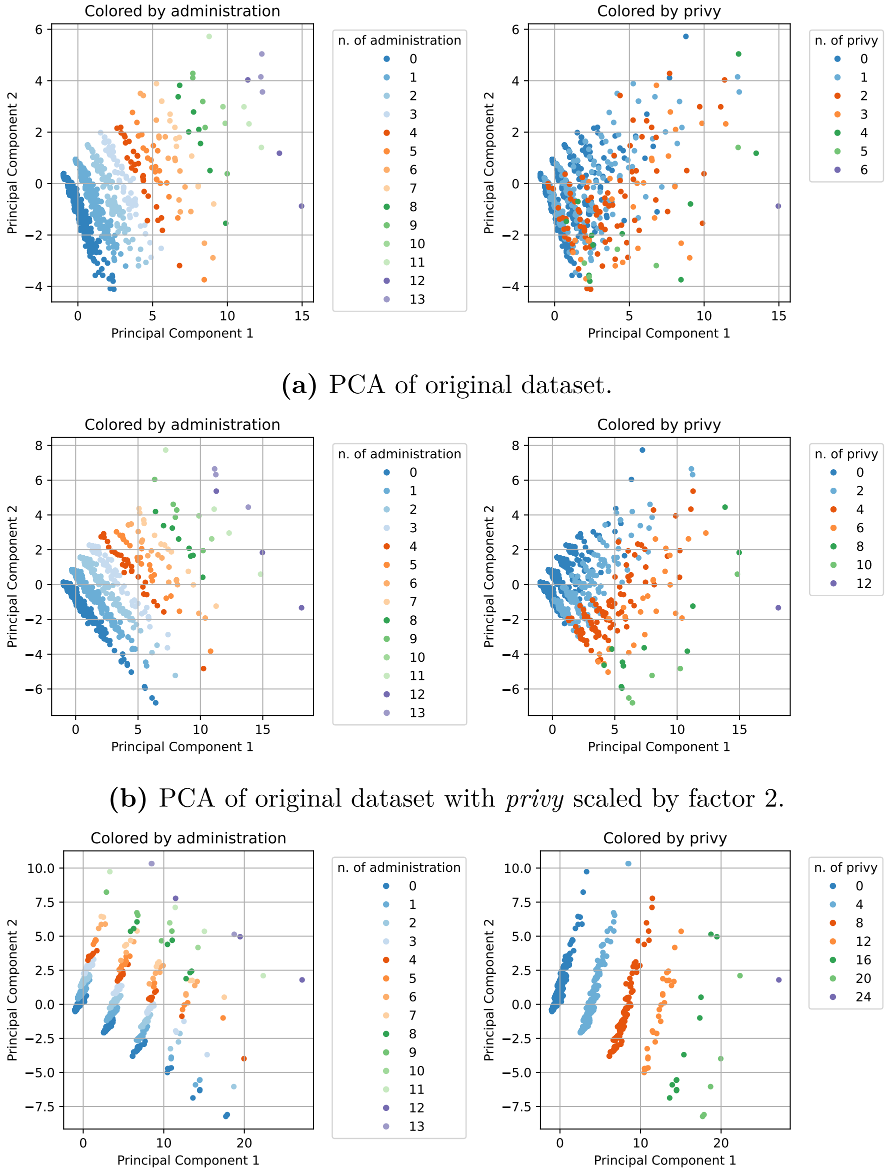
Table 2.2: Table showing all eigenvalues from equation (2.4) for the original data and group *privy* scaled by 2, 3 and 4. We see that the highest eigenvalue λ_1 is increasing significantly while λ_7 is almost uncharged.

scaled	λ_1	λ_2	λ_3	λ_4	λ_5	λ_6	λ_7
by 1	1.00	0.54	0.44	0.37	0.30	0.28	0.21
by 2	1.00	0.52	0.42	0.39	0.33	0.27	0.20
by 3	1.00	0.49	0.37	0.33	0.28	0.22	0.16
by 4	1.00	0.43	0.31	0.27	0.22	0.18	0.13

Table 2.3: The table is the same as 2.2 (see its caption) but the eigenvalues are now normalized to λ_1. We can see that by scaling λ_1 is becoming more and more dominant.

Normalization Before PCA

While speaking about prioritizing given feature over other, one may also ask yourself what happens if we normalize data to have zero-mean and variance equal to one before PCA, i.e. equalize features in the point of PCA view. Figure 2.3 is presented to answer this question. Looking at Figure 2.3b that depicts projection of normalized features, we see that all projections have similar lengths and are more uniformly distributed in the graph than projections of the original 2.3c and scaled *privy* group (rest of the Figures in 2.3)).

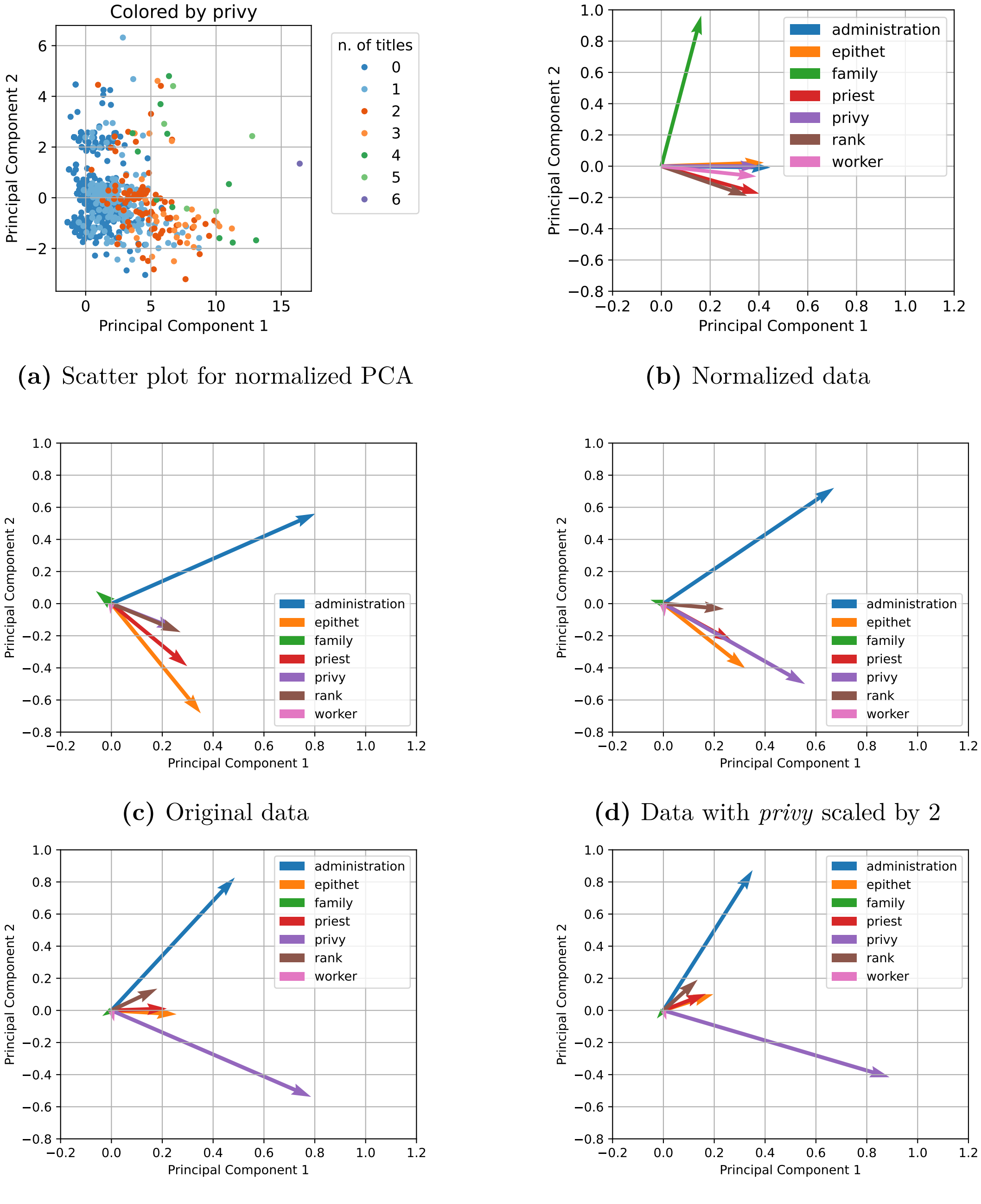


(a) PCA of original dataset.

(b) PCA of original dataset with *privy* scaled by factor 2.

(c) PCA of original dataset with *privy* scaled by factor 4.

Figure 2.2: The figures show effect of scaling the feature *privy* while keeping other features untouched. Plot (a) shows projection onto first two principal components for non-scaled data. Left-side plots shows color-coding of group *administration*, whereas right-side plots depict *privy*. Plots (b), and (c) show dataset, where the group *privy* was scaled by 2, and 4, respectively. As can be seen, it led to the dominance of *privy* over all other features in the PCA transformation.



(a) Scatter plot for normalized PCA

(b) Normalized data

(c) Original data

(d) Data with *privy* scaled by 2

(e) Data with *privy* scaled by 3

(f) Data with *privy* scaled by 4

Figure 2.3: The upper left plot (a) shows the projection of the normalized (zero-mean and variance equal to one for each feature) dataset into the first two principal components. The plot (b) depicts projection of basis vector onto the first two principal components, in this case PCA was applied on normalized dataset. Similarly, plots (c), (d), (e), and (f) show projection into the first principal components for original datasets and dataset with group *privy* scaled by 2, and 4, respectively.

Conclusion of Given PCA Examples

By the above practical example, the author wanted to present a impact of data reprocessing before applying PCA itself. To avoid suppression of low-variance columns, one could use the advantages of normalization of the matrix, which equalizes the features from the PCA perspective. Furthermore, one could also scale the preferred features to have a higher variance and therefore be less suppressed in the PCA algorithm.

2.2 Logistic Regression

This section is dedicated to Logistic Regression (LR), which will be used in several later models. We will mention its background, statistical properties, and criteria of feature selection. Since there are many books dedicated to this topic [14], [18], [19] and [20], this section will briefly summarize them, mainly highlighting these topics related to this thesis.

Logistic regression belongs to generalized linear regression. Generalized models extend standard linear regression in two ways. Firstly, it does not restrict the response variable to be of normal distribution but from the whole exponential family. Secondly, it introduces the link function which allows one to model a nonlinear dependency between the response variable and the predictors. Logistic regression is an appropriate tool to study the binary response. In this case, we consider the response to be a Bernoulli random variable, respectively, binomial distribution, which are both members of the exponential family.

Assume that we have independent random variables Y_1, Y_2, $\ldots$, Y_n. Each from binomial distribution, i.e., for each $i \in \{1, 2, \ldots, n\}$ holds $Y_i \sim Bi\,(n_i, \pi_i)$. Then the general form of binomial regression is given as

$$\mathbb{E}(Y_i) = g^{-1}(\eta_i) \tag{2.6}$$

$$\eta_i = \beta_1 x_{i1} + \beta_2 x_{i2} + \cdots + \beta_p x_{ip} \tag{2.7}$$

where η_i is mean of random variable Y_i. In addition, $x_{i1}, x_{i2}, \ldots, x_{ip}$ are predictors of the i-th covariate pattern. Factors β are coefficients, the task of binomial regression is to optimize these coefficients. The function $g : [0, 1] \to \mathbb{R}$ is a link function. Since the task of binomial regression is to predict probability, the link function must be chosen to have a domain between 0 and 1. The common link function is chosen as the logit function. This function is given as

$$\text{logit}(\pi) = \ln \frac{\pi}{1 - \pi}. \tag{2.8}$$

The binomial regression model with logit link function is called logistic regression. There are also other link functions such probit

$$\text{probit}(\pi) = \Phi^{-1}(\pi), \tag{2.9}$$

where Φ is cumulative distribution function of the standard normal distribution. Another common link function is complementary log-log

$$\text{cloglog}(\pi) = \ln(-\ln(1-\pi)). \tag{2.10}$$

These functions are depicted in Figure 2.4.

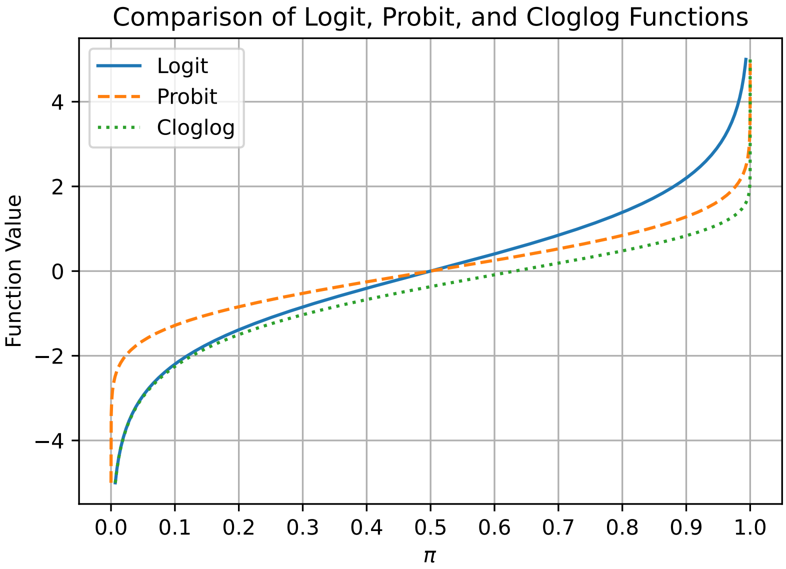


Figure 2.4: Graph of different link functions used for binomial regression. Namely, logit (solid blue), probit (dashed orange), and complementary log-log (dotted green).

The optimal coefficients β are found by means of Maximum Likelihood Estimation (MLE). Since the MLE does not have analytical solution, the solution is found by means of iterative algorithms such Newton-Raphson, Fisher scoring or iteratively reweighted least squares [14], [18]. If the model assumptions are correct, it asymptotically holds [14]

$$\mathbb{E}(\hat{\beta}) \dot{\sim} \beta, \tag{2.11}$$

$$\text{Var}(\hat{\beta}) \dot{\sim} \left(\mathbf{X}^T\mathbf{V}\mathbf{X}\right)^{-1}, \tag{2.12}$$

where $\hat{\beta}$ are fitted coefficients, β are true coefficients. The matrix $\mathbf{X} \in \mathbb{R}^{n\times p}$ is the design matrix of n observation and p features. The diagonal of matrix $\mathbf{V} \in \mathbb{R}^{n\times n}$ contains empirical variance of each observation. Symbol $\dot{\sim}$ refers to asymptotical convergence.

2.2.1 Statistical Inference

Statistical inference for the logistic model is based on the properties of MLE and the likelihood ratio test (LRT). Although these tests hold only asymptotically, they offer a powerful instrument for deeper model analysis. LRT tests offer model comparison based on maximum likelihood from two different models, these models must be nested, that is, one model must be a reduced version of the other [14]. Beloved mentioned testing goodness of fit and deviance statistic belong into LRT. Alternatively to these tests, Wald test offers analysis whether one particular coefficient is non-zero and computing confidence interval of such coefficient.

Testing Goodness of Fit

This test provides insight into how well the full model (FM) capture data compare to the saturated model. The saturated model is a model where each covariate pattern has its own parameter, i.e., its own success probability. By this test, we can see whether our FM model describes model sufficiently or whether we should add another variable into FM model.

The goodness of fit is based on likelihood ratio between these two models

$$\begin{aligned} D &= 2 \ln \frac{L(\text{ saturated model })}{L(FM)} \\ &= 2 \sum_{i=1}^{n} \left[y_i \ln \left(\frac{y_i}{n_i \hat{\pi}_i} \right) + (n_i - y_i) \ln \left(\frac{n_i - y_i}{n_i \left(1 - \hat{\pi}_i\right)} \right) \right], \end{aligned} \tag{2.13}$$

where L denotes maximum-likelihood of particular model. This test has a limitation, for saturated model with greater number of observations we need also greater number of parameters. Therefore, large-sample asymptotics are not reliable here [19]. However, if for all covariate patterns y_i hold $\min(y_i) > 2$ and $\min(n_i - y_i) > 2$ then this statistic approximately follows the chi-square distribution with $n - p$ degrees of freedom [20].

Partial Deviance

Deviance statistic allows testing whether a subset of model coefficients are zero. Mathematically, having the full model with estimated coefficients $\hat{\beta} = (\hat{\beta}_1, \hat{\beta}_2)$, we want to test hypothesis

$$H_0 : \beta_2 = \mathbf{0}, \quad H_1 : \beta_2 \neq \mathbf{0}, \tag{2.14}$$

where β_1 and β_2 contains $p - r$ and r parameters respectively.

This test is based on the comparison of the deviances of two models. The first deviance $D(\beta)$ belongs to the full model which is given as $\eta = \mathbf{X}\beta$. The second deviance $D\left(\beta_1\right)$ belongs to a reduced model which is given as $\eta_1 = \mathbf{X}_1\beta_1$, where matrix $\mathbf{X}_1$ is associated with coefficients β_1.

If the null hypothesis is true and n is large, then partial deviance

$$D\left(\beta_2 \mid \beta_1\right) = D\left(\beta_1\right) - D(\beta) \tag{2.15}$$

has chi-square distribution with r degrees of freedom [20].

Wald Inference

Wald inference allows test individual coefficients, the tested hypothesis is

$$H_0 : \beta_j = 0, \quad H_1 : \beta_j \neq 0. \tag{2.16}$$

The test statistic is given as

$$Z_0 = \frac{\hat{\beta}_j}{\operatorname{se}\left(\hat{\beta}_j\right)}, \tag{2.17}$$

where $\operatorname{se}\left(\hat{\beta}_j\right)$ refers to estimated standard error of coefficient β_j. This statistic follows asymptotically the standard normal distribution. Wald inference can also be used to construct a confidence interval for individual regression coefficients.

2.2.2 Non-Nested Models Comparison

The above subsection introduced the possible comparison of nested models. This section will briefly introduce two information criteria that offer a comparison of non-nested models. Since these criteria are not based on any statistical test, they assign only a score to each model. According to the convention given below, the model with the lower score is considered to be better.

AIC

Akaike Information Criterion (AIC) is based on log-likelihood. Having a model M, its AIC is given as [21]

$$\mathrm{AIC}(M) = -2l(M) + 2p, \tag{2.18}$$

where $l(M)$ is log-likelihood of model M and p is number of its features. Since adding more parameters into a model always decreases the AIC, the second term in (2.18) presents a penalization for the number of parameters. Hence, the AIC balances between maximum-likelihood and model complexity.

AIC is closely related to the Kullback–Leibler (KL) divergence. The aim is to minimize the KL divergence between the unknown data-generating distribution and the distribution given by the model. It can be shown that the penalization term corrects the bias of such minimization [22].

BIC

Bayesian Information Criterion (BIC) is defined as [22]

$$\mathrm{BIC}(M) = -2l(M) + \ln(n)p, \tag{2.19}$$

where n is the sample size. BIC can be derived using Bayesian analysis [22], hence its name. Compared to AIC, BIC penalizes model complexity more strongly[1]. Therefore, BIC tends to models with fewer parameters.

[1] If n $\geq$7.

2.2.3 Regularization and Feature Selection Techniques

High number of predictors causes several problems, such as the risk of model overfitting or the presence of multicollinearity. Multicollinearity can cause training instability and produce unrealistically high variance in predictors. To overcome such issues, predictors are reduced to a smaller subset. There are several common approaches, such as LASSO and Ridge regularization or model selection algorithms based on AIC or BIC.

Ridge

Ridge regression, also known as L_2 regularization, modifies maximum likelihood by penalizing the size of the coefficients. The Ridge solution is given as [23]

$$\min_{\tilde{\beta}} \left(\frac{1}{n} l(M) + \lambda \sum_{j=1}^{p} \tilde{\beta}_j^2 \right), \tag{2.20}$$

where $l(M)$ is the log-likelihood of model M given by the coefficients $\tilde{\beta}$. The hyperparameter λ defines the penalty strength. The intercept is omitted from the penalty term. Moreover, Ridge regularization is not equivariant to scaling. Hence, it is common to standardize the inputs before applying Ridge [24].

The effect of Ridge penalization can be demonstrated on the classical linear regression model. If such a model has many correlated predictors, then these predictors can become poorly determined. This scenario can be identified by unusually large coefficients with opposite signs. It can be shown that Ridge penalizes most columns of $\mathbf{X}$ that correspond to minimal variance in the SVD decomposition [24].

LASSO

LASSO regularization is similar to Ridge, but in this case we penalize the L^1 norm of the coefficients. The modified likelihood is given as [23]

$$\min_{\tilde{\beta}} \left(\frac{1}{n} l(M) + \lambda \sum_{j=1}^{p} \left| \tilde{\beta}_j \right| \right). \tag{2.21}$$

Whereas Ridge regularization shrinks coefficients toward zero but almost never exactly to zero, LASSO, on the other hand, can shrink some coefficients exactly to zero by appropriately setting λ [24].

Regularization Based on Information Criterion

An appropriate subset of model predictors can also be chosen by means of AIC or BIC. There are two core approaches. The first, forward stepwise selection, starts with a model containing only the intercept. The predictors are then added to the model one by one, selecting at each step the one that improves the AIC or BIC most [24].

The second approach, backward stepwise selection, starts with a model containing all predictors and sequentially removes the predictor that has the least impact on the chosen criterion [24]. There is also a stepwise selection strategy that combines both of the above-mentioned algorithms. In each step, this strategy either adds or removes the predictor that minimizes AIC or BIC [25].

Variance Inflation Factor

Variance Inflation Factor (VIF) is a method to analyze multicollinearity in the data. The VIF value of the j-th regressor is defined as

$$\mathrm{VIF}_j = \frac{1}{1 - R_j^2}, \tag{2.22}$$

where R_j^2 is coefficient of determination obtained from regressing j-th regressor on the other regressors [14]. VIF values above 10 indicates serious problems [14].

2.3 Multilayer Perceptron

This section aims to introduce the Multilayer Perceptron (MLP) model. The term multilayer perceptron refers to an neural network Artificial Neural Network (NN) model, which is sometimes also called a deep feedforward network or a feedforward neural network [26]. The word feedforward is used to highlight the fact that input information propagates through the model without any feedback connections. Networks with feedback connections are referred to as recurrent neural networks.

Given a set of input features $\mathbf{x}$ and corresponding target values $\mathbf{y}$, the goal of an MLP model is to learn a function f that best approximates these targets. In our case, it will be a binary classifier that maps input features $\mathbf{x}$ to a numeric value between zero and one. An MLP consists of multiple layers, each composed of neurons. Each neuron first performs a linear transformation on its input, followed by the application of a non-linear activation function. The task of training an MLP is to fit the parameters that control these linear transformations.

In an MLP, there are three types of layers. The input layer refers to the very first layer, which contains only the input features and no activation function. The last layer is called the output layer. Any layer between the input and output layers is called a hidden layer.

An example of an MLP architecture with two hidden layers is shown in Figure 2.5. Gray circles represent neurons. Each neuron in one layer is fully connected to every neuron in the next layer. These connections are represented as edges with arrows, indicating the direction of data flow.

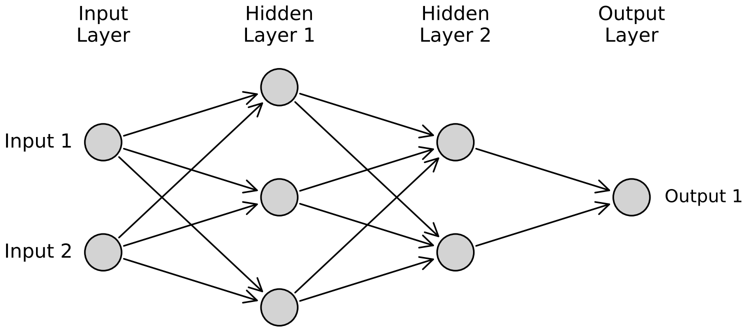


Figure 2.5: An example of an MLP architecture.

2.3.1 Activation Functions

The presence of a non-linear activation function in hidden layers is crucial. Without them, the hidden layers would only be a sequence of several linear transformations. This linear sequence can be replaced by a single linear transformation[2], so the whole model is linear. There are several commonly used activation functions.

ReLU: The recommendation is to use ReLU as the default option [26]–[28]. The name ReLU stands for Rectified Linear Unit. This function is defined as

$$\text{ReLU}(x) = \begin{cases} x & x > 0, \\ 0 & x \leq 0. \end{cases} \tag{2.23}$$

It has a derivative of exactly 1 for values $x > 0$, and exactly 0 for values $x < 0$. Hence, it is similar to linear units and is easy to optimize.

Leaky ReLU: A drawback of ReLU is that it has a zero derivative on half of its domain. Hence, gradient-based methods cannot learn in this part of the domain. Leaky ReLU modifies this part of the domain, it is defined as

$$\text{LeakyReLU}(x) = \begin{cases} x & x > 0, \\ \alpha x & x \leq 0, \end{cases} \tag{2.24}$$

with α typically set to around 0.01 [29].

Sigmoid: This function is also known as the logistic function

$$\sigma(x) = \frac{e^x}{1 + e^x}. \tag{2.25}$$

Similar to logistic regression, this function is also commonly used in MLP for binary classification. It is primarily used in the output layer. In hidden layers, it can cause saturation for high and low input values [26], [28].

Hyperbolic Tangent: The hyperbolic tangent function can be expressed in terms of the sigmoid function σ as

$$\tanh(x) = 2\sigma(2x) - 1. \tag{2.26}$$

[2] In some cases, omitting an activation might be used to reduce the number of parameters [26].

The hyperbolic tangent usually performs better than the sigmoid in hidden layers [26], [28]. Unlike the sigmoid, the hyperbolic tangent is similar to a linear function at the point $x = 0$. Compared to the sigmoid, this makes training neural networks easier [26].

Softmax: This function is a generalization of the sigmoid. Compared to the sigmoid, the softmax function represents a probability distribution over m classes. If $\mathbf{x}$ is an m-dimensional vector, the application of the softmax function to $\mathbf{x}$ can be written as

$$\text{softmax}(\mathbf{x})_i = \frac{\exp(x_i)}{\sum_j \exp(x_j)}, \tag{2.27}$$

where i is the i-th element of the m-dimensional output vector.

2.3.2 Loss Functions

The loss function is an important aspect of any neural network. It measures the difference between the model's outputs and the true labels. The goal of the training process is to find the model parameters that minimize the loss function given.

Alternatively, if we imagine an NN as a function defined by its parameters, then the loss function can be viewed as a functional. This functional assigns each function (NN) a penalty for incorrect outputs. The goal of the training process is to find the function (NN) that minimizes this functional (loss function). Using both perspectives, we will describe the most common loss functions.

MSE: Mean Squared Error (MSE) is defined as

$$\text{MSE}(\theta) = \mathbb{E}_{\mathbf{x},\mathbf{y}\sim\hat{p}_{\text{data}}} \|\boldsymbol{y} - f(\boldsymbol{x}, \boldsymbol{\theta})\|^2, \tag{2.28}$$

where $\hat{p}_{\text{data}}$ is the empirical distribution of the training data, $\boldsymbol{\theta}$ are the model parameters. The function $f(\boldsymbol{x}, \boldsymbol{\theta})$ represents model prediction, whereas $\boldsymbol{y}$ desired output.

Considering the loss function as a functional, the optimization problem can be formulated as

$$f^* = \arg\min_f \mathbb{E}_{\mathbf{x},\mathbf{y}\sim p_{\text{data}}} \|\mathbf{y} - f(\mathbf{x})\|^2, \tag{2.29}$$

where p_{data} is the distribution generating the data. The optimal f^* is given as the one that minimizes the average mean squared error. It can be shown that the solution to this problem is [26]

$$f^*(\boldsymbol{x}) = \mathbb{E}_{\mathbf{y}\sim p_{\text{data}}(\mathbf{y}|\boldsymbol{x})}[\mathbf{y}]. \tag{2.30}$$

Hence, the optimal f^* is the average value of y given input $\mathbf{x}$. This result holds as long as the function f^* is within the set over which we optimize (2.29).

MSE is usually not preferred for modeling binary data [26]. The combination of the MSE loss function with a sigmoid output layer could cause training problems. Some output units may become saturated, thus producing very small gradients [26].

BCE: Cross-Entropy (CE) is defined as

$$\mathrm{CE}(\boldsymbol{\theta}) = -\mathbb{E}_{\mathbf{x},\mathbf{y}\sim\hat{p}_{\mathrm{data}}} \log p_{\mathrm{model}}(\mathbf{y} \mid \mathbf{x}), \tag{2.31}$$

where $p_{\mathrm{model}}(\mathbf{y} \mid \mathbf{x})$ is the probability that the model produces $\mathbf{y}$ given the input $\mathbf{x}$. For a model that predicts binary data, we can rewrite this formula as

$$BCE(\boldsymbol{\theta}) = -\frac{1}{N}\sum_{i=1}^{N} \left[y_i \log f(\boldsymbol{x_i};\boldsymbol{\theta}) + (1-y_i)\log(1-f(\boldsymbol{x_i};\boldsymbol{\theta}))\right], \tag{2.32}$$

where N is the sample size.

Cross-entropy is preferred when used together with the sigmoid activation function in the output layer [26]. Since the sigmoid is expressed in terms of exponentials and cross-entropy in terms of logarithms, their combination results in a simple mathematical formula for the derivative, given as $f(\mathbf{x},\theta) - y$. Consequently, the gradient becomes saturated only when the model already knows the correct answer.

2.3.3 Optimizers

Theoretically, the optimization of a convex function converges from any starting point [26]. However, in neural networks, loss functions are usually non-convex, which makes training more difficult. Non-convex functions can have many local minima and saddle points, where the optimization process can get stuck or slow down. In the following section, we will describe the most common algorithm for training neural networks.

SGD: Stochastic Gradient Descent (SGD) is an alternative to the classical gradient descent algorithm. While gradient descent uses the entire training set to compute the gradient, SGD calculates the gradient using only a sample of the data. Because of this, SGD can significantly speed up the training process.

Adam: One problem with using the SGD algorithm is that the learning rate often needs to be set by hand and changed during training. To solve this, optimizers with adaptive learning rate were implemented. Adam (Adaptive Moment Estimation) is one of these optimizers [30]. It uses information from previous gradients to change the learning rate for each parameter. It also uses moving averages to make the updates more stable. Adam is considered as robust method [26].

2.3.4 Regularization

As a regularization is referred any modification of a learning method that aims to reduce test error, even though such modification might possibly increase the training error [26]. There are many regularizations, we will name the one used in this thesis.

Dropout: Dropout is a powerful regularization method for a broad family of models, which can be combined with other regularization techniques [26]. Dropout is based on randomly multiplying some unit outputs by zero, thus temporarily removing them from the network. Each removal is independent of the others and is performed at the beginning of a mini-batch. Dropout is applied only on the input and hidden units. The removal is controlled by the hyperparameter p, which defines the probability of dropping them out. The removal is followed by classical forward propagation, backward propagation, and parameter update. Dropout can be viewed as training multiple NN models that share the same parameters, each inheriting only a subset from the parent network [26]. At test time, dropout is turned off and the full network is used, with activations scaled to reflect the effect of dropout during training. The main advantage of dropout is that it adaptively disrupts the learned information, thus forcing the model to use all the information provided in the input rather than relying on a small subset of extracted features [26], [31].

L_2 **regularization**: Similarly to L_2 normalization in regression models, we can modify the loss function to penalize the norm of NN weights as

$$\tilde{J}(\mathbf{w};\mathbf{X},\mathbf{y}) = J(\mathbf{w};\mathbf{X},\mathbf{y}) + \frac{\alpha}{2}\mathbf{w}^\top\mathbf{w}, \tag{2.33}$$

where J and $\tilde{J}$ are the original and modified loss functions, respectively. The second term represents the L_2 penalty for the weights denoted by $\mathbf{w}$. The regularization is controlled by the hyperparameter α. In L_2 regularization, only weights are penalized, not biases, because they typically require less data to be fitted accurately [26]. L_2 normalization mainly affects parameters that do not significantly contribute to the reduction of the loss function. These parameters decay away, whereas the significant ones are kept relatively intact [26].

Early stopping: Early stopping might be used either alone or in combination with any other regularization [26]. It determines when to terminate the learning process by watching the evolution of the validation loss. During the training process, each time the validation loss improves, the current parameters are stored. If the validation loss does not improve within a pre-specified number of iterations, the learning process terminates. The final model parameters are restored from the most recent checkpoint. The prespecified number of iterations is a hyperparameter, commonly referred to as patience [26].

2.4 SHAP Values

This section aims to introduce Shapley values, respectively, SHAP values. Shapley values originate in game theory [32], but were adapted for machine learning purposes in 2017 [33]. This adaptation is referred to as SHapley Additive exPlanations (SHAP), it aims to retain fundamental properties of Shapley values despite computational challenges.

The motivation behind SHAP values is the following. Having the best predictive model is sometimes not enough. A super-model might outperform any other model in terms of accuracy, nevertheless, in some applications, such model might not be useful if it lacks interpretability. For example, in NN models, we know the input and we know the output of the model, but we do not know why the model came to this conclusion. Additionally, if we want to change the output in a certain way, we do not know how to modify the input features. This is a disadvantage of neural networks compared to simple linear regression. Linear regression models are preferred in many fields due to their interpretability and reliability. They offer straightforward insight into the model structure and, most importantly, they use a solid statistical background to provide p-values and hypothesis testing. On the other hand, they are not capable of capturing such complex patterns as neural networks.

SHAP values bring an alternative to the classical regression explanation. Although they do not offer any p-values or statistical testing, they are founded on solid mathematical axioms [32]–[35]. Moreover, this approach belongs to model-agnostic methods, which means that it can be used for almost any model. For a one-dimensional output, SHAP values do not require any restriction on the model structure. Therefore, they can be computed for machine learning models, including neural networks and logistic regression models.

Let us first introduce the Shapley values with a simple example. Suppose that we have a model that predicts whether a given person is a vizier. The input features are three binary variables: “is male”, “has their own pyramid”, and “has married the king’s daughter”. Imagine that we have an individual who is “male”, “owns a pyramid”, and “has married the king’s daughter”, and the predicted probability that he is a vizier is 0.9. Additionally, the average probability across all individuals is only 0.1. We see that the predicted probability is quite high compared to the average. Therefore, we wonder how each feature contributed to the output. We would like to know whether “male” moved the prediction up by 0.3 above the average, “has their own pyramid” by 0.4, and “has married the king’s daughter” by 0.1. Summing these up, the output is 0.8 above the average prediction. Shapley values offer an answer in a similar way [34], [35].

The aforementioned example served as motivation, but let us be more precise. SHAP values are computed for each observation. More precisely, each observation has as many SHAP values as there are input features. Let f be a model that takes p features as input. Then, the SHAP value for an observation x and the i-th feature is given as:

$$\phi_i(x) = \sum_{S \subseteq \{x_1,\dots,x_p\} \setminus \{x_i\}} \frac{|S|!\,(p-|S|-1)!}{p!}(f(S \cup \{i\}) - f(S)), \tag{2.34}$$

where S is any subset of all input features except the i-th feature, $f(S)$ is the model prediction for feature values x in the set S and marginalized over features not in S. The difference $f(S \cup \{i\}) - f(S)$ tells us how the prediction would change if we add the feature i into the set S. These differences are summed over all possible sets S. The factorials in the equation serve as a normalization that takes into account the cardinality of the set S. Thanks to this normalization, the differences are summed in a fair way.

The Shapley value is a method from coalitional game theory, whence we can view the above equation in a more intuitive way. Imagine that there is a game where one particular observation x tries to gain the highest score $f(x)$. In addition, each feature is a solo player in the game. We can make different coalitions of players (features), represented by the set S. We can play the game with only players from the set S, or we can also join them feature i, obtaining the set of players $S \cup \{i\}$. The difference between these two scores tells us how beneficial the presence of feature i was. We sum these benefits over all coalitions of S to see the average contribution of feature i. In this way, we can interpret the Shapley value ϕ_i for observation x.

With this understanding, we can name the main properties of Shapley values. It has been proven that the only function that fulfills the following properties is uniquely defined [33], [34]. The main properties of Shapley values are:

1. **Dummy:** for a feature i that never contributes to any marginal value $f(x)$, the corresponding Shapley value ϕ_i must be zero.
2. **Substitutability:** if two features i and j contribute equally to all possible subsets, then $\phi_i = \phi_j$.
3. **Additivity:** if the final model consists of several submodels, the final Shapley value ϕ_i is equal to the sum of SHAP values from these submodels.
4. **Efficiency:** adding up Shapley values for all features must yield the difference between the prediction of $f(x)$ and the overall average prediction, mathematically:

$$\sum_{i=1}^{p} \phi_i(x) = f(x) - \mathbb{E}_X f(X), \tag{2.35}$$

   where $\mathbb{E}_X[f(X)]$ is the mean value computed across all observations.

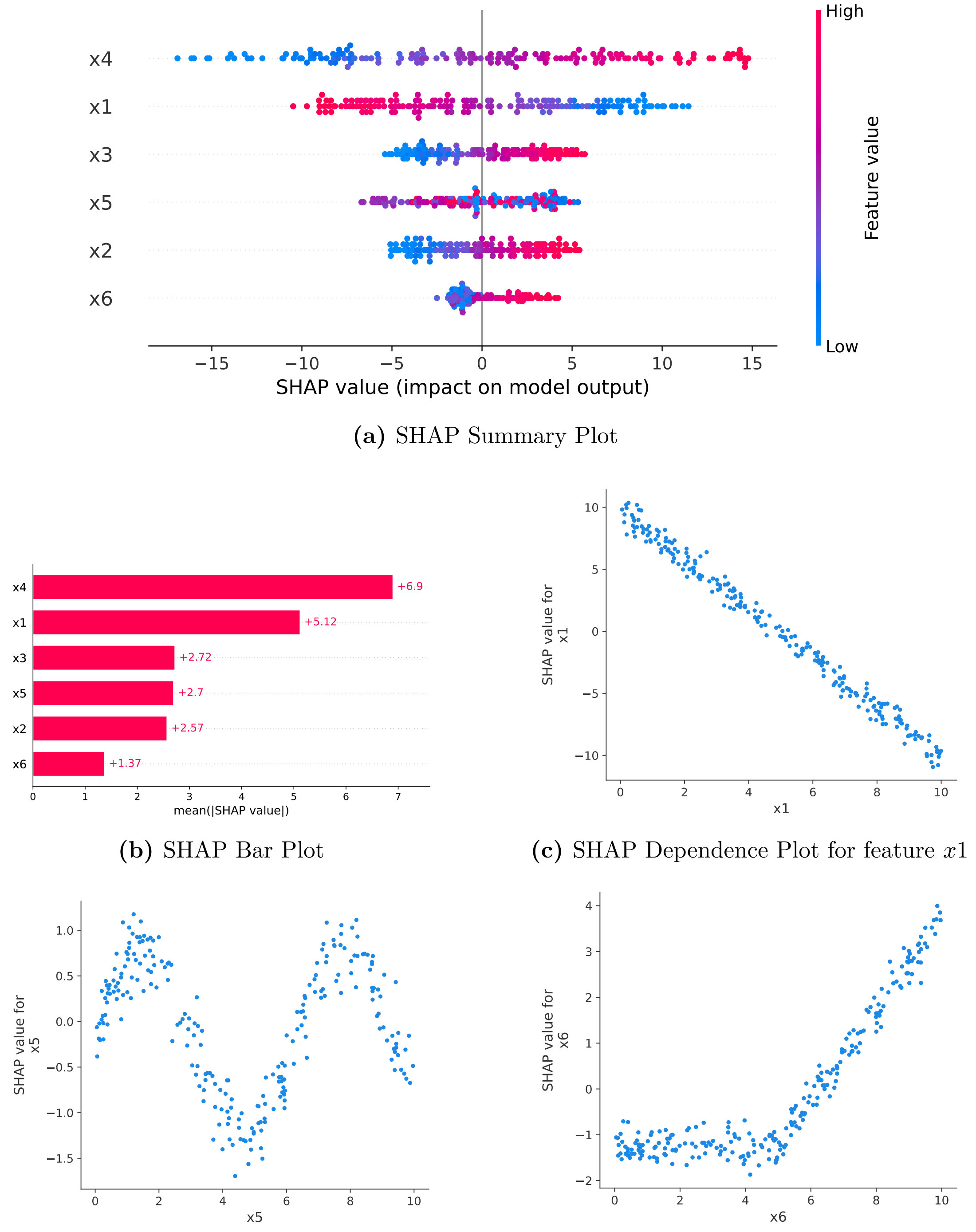
Having mentioned the definition of Shapley values and their properties, in the rest of this section we will show how to inspect these values. Since there are no p-values and no hypothesis testing, the most common way to analyze SHAP values is by plots. Let f be a black-box model (function) that takes six numeric features as input and produces a single number as output. Mathematically, $f : \mathbb{R}^6 \to \mathbb{R}$. Our task is to analyze how each input feature affects the output. Normally, the function f would be a black box for us, but for the purposes of this example, let's say that f is of the form

$$f(x_1, x_2, x_3, x_4, x_5, x_6) = -2x_1 + x_2 + x_3 + 3x_4 + \sin(x_5) + \mathrm{ReLu}(x_6 - 5) + \mathrm{rn}, \tag{2.36}$$

where ReLU is the rectifier function defined in (2.23), and rn is some random noise. We see that x_1 has a negative coefficient and x_2 has the same coefficient as x_3. Features x_5 and x_6 have a non-linear dependency. In the following different plots of SHAP values will be described and demonstrated on the "black-box" function f.

- **SHAP Summary Plot:** Shows dispersion of SHAP value for each feature. Features are shown on the y-axis, SHAP values on the x-axis. This plot can be viewed in 2.6a.

  Inspecting the linear part of the function f, we see that x_4 has the highest dispersion of SHAP values, followed by x_1, x_2, and x_3. The summary plots



(a) SHAP Summary Plot

(b) SHAP Bar Plot

(c) SHAP Dependence Plot for feature $x1$

(d) SHAP Dependence Plot for feature $x5$

(e) SHAP Dependence Plot for feature $x6$

Figure 2.6: Graphs of several types of SHAP value plots. Figure (a) shows a SHAP summary plot, with SHAP values on the x-axis and features on the y-axis. The left legend indicates feature values encoded as colors. Similarly, the plot (b) displays the mean absolute SHAP value for each feature. The last type of plot is the SHAP dependence plot, shown in (c), (d), and (e). These plots depict scatter plots where the x-axis represents feature values, and the y-axis represents SHAP values. Each of the plots (c), (d), and (e) reveals the dependency between different features, namely x_1, x_5, and x_6. The dataset for these plots was randomly generated from a uniform distribution.

show the same pattern for x_2 and x_3 since these two features have the same dependency in f. Furthermore, features x_2, x_3, and x_4 exhibit a positive correlation between SHAP values and feature values. On the other hand, feature x_1 is negatively correlated due to its negative coefficient -2 in f.

The feature x_5 does not have any significant correlation between SHAPs and feature values since its dependency is given as $\sin(x)$ in f. Lastly, for x_6, high feature values seem to push f up, whereas small values appear to have a negligible impact.

- **SHAP Bar Plot:** Show mean of absolute SHAP values. Again, features are shown on the y-axis and ordered according to the mean. In the same way are also ordered features in SHAP summary plot. An example of such a plot is given in 2.6b.

- **SHAP Dependence Plot:** Show a scatter plot of one fixed feature values again its SHAP values. This graph is useful particularly for continuous features as it might reveal dependency between SHAP and feature values. These plots are shown in 2.6c, 2.6d, 2.6e.

  The plot 2.6c shows that there is a negative linear dependency between the feature x_1 and its SHAP values. The plot 2.6d reveals the sinusoidal dependency of x_5 in f. Finally, the plot 2.6e shows a pattern very similar to the ReLU function.

This section introduced SHAP values based on their mathematical definition as well as an example from game theory. It concluded with a demonstration of various SHAP value plots on generated data.

2.5 Methods for Model Evaluation

This section introduces scores for a binary classifier. A common way to analyze a binary classifier is by the receiver operating characteristic (ROC) curve and the calculation of the area under the curve (AUC) [36], [37]. Also, one defines a prediction threshold to get confusion matrix, and rates such recall, specification, and F_1 score [36], [37].

In the next chapter, we will derive a dataset consisting of 3,836 individuals from which only 80 of them have positive target. Furthermore, the test set will include 577 persons with only 12 positive targets. In our study, the target will be vizier, and we will create multiple models.

The ROC curves of any fitted model had the AUC area over 0.97. To analyze confusion matrix, one needs to define a prediction threshold. Turns out that having the threshold the same for each model does not bring many insights into the model itself. Since our test set has only 12 positive targets, the author decided to measure the model quality according to ranked predictions. This approach does not require setting of any threshold. Furthermore, is very intuitive.

All below defined scores use a ranked dataset, constructed as follows. Let us have a dataset where each row represents a unique person. The first column binary encodes if the person is vizier. The second column represents the model prediction whether this person is a vizier. The ranked dataset is obtained by sorting persons according to the prediction. The higher the prediction, the higher the rank. With this ranked dataset we define following scores.

Viziers in overlap: Number of viziers between the non-vizier with the highest rank and the vizier with the lowest rank.

This score is related to *false negatives*. If we would set thresholds for binary classifier as the probability of the non-vizier from the definition, this score gives us *false negatives*.

Non-viziers in overlap: Similar score to the above one, but instead of viziers, we are now counting all non-viziers. The definition is as follows: Number of non-vizier between the non-vizier with the highest rank and the vizier with the lowest rank.

This score is related to *false positives*. If we would set thresholds for binary classifier as the probability of the vizier from the definition, this score gives us *false positives*.

Persons in overlap: Similar score to the above one, but instead of viziers, we are now counting all persons. The definition is as follows: Number of persons between the non-vizier with the highest rank and the vizier with the lowest rank.

Bottom 3rd vizier: The third worst ranked vizier.

Top 3rd non-vizier: The third highest ranked non-vizier.

Custom SHAP plot

SHAP values are commonly visualized by the SHAP bar plot presented in the previous section. For fixed feature, this graph takes all SHAP values and plots the mean of their absolute value. Since we will use this plot only on binary data, we will slightly modify this plot.

As can be seen in Figure 3.2b, all models will be constructed in the way that observation with absence of a given feature (low value) is projected in this plot with zero SHAP value (blue dots), whereas presence of a feature has a positive SHAP (red dots).

Taking the mean of all observation (both blue and red dots) would shift the mean significantly towards zero. Hence, we define our custom SHAP plot as the mean of only these SHAP values that correspond to these observations where the feature was present (red dots). This plot can be seen in Figure 11.

Chapter 3

Model Construction

This chapters describes how the models were constructed. First, we will describe our modeling task. In the following, we will introduce our target feature. In the third section, we will create two datasets based on the Maatbase database. The last section documents how the models were selected. It describes how hyperparameters were adjusted, performance of these models will be summarized in the next chapter.

3.1 Model Definition

The aim of the modeling is as follows. In the Old Kingdom of Egypt there was one high-ranking title, vizier. This title was already introduced in the first chapter, briefly the vizier was the highest official [4], [8]. Since viziers were holding significant power and responsibility, this function can be compared to the current prime minister. The task of our modeling is to create a model that predicts whether a given person holds vizier title, with respect to other titles he held.

The final model is not intended for future prediction. We will study the model from two perspectives. Firstly, we will analyze the impact of each individual input feature on the prediction. This approach will be based on SHAP values. This analysis will point out features that have the highest impact on vizier title.

Secondly, we will analyze the prediction ability of the model. There are known persons who achieved high ranking in the Old Kingdom but were not viziers [1], therefore, we can expect several false positives. The aim of this analysis is to make a list of non-vizier persons with potentially significant influence in the Old Kingdom.

We will create not only one model, but several models. This approach has the following reasons. Firstly, the author does not have a priori knowledge about the significance of the input features. To reduce bias, the author created several models. The second reason is that the aim is to analyze as many input features as possible, but this is impossible due to feature multicollinearity. Therefore, he created three types of model, with the number of input features relatively small, medium, and high. The model with a small number of input features will be relatively simple and hence easy to train. The model with a high number of features will need more regularization techniques to obtain meaningful SHAP values.

We will use two main approaches, logistic regression and multilayer perceptron. The former will be used as a baseline, and the former will try to capture potential nonlinear pattern in the data. The author prefers to use more standard approaches than state-of-the-art models. Standard methods are relatively well documented, which helps ensure that they can be understood by those outside the machine learning field – such as Egyptologists – thereby reducing the risk of misinterpretation.

3.2 Target Feature

All our models will have the same explanatory variable, they will predict whether a given person was a vizier. There are two titles that determine the vizier, they are listed in Table 3.1.

Jones ID	**Title name**	**Translation of the title**
1000–1001/3706	tAyty TAty (n) zAb	vizier
1001/3707	tAyty TAty (n) zAb mAa	true vizier

Table 3.1: Overview of vizier titles in Maatbase database.

The model target was chosen as a binary coded variable, one symbolizing that the given person achieved any of the above-listed vizier titles, otherwise zero.

It is important to mention one challenge of our modeling. There are only 80 viziers in the database, which is about 2.1 % of all persons. The author decided not to use any correction, such as random oversampling or synthetic SMOTE[1], this correction does not necessarily lead to better results [39], [40]. According to paper [39], resampling leads to better model performance only if the classifier is weak. As will be seen, the LR model will be a relatively perform relatively well, so the dataset does not need correction. Furthermore, paper [40] examines imbalance corrections for the LR model, in conclusion authors warn for limitation of its usage in a prediction model.

By observation of titulary of viziers, the author noticed that 5 viziers have only vizier title and not any other. Since our models will make predictions based on persons' titularity, these viziers will be removed from the dataset. There is no information that the model could learn from these persons with one single title. These viziers are listed in Table 3.2.

[1] The Synthetic Minority Over-sampling Technique (SMOTE) is a method used to reduce class imbalance by generating synthetic samples [38].

ID_person	Name
481	Wr-bAw-bA
856	CSm-nfr
2698	M...
3563	KA(.i)-mn(.i)
3600	2nw

Table 3.2: Overview of viziers removed from the dataset. These viziers had only one title, the vizier.

3.3 Feature Selection

Maatbase serves as the only data source for all introduced models. The database contains tables *df_person* and *df_titles*, carrying all recorded individuals and titles. Furthermore, table *df_person_title* holds all relations between persons and their achieved titles. For easier orientation in the database tables and their primary keys, the author created a database schema. This schema can be found in the Attachment, Figure 3.

At the beginning of the study, there were 4,962 persons, 2,219 titles and 13,695 person-title relations in the database. Persons will play the role of observations in our models, whereas titles will serve as features. Since the number of titles is high compared to the number of persons, the main task of this section is to choose an appropriate small subset of titles.

Anyway, we will first restrict the number of persons. It is well known that a vizier was always a male individual. Therefore, we restrict persons to men only. By this step, the future models will not be forced to learn this known pattern, instead, we let them focus on patterns behind titles. After restriction to males only, there are 3,930 records.

Since the feature dynasty will later be added to the models, individuals with an unknown dynasty are excluded. By omitting these persons, we arrive at the final 3,836 records. Besides unknown dynasty, these males also had low number of titles, hence they were not adepts for viziers.

The distribution of titles among persons showed many patterns. Some titles were very rare, occurring only in the titulary of a few persons. Some of them seem to be tightly connected with viziers. To investigate these patterns, the following statistics were computed for each title:

- **count**: number of persons holding the given title,
- $P(\text{vizier}|\text{title A})$: estimated conditional probability that a person is a vizier, given that they achieved the particular title

$$P(\text{vizier}|\text{title A}) = \frac{\#\{\text{viziers having title A}\}}{\#\{\text{persons having title A}\}}, \tag{3.1}$$

- $P(\text{title A}|\text{vizier})$: estimated conditional probability that a person held the particular title, given that the person was a vizier

$$P(\text{title A}|\text{vizier}) = \frac{\#\{\text{viziers having title A}\}}{\#\{\text{viziers}\}}. \tag{3.2}$$

Conditional probabilities were estimated for all males. These statistics pointed out three titles that were strongly connected with viziers. These titles are shown in Table 3.3. The first two were achieved only by viziers, even though these titles have different translations. The third was achieved by viziers in 17 cases out of 18 persons who ever held this title. Since the occurrence of any of these three titles strongly suggests that the person is a vizier, their presence in our models could obscure more hidden patterns. Therefore, all three titles were removed. Removing these dominant players might help other features show their impact.

Jones ID	**Title name**	**V**	**NV**	**Translation of the title**
935/3448	cHD Hm(w)-nTr Mn-nfr-Ppy	7	0	inspector of hem-netjer-priests of Menneferpepy, the pyramid of Pepy I
148–149/577	imy-rA niwt (mr)	6	0	overseer of the pyramid town
165/630	imy-rA Hwt-wrt 6	17	1	overseer of the six great law-court

Table 3.3: Overview of removed titles. Columns *V* and *NV* show the number of viziers and non-viziers holding the given title, respectively.

On the other hand, the main purpose of the models is not to create the best predictive model. These models are not intended to be used for the future prediction of new persons. We prefer the models to discover unknown patterns in the data, and the above-mentioned pattern is too obvious.

Based on the aforementioned statistics, two feature datasets were created. The first focuses only on the probability $P(\text{vizier}|\text{title})$, whereas the second focuses on all three statistics and included some additional features. These datasets will be described separately.

Feature Version 1

This feature version was made with the following motivation: choose only these titles that have enough records among viziers. These features were chosen with the restriction

$$P(\text{title}|\text{vizier}) > 0.125, \tag{3.3}$$

which is equivalent to titles that have appeared more than ten times in the titulary of viziers. This dataset represents titles with relatively high number of occurrences and hence reduces the risk of overfitting. The dataset is summarized in Table 3.6.

Feature Version 2

The second feature version was created with the following motivation: to the previous version add other titles, which are strongly connected to viziers. Furthermore, this version also includes other features that are supposed to have a strong impact on the output.

Titles selected into this version had to fulfill

$$\text{count}(\text{title}) > 10 \land (P(\text{title} \mid \text{vizier}) > 0.125 \lor P(\text{vizier} \mid \text{title}) > 0.5) \tag{3.4}$$

Again, condition $P(\text{title}|\text{vizier} > 0.125$ selects only titles that have occurred more than ten times among viziers. Condition $P(\text{vizier}|\text{title}) > 0.5$ represents titles that occur more often among viziers than non-viziers. Restriction $\text{count}(\text{title}) > 10$ is required to avoid the risk of overfitting.

Moreover, following additional features were added to this version:

Father was vizier: a binary variable indicating whether the father of a given person was already vizier. This feature is intended to model nepotistic behavior in the Old Kingdom [1], [6], [7]. For example, there is a known family in which the title of vizier was held for three generations [7]. This feature was extracted from table *df_family_relations*.

Dynasty: feature indicating the dynasty in which a given person lived. The evolution of the Old Kingdom society was non-linear [1]. It underwent many changes, which also projected into evolution of titulary. There are known periods in which some titles started to appear and periods in which other titles disappeared [2], [10]. Therefore, the presence of dynasty in the models might be important.

The dynasty was encoded as a categorical variable with three levels, with thresholds set at the mid-age of the Fifth and Sixth dynasties. These thresholds are consistent with the findings in [10]. Its summary is in Table 3.4. This feature was extracted from table *df_general*.

Feature name	**Period**
dynasty0	≤ mid-Fifth dynasty
dynasty1	between the mid-Fifth and mid-Sixth dynasties
dynasty2	≥ mid-Sixth dynasty

Table 3.4: Overview of the encoded dynasty feature.

Principal components: first two components of PCA from the example given in Subsection 2.1. This PCA was performed on normalized data from Table 2.1, the PCA is shown in Figure 2.3b.

In summary, Egyptologists classified titles into seven types. Based on this classification, each title can be clustered into one of the following categories: administration, epithet, family, priest, privy, rank, worker, or unclassified.

By applying PCA to this category space, features *PC1* and *PC2* representing the first two principal components were constructed. They might indicate the sphere in which a person had the greatest power.

Splitting Dataset and Multicollinearity Handling

The selection of feature versions 1 and 2 was followed by splitting the dataset into train, validation, and test sets. The split was done in a way that both feature versions share the same persons in each set, hence the model evaluation will be made on the same persons regardless of the feature version. The train, validation, and test sets were created with ratios 70%, 15 %, and 15 %, respectively. The size of these sets was 2684, 575, and 577, respectively.

While splitting the dataset, the author used a stratified split, which ensured that the target feature had the same class proportion in each set. In addition, the author monitored the distribution of each feature in the sets. This monitoring was done by observing the ratio between the mean of the features in a given set and the overall mean. This helped avoid a degenerate case where some feature would be completely omitted in one of the sets, potentially biasing the post-model analyses.

Originally, both feature versions were used directly in modeling, but the fitted models showed issues indicating multicollinearity. Several coefficients of logistic regression were intuitively extremely negative. Also, the training of NN was unstable. Therefore, the author decided to further restrict the feature sets. By calculating the Variance Inflation Factor (VIF) (2.22), features with a VIF value above 10 were removed. This step significantly improved the stability of NNs. Later, the author removed all features with VIF > 5, which resulted in even better stability. The removed features are listed in Table 3.5.

The VIF restriction removed among other also features *dynasty2* and *PC1*, both having very high VIF value 52.0, and 18.4, respectively. Removing *dynasty2* carry a issue of changing the reference level from *dynasty0* to *dynasty0+dynasty2*.

Conclusion of Feature Selection

In this section we introduce two feature versions, they are summarized in Table 3.6. Since some features had a high VIF value, they were removed from the selection, these are listed in Table 3.5.

The final first feature version includes 36 titles. The second version contains 37 titles and 3 additional features (one feature encoding if father was vizier, one dynasty, and one principal component). For a complete list of the features included in each version see Table 2 in the Attachment.

The selected features were spitted into training, validation and test set with ratios 70%, 15 %, and 15 %, respectively. During this split, the author carefully monitored the distribution of features across the sets.

It is important to mention that all features except *PC1* are binary, where the value one means the presence of the feature. Furthermore, it is important to note that both feature versions were made in a way that the presence of any binary feature (except *dynasty1*) should have a positive impact on the model output. Hence, during

Jones ID	Title name	VIF 1	VIF 2	Translation of the title
433/1597; 801/2929	mniw Nxn / zAw Nxn	20.68	20.8	protector/guardian of Hierakonpolis
885 /3241	cm / ctm	9.7	7.4	se(te)m-priest, chief celebrant in the rite of...
904–905 /3318	cmcw cnwt	8.9	7.9	elder of the cnwt-shrine/house
6 /22	iwn knmwt	8.8		support of kenemut; pillar of the knmt-folk
453–454 /1698	mdw rxyt	8.6	11.3	staff of the Rekhyt-people /commoners/ herdsman...
87–88 /374	imy-rA wabty	6.6	9.2	overseerof two workshops
648 /2374	Hry-tp Nxb	6.3	5.8	Chief of Nekhbite
696–697 /2545	xrp (i)m(yw) nTrw		26.8	director of those who are among the gods, dire...
684–685 /2501	Htc(?) Inpw		10.2	... of Anubis(?), (precise reading unknown, me...
806 /2947	aD-mr (n) zAb		7.2	administrator/boundary official of the king ju...
209–210 /780	imy-rA zS(w) a(w) (nw) ncwt		5.2	overseer of the royal document scribes
695 /2541	xrp iAwt nbwt nTrwt		5.2	director of every divine office According to S...
	dynasty2		52.0	artificial feature, coding dynasty ≥ 6.5
	PC1		18.4	articicial feature, coding the first principal component

Table 3.5: Overview of features that were removed from the dataset due to high VIF values. Columns *VIF 1* and *VIF 2* show VIF values in the feature versions 1 and 2, respectively. If the VIF value is not present, the feature was not removed.

Feature version 1

Description	most common titles among viziers
Restriction	$P(\text{title}\vert\text{vizier}) > 0.125$
Number of features	36 titles

Feature version 2

Description	most common titles among viziers plus titles more common among viziers than non-vizier plus nepotism, dynasty, PCA features
Restriction	$\text{count}(\text{title}) > 10 \wedge (P(\text{title} \mid \text{vizier}) > 0.125 \vee P(\text{vizier} \mid \text{title}) > 0.5)$
Number of features	37 titles + 3 additional features

Table 3.6: Overview of feature versions 1 and 2.

modeling, we will monitor SHAP values already in the validation set, to see if this assumption holds.

3.4 Model Selection

This section describes the process of creating models. Namely, it describes used methods and adjusting hyperparameters. The model performance will be commented on in the next chapter.

First, we will describe the logistic regression model (LR). In total, we will train six different LR models. Three for feature version 1 and three for version 2. These types will differ in input size. The first type will always have up to ten input features, the second type between 10 and 20, and the third will use the entire feature version.

Later, we will create six Multilayer perceptron models (MLP) as an alternative to these six LRs models. LR and its corresponding MLP alternative will share the input features.

As discussed in the previous section, both feature versions were created in a way that each feature should be positively correlated with the output. Hence, while training a model, we will monitor its coefficients, respectively, SHAP values to see the sing of the feature impact. If any SHAP value turns out to be negative, this model is most likely overfitting.

A very common feature that signalized overfitting was title *imy iz*. The statistics for this title are summarized in Table 3.7. This title has $P(\text{vizier}|\text{title}) = 0.6522$, so more than half of the holders of this title were viziers. There is no evidence why this title should decrease the probability of becoming a vizier. Hence, if the fitted coefficient or SHAP value is negative, the model is wrong. This example is the reason why we will monitor SHAP values already during the training process.

Jones ID	**Title name**	**P(vizier|title)**	**P(title|vizier)**	**count**	**Translation of the title**
49/247	imy iz	0.6522	0.1875	23	he who is in the iz-bureau, councillor

Table 3.7: Overview of title *imy iz*. This title was very often signalizing overfitting.

Similarly, we checked whether the absence of a title corresponds to a zero (or very low) SHAP value. However, no model had such an issue.

3.4.1 Logistic Regression

We will create three types of LR model, each differ in the input features. The first model will choose input features based on a stepwise selection, the second based on LASSO regularization. The third model will use Ridge regularization, hence using the entire feature version.

Step: The input features of these models were chosen based on a stepwise selection. At first, a stepwise selection was performed with both the AIC and the BIC criterion. Since the BIC model was a reduced version of the AIC model, we could use the deviation statistic to compare these two models.

For feature version 1, the test point out that the AIC model is better, but after analysis of its coefficients, the AIC model was rejected. The AIC model fitted the negative coefficient for the aforementioned title *imy iz*, and therefore was not reliable. Table 3.8 shows the fitted coefficients for both BIC and AIC models. The BIC model has all coefficients positive, while the AIC does not.

For feature version 2, both AIC and BIC models fitted negative coefficient for the problematic title *imy iz*. The final model was chosen using BIC with omitted *imy iz* title.

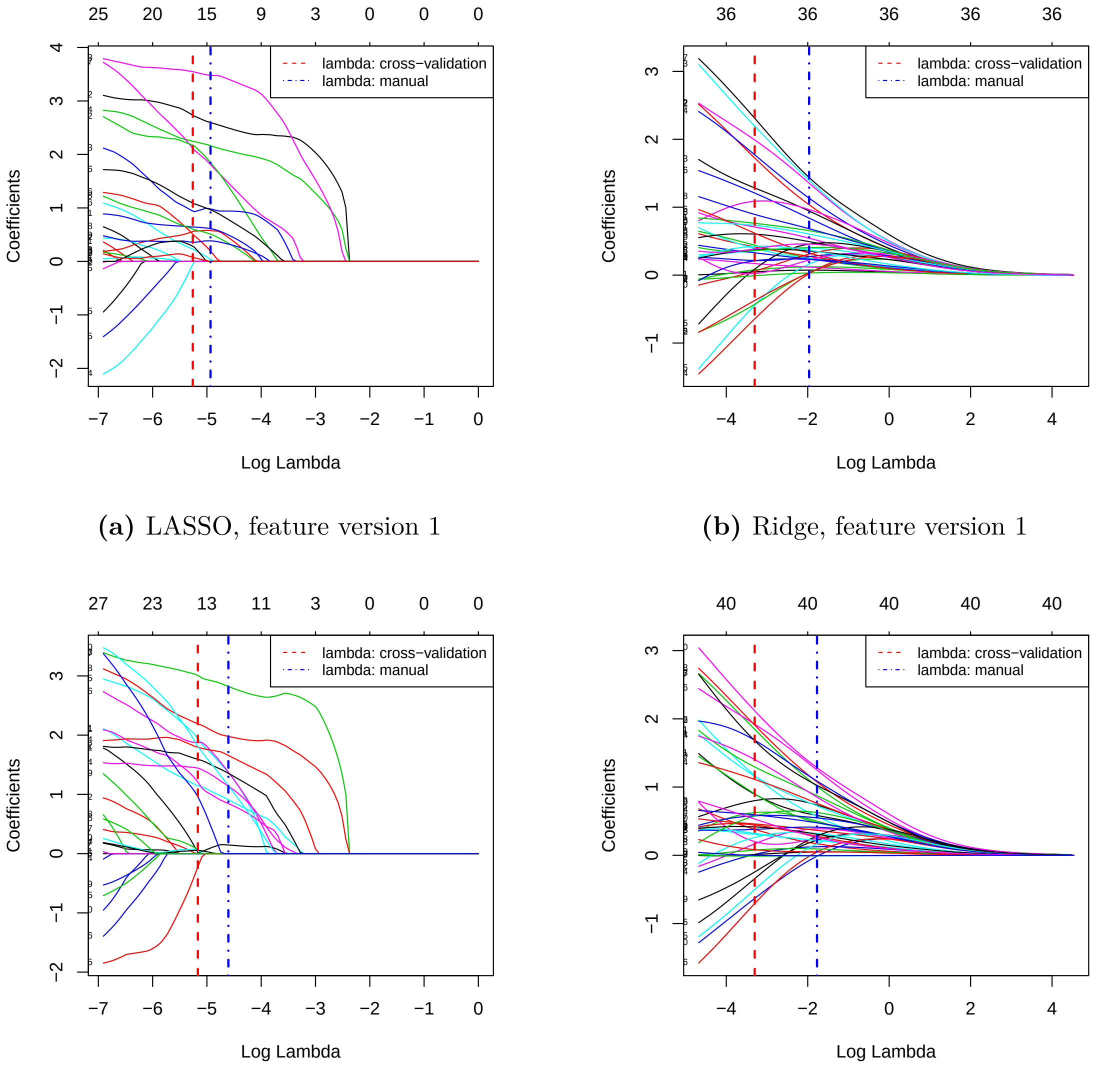
LASSO and Ridge: The input features for these models were chosen based on LASSO, and Ridge regularization, respectively. The hyperparameter λ, defining the strength of the regularization, was approximated by cross-validation. This estimation was used only as rough approximation[2], the final λ was chosen manually.

Figure 3.1 shows the regularization path together with the chosen λ. In all cases, the hyperparameter λ was adjusted in such a way that all coefficients are non-negative (or almost non-negative).

	AIC		**BIC**	
Feature	**Coefficient**	**p-value**	**Coefficient**	**p-value**
Intercept	-7.4776	<2e-16	-6.7074	2E-16
HAty-a	3.349	2.19E-05	3.6504	1.57E-08
iry-pat	3.1129	1.45E-05	2.8294	1.47E-05
wr 5 (m) pr 9Hwty	3.6139	0.00504	3.6768	0.00694
zA ncwt n Xt.f cmcw	3.6724	0.11584	5.9915	2E-16
zA ncwt n Xt.f	2.0276	0.07883	3.5772	2.25E-05
imy iz	-2.7628	0.00516		
imy-rA kAt nbt (nt) ncwt	2.1876	0.00374		
cmr waty	1.1048	0.09911		
imy-rA zS(w) a(w) (nw) ncwt	3.8476	1.83E-06		
aD-mr (n) zAb	1.9902	0.00888		
imy-rA Snwty	2.0854	0.03526		

Table 3.8: Table showing the fitted coefficients and p-values of the logistic regression model. The model corresponds to stepwise selection applied in feature version 1. Whereas the model constructed by BIC has all coefficients positive, the AIC has a negative slope for *imy iz* .

[2] Based on experience with the given dataset, the author observed that random splits often result in degenerate subsets in which certain features are completely missing. Therefore, cross-validation is used only as a rough estimate, and the hyperparameter λ is adjusted manually.



(a) LASSO, feature version 1

(b) Ridge, feature version 1

(c) LASSO, feature version 2

(d) Ridge, feature version 2

Figure 3.1: Regularization path for LASSO (left) and Ridge (right) in LR models. The upper plots correspond to feature version 1, the bottom to version 2. Blue (dot-dashed), and red (dashed) vertical lines corresponds to lambda chosen manually, and by cross-validation, respectively. The horizontal axis represents the hyperparameter λ that controls the strength of regularization. The vertical axis represents the sizes of the fitted coefficients. The curves represent the evolution of the coefficients in different settings of λ.

3.4.2 Multilayer Perceptron

This subsection describes how MLP models were created. First, we explain the importance of monitoring SHAP values. Later we will comment on how the final model architecture was derived.

SHAP Values Monitoring

During the fitting of the models, the biggest challenge was to find a model that had only positive SHAP values. Positive SHAP value indicates that the presence of a given feature increases the average prediction, while negative SHAP values decrease the average prediction [34]. The evaluation of SHAP values was done on the validation set immediately after the model was trained.

While in LR the negative coefficients were suppressed by LASSO and Ridge regularization, in the MLP models the main tool for this regularization was dropout. The effect of dropout is demonstrated in Figure 3.2. This figure shows two NN models, their architecture differs only in usage of dropout. The first model shown in Figure 3.2a has zero dropout, while the second model given in Figure 3.2b has dropout with probability 0.5 in each hidden layer. The model without dropout has two features with negative SHAP values, whereas the model with the dropout none.

Additionally, the second model given in Figure 3.2b shows that the non-presence of any feature has zero SHAP value (blue dots), only the presence of a feature increase prediction (red dots). Therefore, this model behaves as expected.

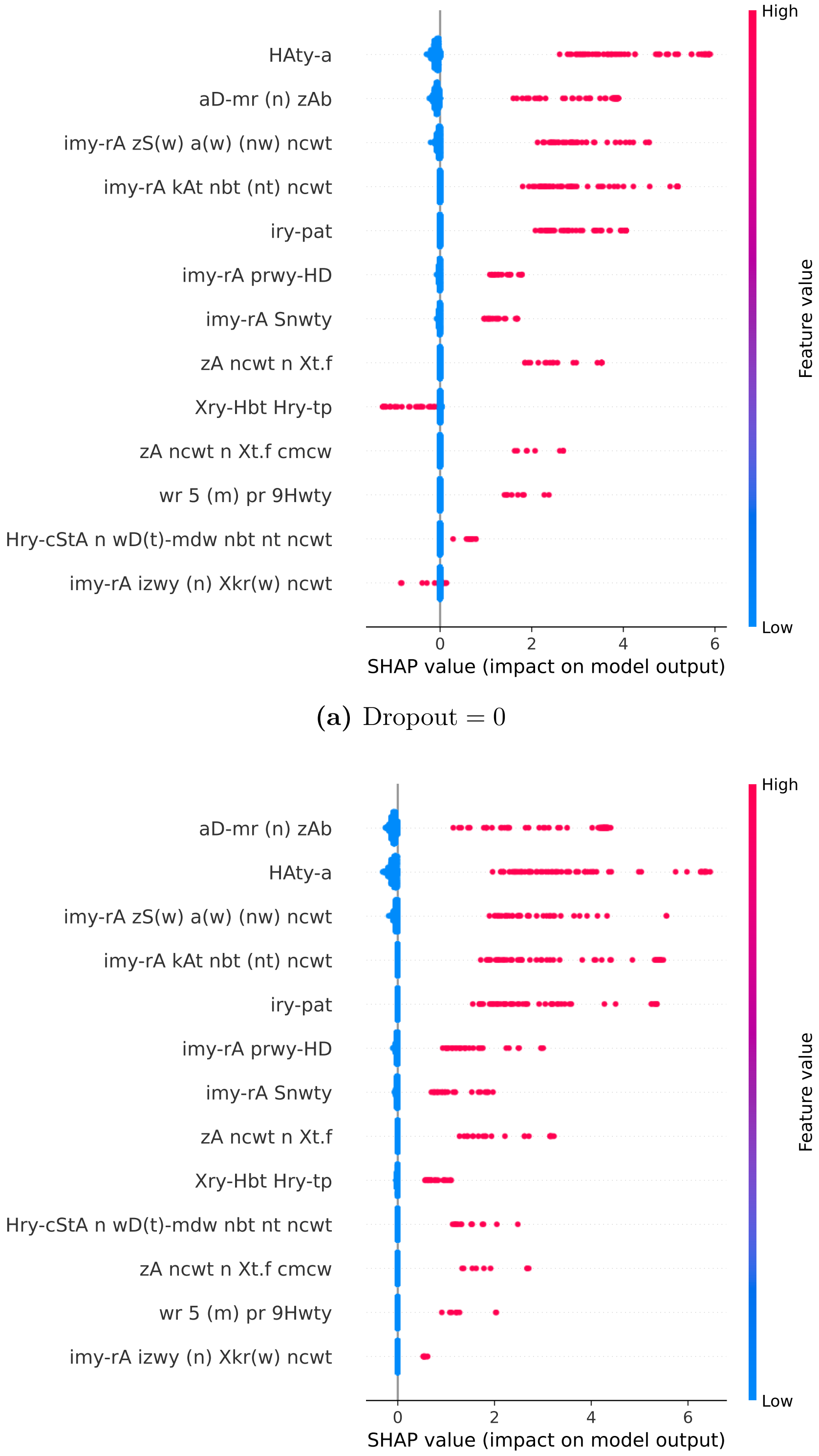
However, dropout was not always sufficient to have strictly positive SHAP values. Hence, in some models we combined dropout with L_2 regularization.

Models Architecture

Each model was trained with batch size 512. Since only 2.1 % of the observations are viziers, this batch size quarantines that there will be on average more than 10 vizier within each batch. In addition, to suppress the class imbalance, the author decided to put extra loss weight on the viziers. This step led to less negative SHAP values.

Every model was trained with Adam optimizer, which performed better than SGD. In some cases, LeakyRelu outperformed ReLu, but only in models with high dropout. The termination of the training process was controlled by early stopping, see Subsection 2.3.4.

The number of hidden layers and their sizes were chosen by starting with zero hidden units and sequentially adding more neurons. For each such setup, the author tried different levels of dropout. The final architecture was selected as the first that produced meaningful SHAP values together with reasonably low validation loss. The final architectures are given in Table 3.9.



(a) Dropout = 0

(b) Dropout = 0.5

Figure 3.2: SHAP summary plots demonstrating the effect of dropout. The upper plot (a) shows the NN model without dropout. The bottom plot (b) shows the NN model with the same architecture but with dropout = 0.5. The blue dots represent observations where the given feature equals 0, the red dots represent observations where it equals 1. The model architecture corresponds to LASSO with feature version 1, as shown in Table 3.9. These SHAP values were computed on the validation set immediately after the training process. The validation loss was 0.0541 for the model with dropout and 0.0715 for the model without dropout.

Model type	**step**		**LASSO**		**Ridge**	
Feature version	1	2	1	2	1	2
Input size	5	10	13	14	36	40
N. of hidden layers	1	1	2	2	2	2
Size of hidden layers	4	4	12, 8	12, 8	12, 8	12, 8
N. of parameters	29	49	281	293	557	605
Learning rate	0.005	0.005	0.002	0.002	0.002	0.002
Dropout	0.1	0.1	0.5	0.5	0.3	0.3
L2 lambda	0	0	0	0.02	0.043	0.043
Activation	ReLu					
Output activation	Sigmoid					
Criterion	BCE					
Positive weight	50.6					
Optimizer	Adam					
Batch size	512					
Patience	40					

Table 3.9: Summary of hyperparameter settings for the six implemented NN models. The *positive weight* represents the modified loss weight for positive labels. The *patience* is the hyperparameter for early stopping.

In feature version 2 we introduced additional variables *father_was_vizier*, *dynasty1*, and *PC1*. Although these variables were not selected in step and the LASSO LR models, we will require the associated MLP models to include them. These additional features were added to the dataset for the purpose, and we would like to inspect their impact in a non-linear model.

Conclusion of Model Selection

In this subsection we introduced six LR models, each model has also associated NN model. Each model might learn different patterns, we will summarize them in the next chapter.

The LR models were selected based on stepwise algorithm, LASSO and Ridge regularization. The estimated coefficients can be found in the attachment, Table 3.

The NN models were first regularized by dropout, if this regularization was not sufficient, the L_2 norm was added. The architecture of the NN models is summarized in Table 3.9. In feature version 2, we required NN models to include features *father_was_vizier*, *dynasty1*, and *PC1* regardless whether they were included in associated LR model.

While finding the appropriate architecture for the NN models, we were monitoring the SHAP values immediately after the training process. This monitoring was crucial while selecting the final model architecture. It enables us to recognize models that learned intuitive patterns.

Chapter 4

Model Evaluation and Results

This chapter comments on results from twelve models derived in the previous chapter. First, we will validate the models performance on the train set. In the second section, we will analyze non-viziers that although were not viziers achieved high ranking in our models. In the section dedicated to titles, we will analyze SHAP values from the models and present these features that have the highest impact on the model prediction.

4.1 Model Summary

This section compares the models described in the previous chapter. In total, we prepared six LR models and six MLP models. The list of all models, together with several scores, is included in Table 4.1. All numbers given in this table were calculated on the test set. This set was the same for all models, it included 12 viziers and 565 non-viziers.

Since the dataset is relatively small and all ROC curves had AUC over 0.97, the author defined its own scores how to evaluate models. All these scores are described in Section 2.5. These scores are also intuitive for people outside the machine learning field. They are based on a ranked dataset and therefore do not require any thresholds such as the confusion matrix, Section 2.5.

This table shows mean overall prediction (column *mean*), mean prediction among viziers (column *mean viziers*) and mean prediction among non-viziers (column *mean non-viziers*). Since MLP models were trained with the extra penalty for positive labels, all MLP models have significantly higher means compared to LR models. Nevertheless, other columns in the table are based on rank, therefore, other columns are not affected by this shift.

In addition, Table 4.1 presents also columns *bottom 3rd vizier*, *top 3rd non-vizier*, *viziers in overlap*, *non-viziers in overlap* and *persons in overlap*. Since the test set included only 12 viziers, the score *viziers in overlap* deviates between 3 and 5 persons for all models. According to this score, no model is significantly better or worse. Also,

Feature version	Model type	Model	Mean	Mean vizier	Mean non-vizier	Bottom 3rd vizier	Top 3rd non-vizier	Viziers in overlap	Non-viziers in overlap	Persons in overlap
1	step	glm	0.0170	0.6236	0.0042	0.0449	0.3283	4	10	14
		nn	0.0656	0.8549	0.0489	0.8673	0.9337	4	10	14
	lasso	glm	0.0176	0.5917	0.0054	0.0408	0.1820	4	26	30
		nn	0.0543	0.8488	0.0375	0.8075	0.9768	5	26	31
	ridge	glm	0.0192	0.5875	0.0071	0.0676	0.0771	3	80	83
		nn	0.0710	0.8775	0.0538	0.8783	0.7262	4	64	68
2	step	glm	0.0179	0.7012	0.0034	0.0417	0.0422	4	17	21
		nn	0.0522	0.8900	0.0344	0.8427	0.8618	3	257	260
	lasso	glm	0.0187	0.6108	0.0061	0.0671	0.0671	4	20	24
		nn	0.0848	0.8674	0.0682	0.7041	0.7774	4	32	36
	ridge	glm	0.0199	0.5850	0.0079	0.1040	0.0926	3	67	70
		nn	0.0859	0.8846	0.0690	0.9182	0.7395	4	62	66

Table 4.1: Overview of the model performance for the twelve implemented models. All numbers were computed on the test set. The column *mean* represents mean over all predictions. Similarly, columns *mean vizier* and *mean non-vizier* represents mean prediction among viziers and non-viziers, respectively. For the explanation of other columns, see Section 2.5. All numbers were calculated on the test set. The architecture for MLP models is described in Table 3.9.

no model managed separate complete the data.

On the contrary, there was a contrast in the score *non-viziers in overlap*. Generally, the smaller input size, the smaller *non-viziers in overlap*. The best *non-viziers in overlap* score was achieved by step MLP model in feature version 1. In this model, only 14 non-vizier out of 565 were located in overlap. With increasing input size, this score was also increasing up to 83 for the Ridge model in feature version 1. The only exception is the step MLP model for feature version 2, but we will comment on this case later.

Feature version 1

Let us first comment feature version 1. In models with smaller input sizes, the MLP models did not outperform LRs, which was expected by the nature of how small sets were created – these input features were obtained by a stepwise algorithm and LASSO regularization in LR models. Mainly, the LR and MLP step model had the same scores in both *viziers in overlap* and *non-viziers in overlap*, 4 viziers and 10 non-viziers, respectively. Since the MLP model was in this case very simple, this result shows consistency between LR and MLP outcomes. Also, the scores for LASSO models were the same, differing only in one vizier in overlap. In the Ridge model, the input features represented the entire feature version 1. In this case MLP outperformed LR model with 68, respectively, 83 people in overlap.

Feature version 2

Feature version 2 included three additional features. Namely, *father_was_vizier*, *dynasty1*, and *PC1*. In the MLP models they presence was required regardless of whether they were chosen within the associated LR model. This is the reason why the step MLP model has 260 *persons in overlap*.

To prove that the root cause for such a high number are the required features, the author also fitted the step LR model with these required features, regardless of whether they were originally chosen by the stepwise algorithm. In this extra model, *person in overlap* exceeded the number 500. Hence, the presence of the additional features seems to harm the model.

Generally, feature version 2 exhibits the same patterns as feature version 1. The smaller the input size, the smaller the score *person in overlap*. Similarly, the Ridge MLP model outperformed its MLP alternative.

Conclusion of model summary

The presence of additional features did not improve the models results, namely the score *non-viziers in overlap*. The models exhibit the general pattern, the smaller input feature size, the better score *persons in overlap*.

Except the step MLP model for feature version 2, all models performed well on the test set. Hence, while analyzing persons and title in the following sections, we will use all models except the step one.

The performance of the models is summarized in Table 4.1. The estimated coefficients of all fitted LR models are shown in the Attachment, Table 3. Calculated SHAP values for feature versions 1 and 2 are presented in Attachment, Table 4 and Table 5, respectively. We will comment on these SHAP values later in the title summary.

4.2 Person Summary

In this section, we will highlight non-viziers that achieved highest scores in fitted models. For this analysis, we will use model prediction from all models except the step MLP model with feature 2. This model, as discussed in the previous section, had too high *non-viziers in overlap*.

Since MLP models were trained with the extra penalty for positive target, predictions of these models are shifted compared to LR models, see Table 4.1. Therefore, while analyzing persons, we will use ranked predictions.

Let us have predictions for all non-viziers from one model, the rank of a person is computed by sorting the prediction in descending order. The higher the prediction, the lower the rank. In this way, we computed the rank for each person in each model. Furthermore, we computed the mean and standard deviation of each person rank.

We will inspect the rank in each data split separately. For train set we will find top 20 non-viziers with highest average rank, for validation and test set we will find top 5 non-viziers. These top non-viziers from train, validation and test set are shown in Figure 4.1a, 4.1b and 4.1c, respectively. These figures show non-viziers with highest mean rank together with rank standard deviation. The complete information on rank can be found in the Attachment, Tables 6, 7, and 8.

Furthermore, Figures 4, 5, 6, 7, 8 from Attachement show the ranking of both non-viziers and viziers. For the train, validation and test set top 120, 40, and 40 persons are listed, respectively. The vizeirs are presented with cross marker, whereas non-viziers with dot marker. These plots reveal several non-viziers that obtained higher prediction than some viziers.

4.3 Title Summary

In this last section, we will inspect the impact of each title in the fitted models. This analysis will be based on SHAP values, which were computed on the test set. However, before the analysis itself, we will recap some findings from the previous chapter, where we created our data set.

Findings from Feature Selection

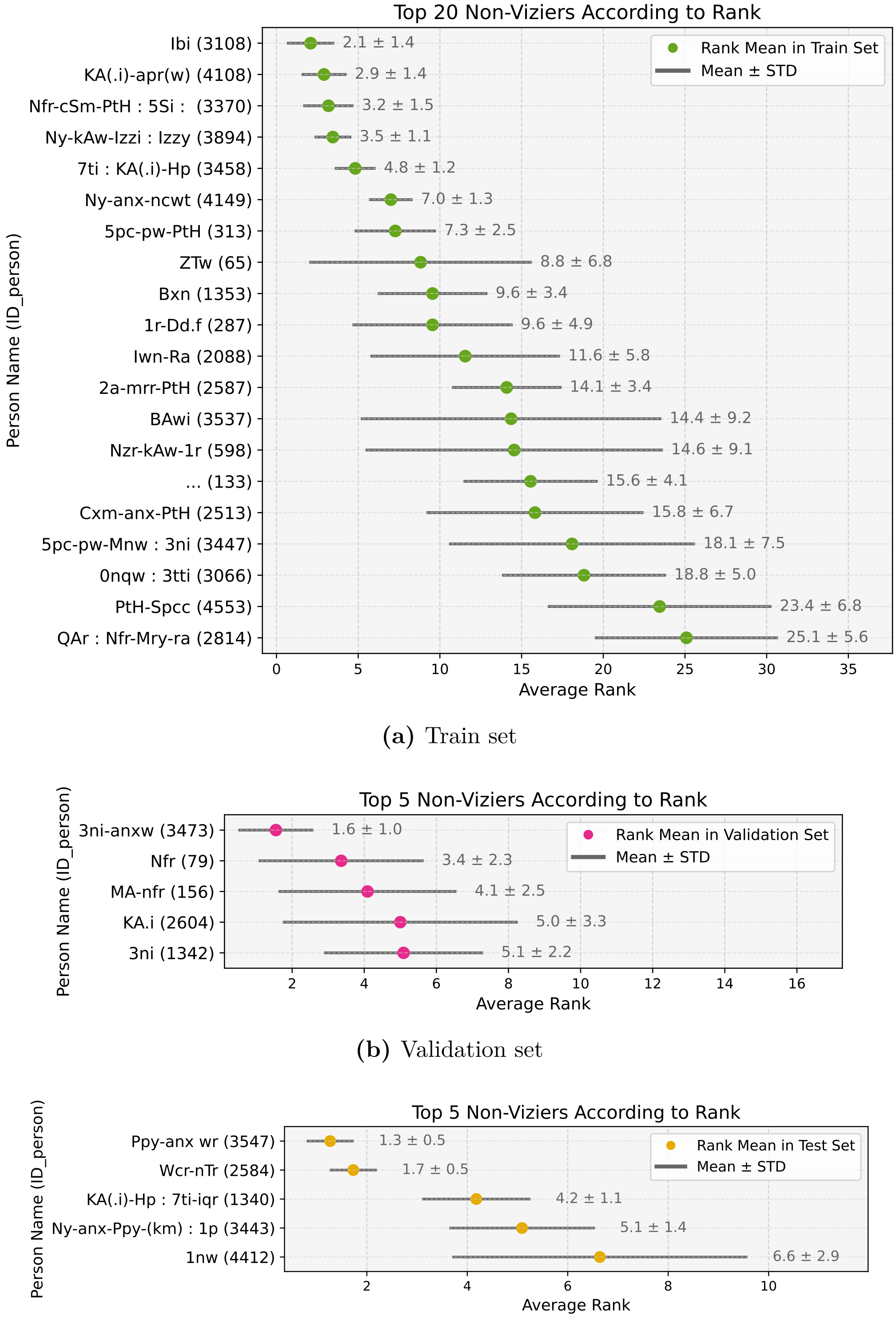
In feature selection, we recognized three titles that were strongly connected with our target title vizier. We omitted these titles because the pattern behind these three was obvious: presence of any such title was directly pointing out that the person is a vizier. These titles are listed in Table 3.3.

Furthermore, in the last step of feature selection, we used VIF to suppress multicollinearity in the data. Several titles were removed. As a result, the LR models lost signs of multicollinearity, such unreasonably negative coefficients. In addition, the training process of MLP was more stable. These removed titles are listed in Table 3.3.

Hence, the following summary did not take any of the aforementioned titles into account. The titles that were part of our modeling are described in Attachment, Table 2. All these titles served as the input features for our modeling, in the following we will summarize their impact.

SHAP value analysis

The analysis was performed using SHAP values. As described in Section 2.5, since our data are binary, we slightly modified the standard SHAP plots. As was shown



(a) Train set

(b) Validation set

(c) Test set

Figure 4.1: Overview of the top-ranked non-viziers in the train set (a), validation set (b), and test set (c). The rank was computed based on predictions among all non-viziers in the corresponding set. The lower the rank, the higher the prediction of the model. The dots represent the mean rank. The vertical lines represent one standard deviation interval. This interval is given as mean minus, respectively plus standard deviation of the rank. The mean and standard deviation are also shown on the right of the deviation interval. These figures are visualizations of Tables 6, 7, and 8 from Attachment. Due to lack of space, the names on horizontal axis are shown only up to 20th character.

in Figure, for observations where the given feature is absent, the SHAP values are almost zero. Hence, in this chapter we will present only means of SHAP values, which were calculated on observations where the given feature was present.

The calculated means of SHAP values for feature versions 1 and 2 are presented in Attachment, Table 4 and 5, respectively. The SHAP mean values from these tables are visualized in Figures 9a–16. Each figure shows SHAP mean values for different models. The higher the SHAP mean value, the higher the impact on model prediction.

These figures show that the impact on the model output differs among features. This pattern is mainly visible in the Ridge models, Tables 15 and 16. Titles such *imy-rA zS(w) a(w) (nw) ncwt*, *HAty-a*, and iry-pat played a dominant role in every model. On the other hand, there are features that have a low SHAP mean value in every model.

For example, feature version 2 introduced additional binary features *father_was_vizier* and *dynasty1*. None of these features turned out to have a high SHAP mean value. Firstly, these features were not chosen by the step LR model, nor LASSO LR model. Furthermore, in the models where these features were present, they achieved the lowest SHAP means values.

The feature *father_was_vizier* was added due to nepotistic practices in the Old Kingdom of Egypt [1], [7]. The small SHAP means values do not necessarily lead to a contradiction with the literature. More likely while a person inherited a vizier title from his father, he also inherited other titles [7], so the information *father_was_vizier* become redundant.

As can be seen in Table 16, the size of the SHAP mean values depends on the model. In models with a low input size, the features tend to have higher SHAP mean values. In the Ridge model, the impact was divided into more features, hence generally smaller SHAP mean values.

To compare the impact of features among models, the author decided to use similar approaches as in the previous section. The comparison of features among models will be done by rank. In each model, we will sort the SHAP mean values in descending order. The higher the SHAP mean value, the lower the rank. Subsequently, we will compute the mean rank and its standard deviation for each feature.

Figure 4.2 shows the average rank of all features included in any model. Each title is presented by its mean rank and by an interval representing one standard deviation. The titles with the lowest rank had the highest impact on the model outputs. The titles located in the bottom half of the figure correspond to titles that turned out to have small impact, see Tables 4 and 5 for their SHAP mean values.

Conclusion

In this chapter, we presented models evaluation on the test set. The performance of the model was measured by scores defined by the author. These scores were based

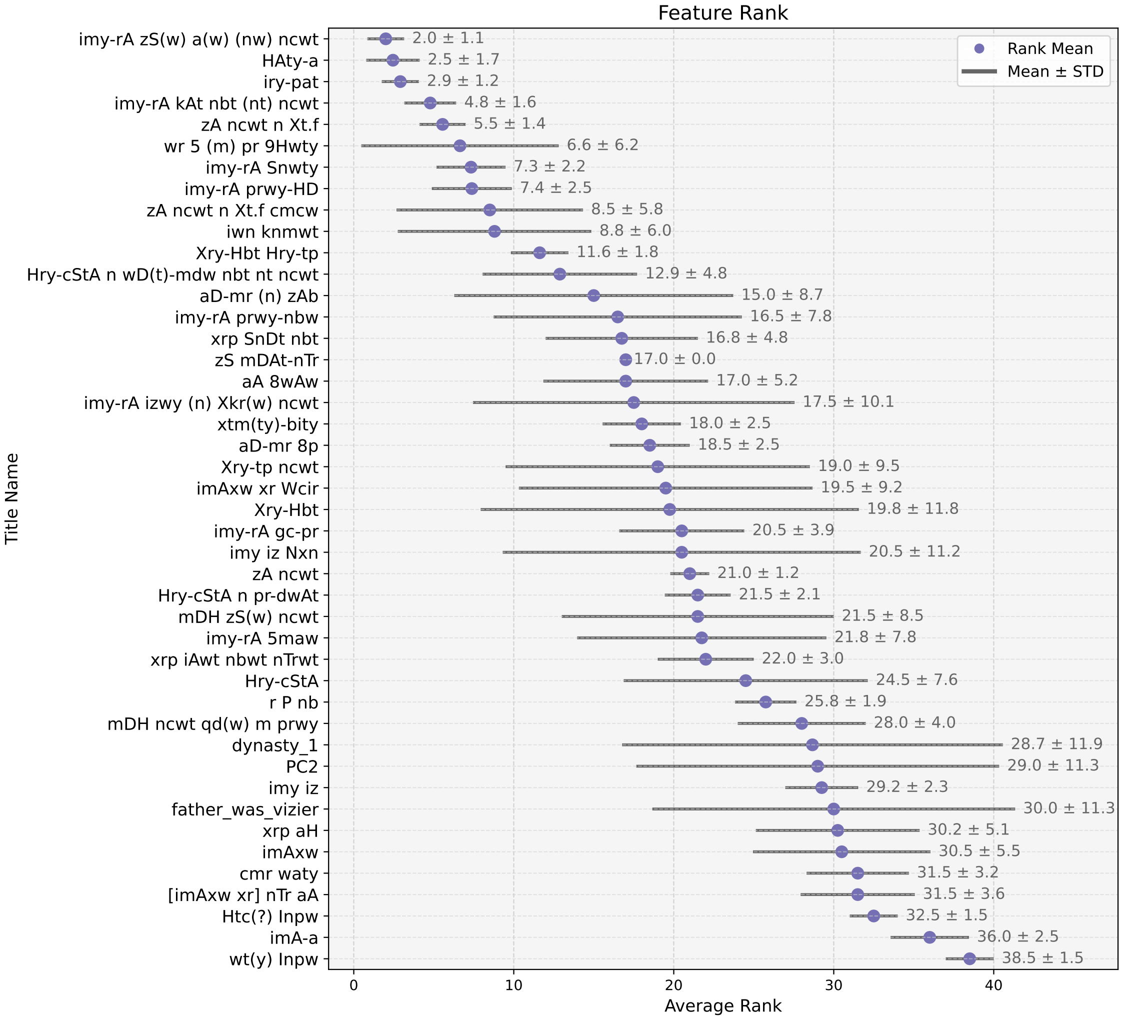


Figure 4.2: Overview of ranked titles, respectively, other features used in the model. The rank was computed from the SHAP mean. These SHAP values are presented in Attachment, Table 4, and 5. The lower the rank, the higher the prediction of the model. The dots represent the mean rank. The vertical lines represent one standard deviation interval. This interval is given as mean minus, respectively, plus standard deviation of the rank. The mean and standard deviation are also shown on the right of the deviation interval. All features named in the y-axis are described in Attachment, Table 2. For visualization of SHAP values in a particular model, see Attachment, Tables 9a–16.

on the ranked dataset. All models except one revealed good performance on test set, therefore, these models were used in following analyses of persons and titles.

The second section analyzed models predictions. Namely, highlighting these non-viziers that gained high prediction, despite these persons were not viziers. By means of ranked prediction we identified in total 30 persons which were hard to predict. Twenty persons in the train set, five in the validation set, and five in the test set.

The last section focused on title analyses. By means of SHAP values we analyzed impact of input features on the model. Based on rank, we ordered the titles according to their impact among all models. In addition, in this section, we recapped the patterns that were discovered while creating our dataset.

The results from the last section are related to publication [4]. Nevertheless, the author used different approach. He analyzed titles strongly connected to vizier by purely count scores. Performance of each title was defined by count of vizier and non-vizier holders. Our analyzes used different methodologies, we defined our datasets by means of scores similar to [4], but our results are not based on a pairwise comparison between vizier and other titles. Our approach takes into account combination of all features for a given person.

The result of this publication overlaps with our findings, but not in the same order. The author presents top fifteen titles connected to vizier. The most imporant one are *imy-rA zS(w) a(w) (nw) ncwt*, *iry-pat* which both reached the highest scores also in our results.

Finally, we would like to mention results from PCA applied on title categories. This approach was applied in Section 2.1. As a by-product of our analysis we discovered, that application of PCA revealt two main cluster of titles. According to Figure 2.3b, we see that category *family* was mainly projected onto the second principal component, while all other categories were dominant in the first principal component.

Conclusion

This thesis was created in connection with the project *Titles and bones of ancient Egyptian officials: New mathematical approach to analysing Old Kingdom data, Czech Science Foundation (GAČR), Junior Star 2024–2028, Project No. 24-10275M.*

The thesis was dedicated to the analysis of the relationship between the vizier and other titles from the Old Kingdom of Egypt. Viziers were high-ranking dignitaries [1], [8]. By examining viziers, we aimed to identify titles closely associated with them. Furthermore, we sought to discover non-vizier individuals whose titulary suggests a close connection to the vizierate.

We created our own dataset based on the Maatbase database. Using logistic regression and a multilayer perceptron, we developed twelve different models and analyzed them from multiple perspectives. First, we evaluated the performance of the models in the test set to confirm their ability to learn the given patterns, Table 4.1. Then, by analyzing the model predictions, we identified non-vizier individuals who were assigned high prediction scores despite not holding the vizier title, Figure 4.1. According to their titulary, these individuals held significant power in the Old Kingdom. Finally, using SHAP values, we analyzed the importance of each title and highlighted the most influential ones, Figure 4.2. These result were discussed in Chapter 4.

Chapter Overview

In the following, we recapitulate each chapter of this thesis. The first chapter was dedicated to introducing the Old Kingdom of Egypt in its historical context. We summarized the structure of its society, nepotistic behavior, pharaons marriage strategies, tomb architecture, and titulary of individuals. In the next section, we introduced the Maatbase database, which served as our data source. Later in this chapter, we reviewed previous research on the Old Kingdom of Egypt as well as on other societies from different spatial and temporal contexts. This research included several studies in which authors analyzed social structures using mathematical approaches.

In the second chapter, we described the machine learning methods used in our modeling. First, we introduced principal component analysis (PCA), followed by an example in which we applied PCA to Egyptological data. The resulting components were later used in our models. In the second section, we presented logistic regression along with its statistical properties. We then described the multilayer perceptron and its regularization techniques. The next section was dedicated to SHAP values, which played a crucial role in both model training and post-model analysis. This section concluded with a demonstration of SHAP value analysis on artificial data. The final section of the chapter focused on methods used for model evaluation. We defined custom scores to intuitively measure model performance without requiring a predefined threshold. In addition, we introduced a custom SHAP plot designed specifically for analyzing SHAP values in binary data.

The third chapter described the process of model construction. First, we defined the objectives of our modeling. In the following steps, we introduced our target feature, followed by a detailed description of the feature selection process. We created two datasets used in our modeling, which are summarized in Table 2. The final section of this chapter focused on model selection, where we explained how we derived our twelve final models. Since we designed our dataset so that each input feature would have a positive impact on model prediction, we monitored SHAP values immediately after the training process. This monitoring was crucial to avoid model overfitting.

The fourth chapter summarized our findings. In the first section, we comment on the performance of the models in the test set, Table 4.1. In the next section, we analyzed our models from two perspectives. First, we identified non-vizier individuals who achieved high scores in our models, Figure 4.1. These individuals had a titulary similar to that of viziers, although there is no evidence that they actually held the vizier title. Second, we used SHAP values to analyze the importance of each feature in the model's predictions. This analysis allowed us to highlight the most important titles, Figure 4.2.

This thesis concludes with the reference section and an attachment. In the attachment, the author presents several tables and figures that complement the preceding chapters. References to these tables and figures are made throughout the main text.

Summary of Thesis Assignment

We will conclude this chapter by reflecting on the assignment of this thesis.

The author analyzed the Maatbase database. Its main characteristics are presented in Table 1.1. Furthermore, during feature selection in Section 3.3, several title analyses were performed, which led to the creation of two datasets, see Table 2.

Regarding the models, in total six logistic regression models were implemented, each with different input features. These models are summarized in Table 3. In addition, six multilayer perceptron models were created, again each with different input features. For their architecture, see Table 3.9.

Furthermore, the creation of these models was described in detail, see Section 3.4. The theoretical background for all implemented models is provided in Chapter 2. Additionally, in Section 2.1, the application of PCA was demonstrated on real Egyptological data. Also, Section 2.4 demonstrates the analysis of SHAP values on artificial data.

All results were presented in Chapter 4, namely the performance of the models, the summary of people with the highest predictions, and the summary of the most important features. See Table 4.1, respectively Figures 4.1 and 4.2. The list of these figures and tables is not complete, many complementary materials can be found in the Attachment.

In closing, the author invites interested individuals to visit the GitHub repository[1] where the core scripts have been published.

[1] https://github.com/pavelspal/Titles-of-the-Old-Kingdom-of-Egypt

References

[1] M. Bárta, V. Dulíková, *Ty: An extraordinary courtier of his king: Social network analysis, status race and punctuated equilibria in a complex society.* In 'Addressing the Dynamics of Change in Ancient Egypt: Complex Network Analysis'. Charles University Faculty of Arts, 2020, 1–28.

[2] M. Bárta, V. Dulíková, R. Mařík, M. Cibuľa, *Modelling the Dynamics of Ancient Egyptian State During the Old Kingdom Period: Hidden Markov Models and Social Network Analysis.* Zeitschrift für Ägyptische Sprache und Altertumskunde 149, 2022, 1–16.

[3] E. Hornung, R. Krauss, D. A. Warburton, *Ancient Egyptian Chronology.* Brill, Leiden, 2006.

[4] V. Dulikova, R. Marik, *Uncovering Old Kingdom society arrangement: Detection of powerful dignitaries using complex network analysis.* In 'Handbook of Digital Egyptology: Texts'. Editorial Universidad de Alcalá, 2021, 69–102.

[5] V. Dulíková, *The Reign of King Nyuserre and its Impact on the Development of the Egyptian State: A Multiplier Effect Period during the Old Kingdom.* Charles University, unpublished PhD thesis, 2016.

[6] R. Mařík, V. Dulíková, *Cyber-Egyptology: An overview of tools: Cybernetics, artificial intelligence, complex networks.* In 'Addressing the Dynamics of Change in Ancient Egypt: Complex Network Analysis'. Charles University Faculty of Arts, 2020, 29–70.

[7] V. Dulíková, R. Mařík, *Complex network analysis in Old Kingdom society: a nepotism case.* In 'Abusir and Saqqara in the Year 2015'. Charles University Faculty of Arts, 2017, 63–83.

[8] R. Marik, *Feature Space Decomposition using Information Theory.* In 'Proceedings of the 2018 International Conference on Algorithms, Computing and Artificial Intelligence'. Association for Computing Machinery, 2018, 1–10.

[9] R. Marik, M. Cibula, *Multi-attribute sequence interpretation using HMM.* In '2017 4th International Conference on Systems and Informatics (ICSAI)'. IEEE, 2017, 1529–1534.

[10] V. Dulikova, R. Marik, M. Bárta, M. Cibula, *Invisible History: Hidden Markov Model of Old Kingdom Administration Development and its Trends.* In 'Edal VI: Old Kingdom Art and Archaeology International Conference 2017'. Pontremoli Editore, 2019, 226–237.

[11] J. F. Padgett, *Open Elite? Social Mobility, Marriage, and Family in Florence, 1282–1494.* Renaissance Quarterly 63, 2010, 357–411.

[12] J. S. Long, *Regression Models for Categorical and Limited Dependent Variables.* SAGE Publications, Thousand Oaks, 1997.

[13] M. Catino, S. Rocchi, G. Vittucci Marzetti, *The network of interfamily marriages in 'Ndrangheta.* Social Networks 68, 2022, 318–329.

[14] D. Montgomery, E. Peck, G. Vining, *Introduction to Linear Regression Analysis.* John Wiley & Sons, New York, 2021.

[15] J. Y. Chan, S. M. H. Leow, K. T. Bea, *et al.*, *Mitigating the Multicollinearity Problem and Its Machine Learning Approach: A Review.* Mathematics 10, 2022, 1283.

[16] I. T. Jolliffe, *Principal Component Analysis.* Springer, New York, 2002.

[17] F. L. Gewers, G. R. Ferreira, H. F. D. Arruda, *et al.*, *Principal Component Analysis: A Natural Approach to Data Exploration.* ACM Comput. Surv. 54, 2021, 1–34.

[18] E. Andersen, *Introduction to the Statistical Analysis of Categorical Data.* Springer, Berlin, 1997.

[19] P. Dunn, G. Smyth, *Generalized Linear Models With Examples in R.* Springer, New York, 2018.

[20] R. Myers, D. Montgomery, G. Vining, T. Robinson, *Generalized Linear Models: with Applications in Engineering and the Sciences.* Wiley, New Jersey, 2012.

[21] H. Akaike, *A new look at the statistical model identification.* IEEE Transactions on Automatic Control 19, 1974, 716–723.

[22] G. Claeskens, N. L. Hjort, *Model Selection and Model Averaging.* Cambridge University Press, Cambridge, 2008.

[23] J. H. Friedman, T. Hastie, R. Tibshirani, *Regularization Paths for Generalized Linear Models via Coordinate Descent.* Journal of Statistical Software 33, 2010, 1–22.

[24] T. Hastie, R. Tibshirani, J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction, Second Edition.* Springer, New York, 2009.

[25] T. Hastie, *Statistical Models in S.* Chapman & Hall, London, 1992.

[26] I. Goodfellow, Y. Bengio, A. Courville, *Deep Learning.* MIT Press, Cambridge, 2016.

[27] A. Krizhevsky, I. Sutskever, G. E. Hinton, *ImageNet classification with deep convolutional neural networks.* Communications of the ACM 60, 2017, 84–90.

[28] X. Glorot, A. Bordes, Y. Bengio, *Deep Sparse Rectifier Neural Networks.* In 'Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics'. PMLR, 2011, 315–323.

[29] A. L. Maas, A. Y. Hannun, A. Y. Ng, *Rectifier Nonlinearities Improve Neural Network Acoustic Models.* In 'Proceedings of the 30th International Conference on Machine Learning'. JMLR.org, 2013, 28.

[30] D. P. Kingma, J. Ba, *Adam: A Method for Stochastic Optimization.* International Conference on Learning Representations 1, 2015, 13.

[31] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, R. Salakhutdinov, *Dropout: A Simple Way to Prevent Neural Networks from Overfitting*. Journal of Machine Learning Research 15, 2014, 1929–1958.

[32] L. S. Shapley, *A Value for n-person Games*. In 'Contributions to the Theory of Games volume II'. Princeton University Press, 1953, 307–317.

[33] S. M. Lundberg, S.-I. Lee, *A Unified Approach to Interpreting Model Predictions*. Advances in Neural Information Processing Systems 30, 2017, 4765–4774.

[34] C. Molnar, *Interpretable Machine Learning: A Guide For Making Black Box Models Explainable*. Leanpub, 2022.

[35] S. Masís, *Interpretable Machine Learning with Python: Learn to build interpretable high-performance models with hands-on real-world examples*. Packt Publishing, Birmingham, 2021.

[36] P. Flach, *Machine Learning: The Art and Science of Algorithms that Make Sense of Data*. Cambridge University Press, Cambridge, 2012.

[37] K. Murphy, *Machine Learning: A Probabilistic Perspective*. MIT Press, Massachusetts, 2012.

[38] N. V. Chawla, K. W. Bowyer, L. O. Hall, W. P. Kegelmeyer, *SMOTE: synthetic minority over-sampling technique*. Journal of artificial intelligence research 16, 2002, 321–357.

[39] Y. Elor, H. Averbuch-Elor, *To SMOTE, or not to SMOTE?* arXiv, 2022, 1–10.

[40] R. Van den Goorbergh, M. van Smeden, D. Timmerman, B. Van Calster, *The harm of class imbalance corrections for risk prediction models: illustration and simulation using logistic regression*. Journal of the American Medical Informatics Association 29, 2022, 1525–1534.

Attachments

Jones ID	Title name	V1	V2	Translation of the title
10/39	imA-a	Yes	Yes	gracious of arm Priest responsible for handing...
11/42	imAxw	Yes	Yes	the honoured one, revered, venerated, the one ...
132–133/522	imy-rA prwy-nbw	Yes	Yes	overseer of the two houses of gold
133–134/524	imy-rA prwy-HD	Yes	Yes	overseer of the two treasuries
19–20/89	imAxw xr Wcir	Yes	Yes	revered with Osiris
246–247/896	imy-rA 5maw	Yes	Yes	overseer of Upper Egypt
254–255/923	imy-rA Snwty	Yes	Yes	overseer of the two granaries
262–263/950	imy-rA kAt nbt (nt) ncwt	Yes	Yes	overseer of all works of the king
269–270/969	imy-rA gc-pr	Yes	Yes	overseer of an troop-house (of workers)/work p...
30/142	[imAxw xr] nTr aA	Yes	Yes	revered with the great god
315/1157	iry-pat	Yes	Yes	hereditary prince/nobleman
351/1308	aA 8wAw	Yes	Yes	assistant of Duau
399/1471	wr 5 (m) pr 9Hwty	Yes	Yes	Greatest of the Five in the temple of Thoth
49/247	imy iz	Yes	Yes	he who is in the iz-bureau, councillor
490–491/1831	r P nb	Yes	Yes	"mouth of every Butite"
496–497/1858	HAty-a	Yes	Yes	count
49–50/248	imy iz Nxn	Yes	Yes	councillor of Nekhen(?)
609/2233	Hry-cStA	Yes	Yes	privy to the secret
617–618/2265	Hry-cStA n wD(t)-mdw nbt nt ncwt	Yes	Yes	master of secrets of all commands of the king
620–621/2275	Hry-cStA n pr-dwAt	Yes	Yes	secretary/privyto the secret of the House of ...
67–69/310	imy-rA izwy (n) Xkr(w) ncwt	Yes	Yes	overseer of the two bureaux of the royal regal...
707/2579	xrp aH	Yes	Yes	director of the aH-palace

Table 2 continued from previous page

Jones ID	Title name	V1	V2	Translation of the title
751/2737	xrp SnDt nbt	Yes	Yes	director of every kilt According to Strudwick ...
763–764/2775	xtm(ty)-bity	Yes	Yes	sealer of the King of Lower Egypt
781/2848	Xry-Hbt	Yes	Yes	lector priest The many titles that were append...
784/2860	Xry-Hbt Hry-tp	Yes	Yes	chief lector priest
788/2874	Xry-tp ncwt	Yes	Yes	royal chamberlain
799/2911	zA ncwt	Yes	Yes	king's son
799/2912	zA ncwt n Xt.f	Yes	Yes	king's son of his body
799/2914	zA ncwt n Xt.f cmcw	Yes	Yes	king's eldest son of his body
857–858/3132	zS mDAt-nTr	Yes	Yes	scribe of the god's book a scribal function, p...
892/3268	cmr waty	Yes	Yes	sole companion
209–210/780	imy-rA zS(w) a(w) (nw) ncwt	Yes		overseer of the royal document scribes
684–685/2501	Htc(?) Inpw	Yes		... of Anubis(?), (precise reading unknown, me...
695/2541	xrp iAwt nbwt nTrwt	Yes		director of every divine office According to S...
806/2947	aD-mr (n) zAb	Yes		administrator/boundary official of the king ju...
365/1348	aD-mr 8p		Yes	administrator/boundary official of Dep (Buto)
405–406/1493	wt(y) Inpw		Yes	embalmer of Anubis
464–465/1733	mDH ncwt qd(w) m prwy		Yes	king's architect in the two houses royal maste...
467–468/1739	mDH zS(w) ncwt		Yes	overseer of the royal scribes, master architec...
6/22	iwn knmwt		Yes	support of kenemut; pillar of the knmt-folk
	father_was_vizier		Yes	binary variable encoding whether the father was vizier
	dynasty1		Yes	encoding period between the mid-Fifth and mid-Sixth dynasties
	PC2		Yes	the second component of PCA on title categories

Table 2: Overview of features selected during feature selection. Columns *V1* and *V2* indicate whether a given feature was included in feature version 1 and 2, respectively.

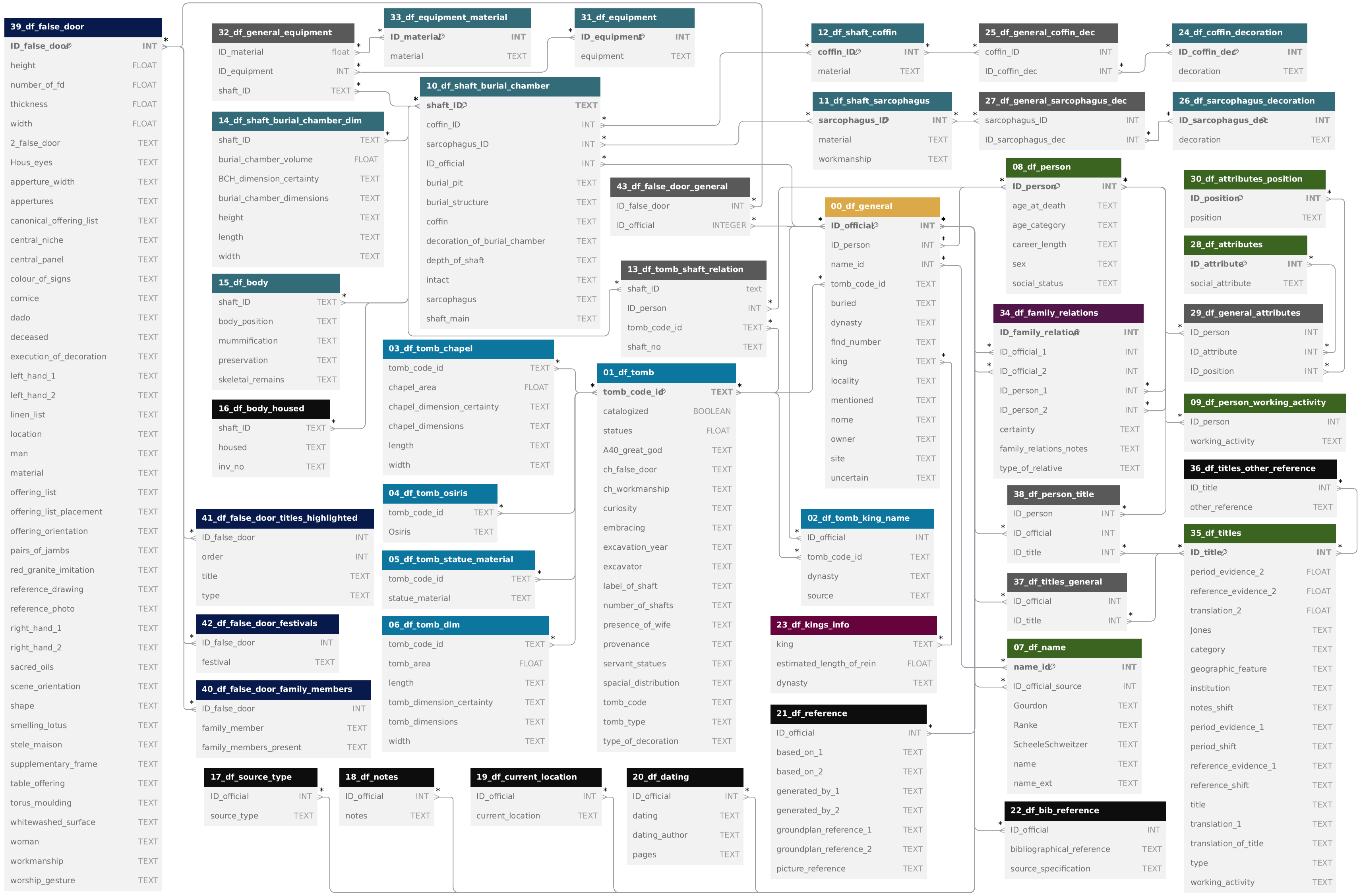


Figure 3: Schema of the Maatbase database. The symbol * denotes that the relation does not necessarily have to be many-to-many.

Model	**LR**					
Model type	**Ridge**		**LASSO**		**Step**	
Feature version	**2**	**1**	**2**	**1**	**2**	**1**
Feature name	**Estimated coefficients**					
(Intercept)	-5.060	-5.204	-5.454	-5.754	-6.687	-6.707
imy-rA zS(w) a(w) (nw) ncwt		1.396		3.481		5.992
wr 5 (m) pr 9Hwty	1.284	1.438	1.147	1.810	3.564	3.677
HAty-a	1.011	1.037	2.822	2.597	3.552	3.650
zA ncwt n Xt.f	0.874	0.945	0.845	0.981	2.762	3.577
iry-pat	1.075	1.134	1.979	2.164	3.017	2.829
iwn knmwt	0.570		1.370		3.253	
imy-rA Snwty	0.725	0.684	0.911	0.510	3.014	
imy-rA kAt nbt (nt) ncwt	0.852	0.840	1.658	0.965	2.544	
zA ncwt n Xt.f cmcw	1.250	1.356	1.375	1.848		
imy-rA prwy-HD	0.763	0.655	1.140	0.613		
Hry-cStA n wD(t)-mdw nbt nt ncwt	1.088	0.967	1.356	0.577		
Xry-Hbt Hry-tp	0.549	0.606	0.146	0.384		
aD-mr (n) zAb		0.326		0.194		
imy-rA izwy (n) Xkr(w) ncwt	0.502	0.545		0.041		
xtm(ty)-bity	0.477	0.494				
imy-rA 5maw	0.389	0.458				
imy-rA gc-pr	0.737	0.457				
imy iz Nxn	0.584	0.405				
xrp SnDt nbt	0.411	0.398				
zA ncwt	0.321	0.369				
zS mDAt-nTr	0.366	0.356				
aA 8wAw	0.282	0.310				
Xry-tp ncwt	0.280	0.277				
imy-rA prwy-nbw	0.248	0.272				
imAxw xr Wcir	0.263	0.251				
xrp aH	0.220	0.246				
cmr waty	0.219	0.235				
Hry-cStA n pr-dwAt	0.310	0.232				
xrp iAwt nbwt nTrwt		0.222				
r P nb	0.222	0.163				
Xry-Hbt	0.126	0.122				
Hry-cStA	0.170	0.120				
[imAxw xr] nTr aA	0.094	0.074				
imA-a	0.048	0.039				
Htc(?) Inpw		0.034				
imAxw	0.049	0.033				
imy iz	0.168	0.023				
mDH ncwt qd(w) m prwy	1.026					
mDH zS(w) ncwt	0.626					
aD-mr 8p	0.480					
father_was_vizier	0.052					
PC2	-0.005					
dynasty1	-0.007					
wt(y) Inpw	-0.013					

Table 3: Overview of **estimated coefficients** from **LR models**. If the value is not present, the feature was not used in the model.

Feature version	1					
Model type	Ridge		LASSO		step	
Model	MLP	LR	MLP	LR	MLP	LR
Feature name	Mean SHAP value					
imy-rA zS(w) a(w) (nw) ncwt	0.2521	0.0899	0.4767	0.2535	0.6855	0.4138
wr 5 (m) pr 9Hwty	0.0303	0.1402	0.1480	0.1504	0.0639	0.2457
zA ncwt n Xt.f	0.0920	0.0856	0.1991	0.0755	0.2201	0.2322
HAty-a	0.2572	0.0807	0.4823	0.1736	0.5828	0.1863
iry-pat	0.2898	0.0944	0.2304	0.1692	0.3134	0.1692
zA ncwt n Xt.f cmcw	0.0355	0.1064	0.0613	0.1241		
imy-rA kAt nbt (nt) ncwt	0.2081	0.0637	0.3743	0.0646		
Hry-cStA n wD(t)-mdw nbt nt ncwt	0.0237	0.0407	0.0704	0.0449		
imy-rA prwy-HD	0.0943	0.0541	0.0685	0.0425		
imy-rA Snwty	0.0885	0.0523	0.1006	0.0326		
Xry-Hbt Hry-tp	0.0592	0.0522	0.0778	0.0310		
imy-rA izwy (n) Xkr(w) ncwt	0.0054	0.0563	0.0117	0.0040		
aD-mr (n) zAb	0.0625	0.0064	0.2079	0.0038		
imy iz Nxn	0.0060	0.0457				
xrp SnDt nbt	0.0357	0.0423				
imy-rA 5maw	0.0040	0.0299				
xtm(ty)-bity	0.0365	0.0272				
zS mDAt-nTr	0.0359	0.0260				
aA 8wAw	0.0712	0.0241				
xrp iAwt nbwt nTrwt	0.0197	0.0241				
zA ncwt	0.0335	0.0223				
imy-rA gc-pr	0.0182	0.0195				
imy-rA prwy-nbw	0.0713	0.0153				
Hry-cStA n pr-dwAt	0.0235	0.0151				
xrp aH	0.0058	0.0124				
r P nb	0.0077	0.0121				
imAxw xr Wcir	0.0745	0.0070				
Xry-tp ncwt	0.0746	0.0067				
cmr waty	-0.0019	0.0067				
Xry-Hbt	0.0889	0.0036				
Htc(?) Inpw	0.0000	0.0033				
Hry-cStA	0.0477	0.0028				
imy iz	0.0073	0.0021				
imA-a	-0.0050	0.0020				
[imAxw xr] nTr aA	0.0117	0.0014				
imAxw	0.0224	0.0007				

Table 4: Overview of **SHAP values** for **feature version 1**. The numbers represent the mean of SHAP values, but only among the observations where the given feature was present.

Feature version	**2**					
Model type	**Ridge**		**LASSO**		**step**	
Model	**MLP**	**LR**	**MLP**	**LR**	**MLP**	**LR**
Feature name	**Mean SHAP value**					
HAty-a	0.2570	0.0783	0.4278	0.2157	0.4560	0.2014
wr 5 (m) pr 9Hwty	0.0386	0.1200	0.0716	0.0911	0.0995	0.2009
imy-rA Snwty	0.1193	0.0593	0.1627	0.0683	0.3777	0.1883
iry-pat	0.3546	0.0879	0.3374	0.1650	0.3906	0.1793
zA ncwt n Xt.f	0.0900	0.0817	0.1080	0.0704	0.1166	0.1657
imy-rA kAt nbt (nt) ncwt	0.2268	0.0700	0.3040	0.1234	0.2927	0.1227
iwn knmwt	0.1172	0.0251	0.3786	0.0603	0.5010	0.1030
father_was_vizier	-0.0096	0.0006	-0.0151		0.0080	
dynasty_1	0.0005	-0.0000	0.0004		0.0049	
PC2	-0.0000	0.0000	-0.0001		-0.0000	
zA ncwt n Xt.f cmcw	0.0456	0.1072	0.0704	0.0918		
imy-rA prwy-HD	0.1592	0.0586	0.2250	0.0708		
Hry-cStA n wD(t)-mdw nbt nt ncwt	0.0517	0.0260	0.0739	0.0484		
Xry-Hbt Hry-tp	0.0425	0.0481	0.0613	0.0110		
imy iz Nxn	0.0048	0.0658				
imy-rA izwy (n) Xkr(w) ncwt	0.0067	0.0591				
xrp SnDt nbt	0.0247	0.0431				
mDH zS(w) ncwt	0.0090	0.0362				
imy-rA 5maw	0.0236	0.0354				
imy-rA gc-pr	0.0337	0.0338				
aD-mr 8p	0.0326	0.0307				
zS mDAt-nTr	0.0364	0.0275				
xtm(ty)-bity	0.0293	0.0269				
zA ncwt	0.0249	0.0206				
Hry-cStA n pr-dwAt	0.0343	0.0201				
r P nb	0.0151	0.0175				
aA 8wAw	0.0514	0.0169				
mDH ncwt qd(w) m prwy	0.0055	0.0169				
imy-rA prwy-nbw	0.1154	0.0163				
imy iz	0.0117	0.0158				
xrp aH	-0.0027	0.0130				
Xry-tp ncwt	0.0792	0.0081				
cmr waty	0.0008	0.0071				
imAxw xr Wcir	0.0660	0.0070				
Hry-cStA	0.0339	0.0038				
Xry-Hbt	0.0829	0.0037				
imA-a	-0.0109	0.0025				
[imAxw xr] nTr aA	0.0110	0.0019				
imAxw	0.0201	0.0010				
wt(y) Inpw	-0.0014	-0.0013				

Table 5: Overview of **SHAP values** for **feature version 2**. The numbers represent the mean of SHAP values, but only among the observations where the given feature was present.

	Model type				step			LASSO				Ridge			
	Model				LR		MLP	LR		MLP		LR		MLP	
	Feature version				1	2	1	1	2	1	2	1	2	1	2
	ID*	Person name	Mean	Std	Ranked prediction among non-viziers										
1	3108	Ibi	2.09	1.45	5	2	4	3	1	3	1	1	1	1	1
2	4108	KA(.i)-apr(w)	2.91	1.38	2	3	2	1	3	1	5	3	5	3	4
3	3370	Nfr-cSm-PtH **	3.18	1.54	5	1	4	6	4	4	3	2	2	2	2
4	3894	Ny-kAw-Izzi : Izzy	3.45	1.13	5	3	4	4	2	2	2	5	4	4	3
5	3458	7ti : KA(.i)-Hp	4.82	1.25	5	8	4	5	5	5	4	4	3	5	5
6	4149	Ny-anx-ncwt	7.00	1.34	5	8	4	8	8	8	7	8	7	7	7
7	313	5pc-pw-PtH	7.27	2.49	5	8	4	7	6	11	6	12	9	6	6
8	65	ZTw	8.82	6.82	1	14	1	2	15	6	19	6	6	8	19
9	1353	Bxn	9.55	3.36	5	8	4	9	9	15	10	13	13	11	8
10	287	1r-Dd.f	9.55	4.91	3	5	3	14	7	7	8	15	14	14	15
11	2088	Iwn-Ra	11.55	5.80	5	8	4	9	9	15	9	20	22	15	11
12	2587	2a-mrr-PtH	14.09	3.36	14	18	13	13	18	10	17	10	17	9	16
13	3537	BAwi	14.36	9.20	5	8	4	9	9	15	11	27	33	19	18
14	598	Nzr-kAw-1r	14.55	9.10	4	7	37	16	17	18	14	7	8	20	12
15	133	...	15.55	4.11	21	17	20	18	12	12	12	21	16	12	10
16	2513	Cxm-anx-PtH	15.82	6.65	14	15	13	12	16	9	16	9	21	16	33
17	3447	5pc-pw-Mnw : 3ni	18.09	7.53	21	6	20	26	13	32	15	18	10	24	14
18	3066	0nqw : 3tti	18.82	5.02	21	25	20	23	22	22	23	14	11	13	13
19	4553	PtH-Spcc	23.45	6.85	21	25	20	22	14	29	13	34	32	28	20
20	2814	QAr : Nfr-Mry-ra	25.09	5.61	21	25	20	25	24	30	34	22	18	22	35

* *ID_person.*

** The full name is *Nfr-cSm-PtH : 5Si : WDA-HA-tti.*

Table 6: Overview of **top 20 ranked non-viziers** in the **train set**. The rank was computed based on predictions among all non-viziers in the train set. The lower the rank, the higher the prediction of the model. Column *mean* represents mean of all ranks. Similarly, column *std* represents standard deviation of the rank. This table is visualized in Figure 4.1a.

	Model type				step			LASSO				Ridge			
	Model				LR		MLP	LR		MLP		LR		MLP	
	Feature version				1	2	1	1	2	1	2	1	2	1	2
	ID_person	Person name	Mean	Std	Ranked prediction among non-viziers										
1	3473	3ni-anxw	1.55	1.04	1	1	1	1	1	3	1	4	2	1	1
2	79	Nfr	3.36	2.29	2	8	2	2	7	1	3	2	5	2	3
3	156	MA-nfr	4.09	2.47	2	5	2	3	6	2	7	1	4	4	9
4	2604	KA.i	5.00	3.26	8	11	8	8	5	4	2	3	1	3	2
5	1342	3ni	5.09	2.21	5	2	5	5	2	5	8	5	3	8	8

Table 7: Overview of **top 5 ranked non-viziers** in the **validation set**. For its description see Table 6. This table is visualized in Figure 4.1b.

	Model type				step			LASSO				Ridge			
	Model				LR		MLP	LR		MLP		LR		MLP	
	Feature version				1	2	1	1	2	1	2	1	2	1	2
	ID_person	Person name	Mean	Std	Ranked prediction among non-viziers										
1	3547	Ppy-anx wr	1.27	0.47	1	1	1	2	1	2	1	2	1	1	1
2	2584	Wcr-nTr	1.73	0.47	2	2	2	1	2	1	2	1	2	2	2
3	1340	KA(.i)-Hp : 7ti-iqr	4.18	1.08	6	4	5	5	3	5	3	5	4	3	3
4	3443	Ny-anx-Ppy-(km) : 1py-km	5.09	1.45	6	4	5	5	3	5	4	8	7	4	5
5	4412	1nw	6.64	2.94	6	4	5	5	3	5	5	9	8	12	11

Table 8: Overview of **top 5 ranked non-viziers** in the **test set**. For its description see Table 6. This table is visualized in Figure 4.1c.

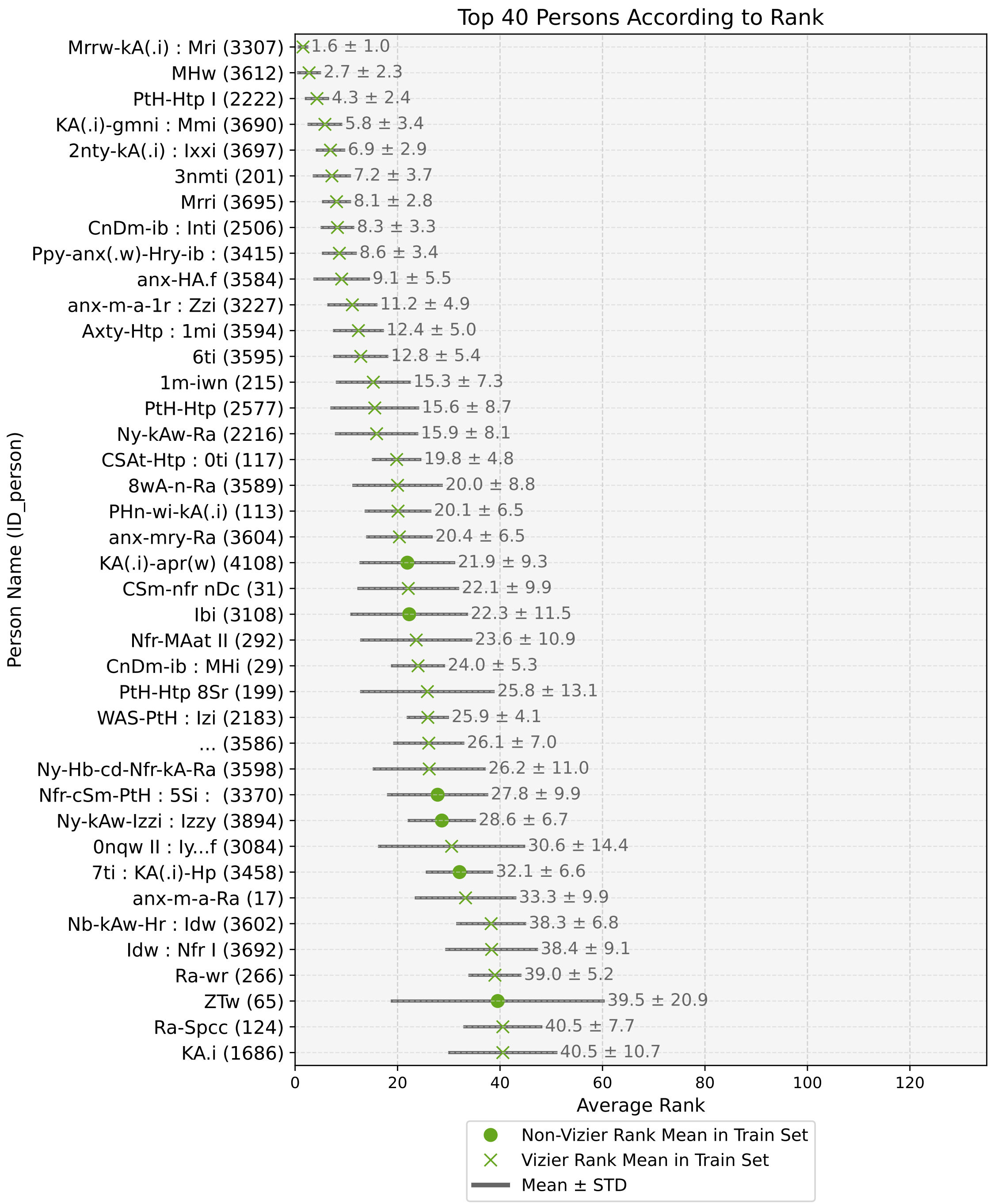


Figure 4: Overview of the **top 40 ranked persons** in the **train set**. The rank was computed based on predictions among all persons in the corresponding set. The lower the rank, the higher the prediction of the model. The dots represent the mean rank. The vertical lines represent one standard deviation interval. This interval is given as mean minus, respectively plus standard deviation of the rank. The mean and standard deviation are also shown on the right of the deviation interval. This table continues in Table 5 and 6. Due to lack of space, the names on horizontal axis are shown only up to 20th character.

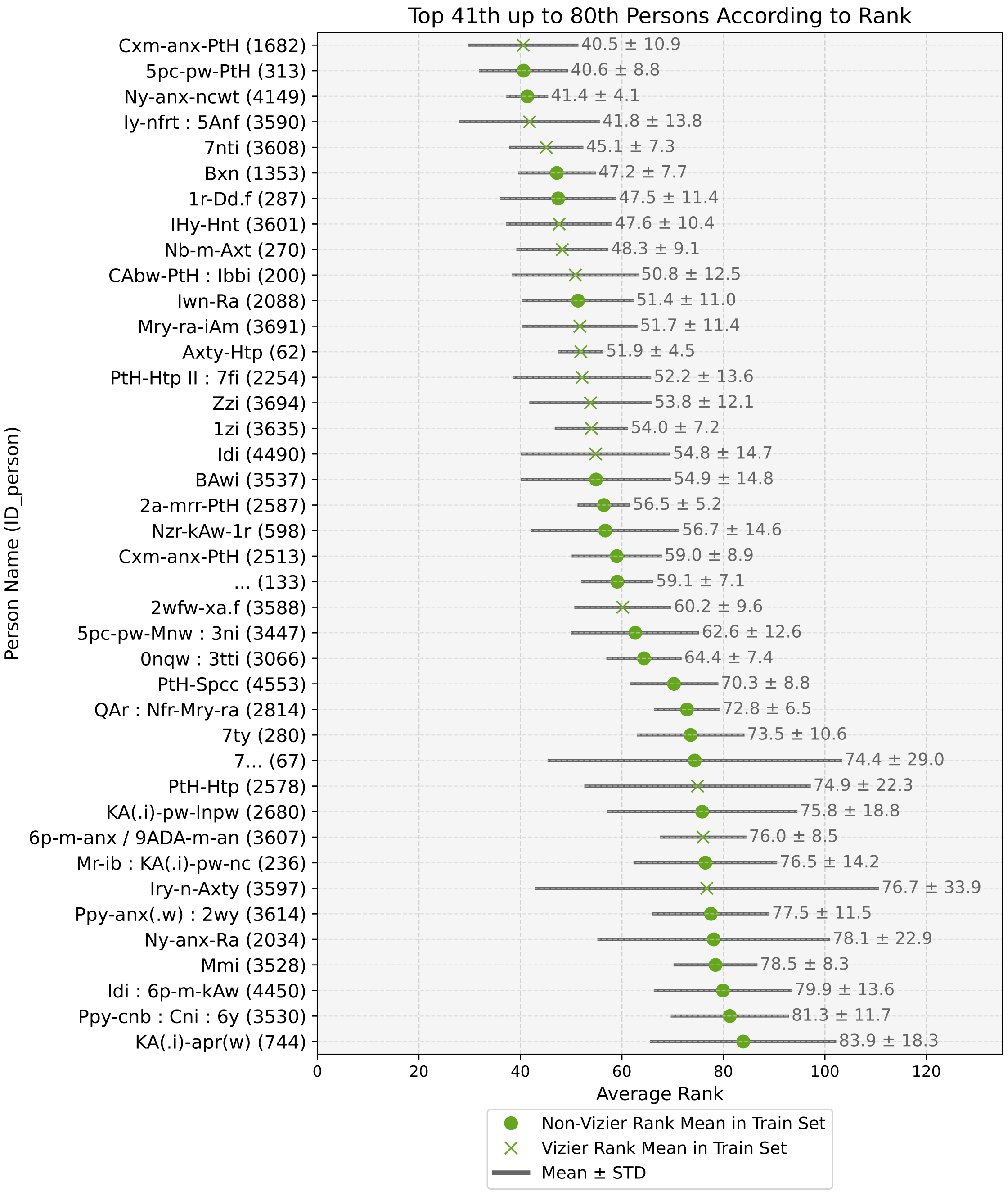


Figure 5: Overview of the **41th up to 80th top ranked persons** in the **train set**. The rank was computed based on predictions among all persons in the corresponding set. The lower the rank, the higher the prediction of the model. The dots represent the mean rank. The vertical lines represent one standard deviation interval. This interval is given as mean minus, respectively plus standard deviation of the rank. The mean and standard deviation are also shown on the right of the deviation interval. This table is a continuation of Table 4. Due to lack of space, the names on horizontal axis are shown only up to 20th character.

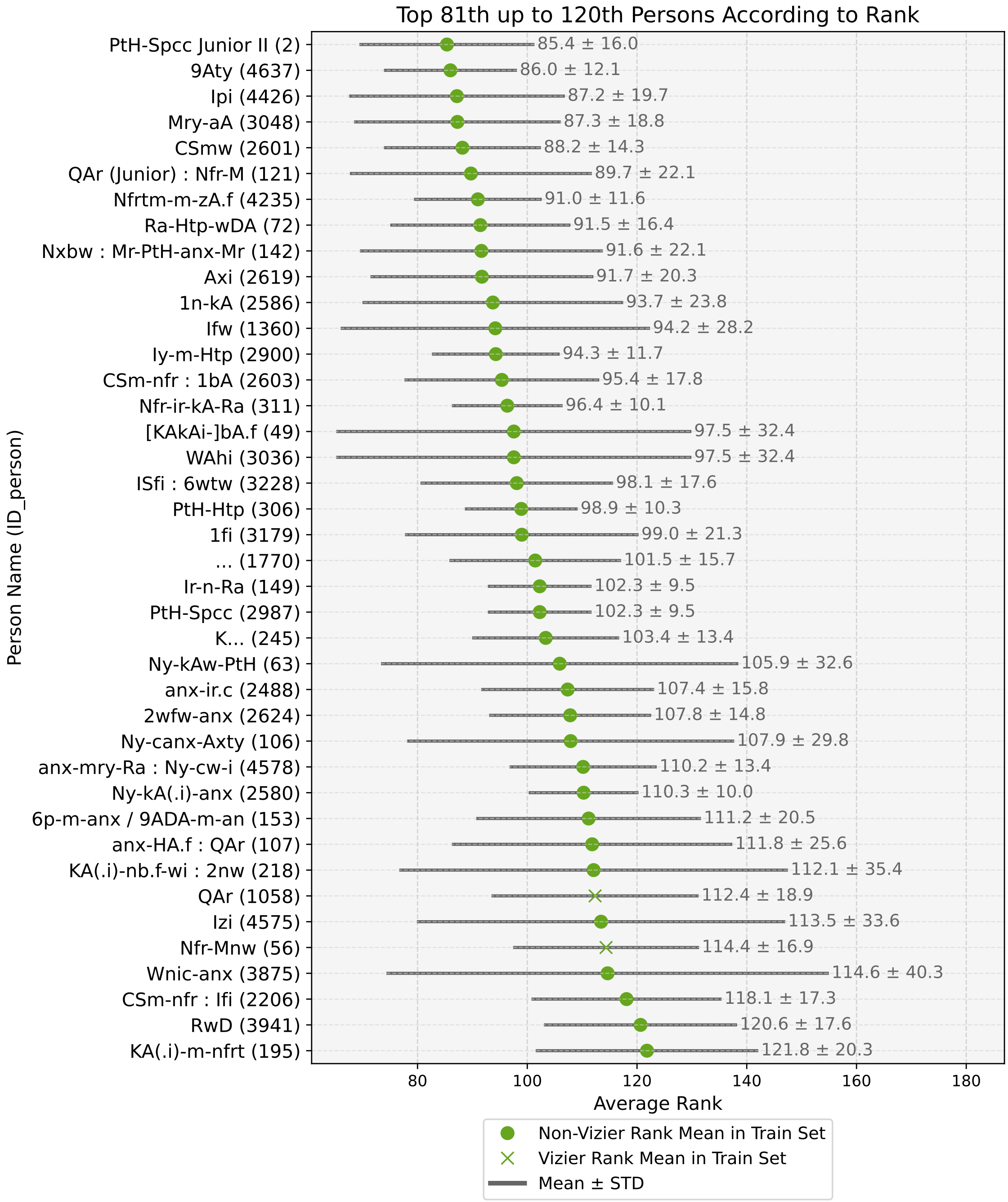


Figure 6: Overview of the **81th up to 120th top ranked persons** in the **train set**. The rank was computed based on predictions among all persons in the corresponding set. The lower the rank, the higher the prediction of the model. The dots represent the mean rank. The vertical lines represent one standard deviation interval. This interval is given as mean minus, respectively plus standard deviation of the rank. The mean and standard deviation are also shown on the right of the deviation interval. This table is a continuation of Table 4. Due to lack of space, the names on horizontal axis are shown only up to 20th character.

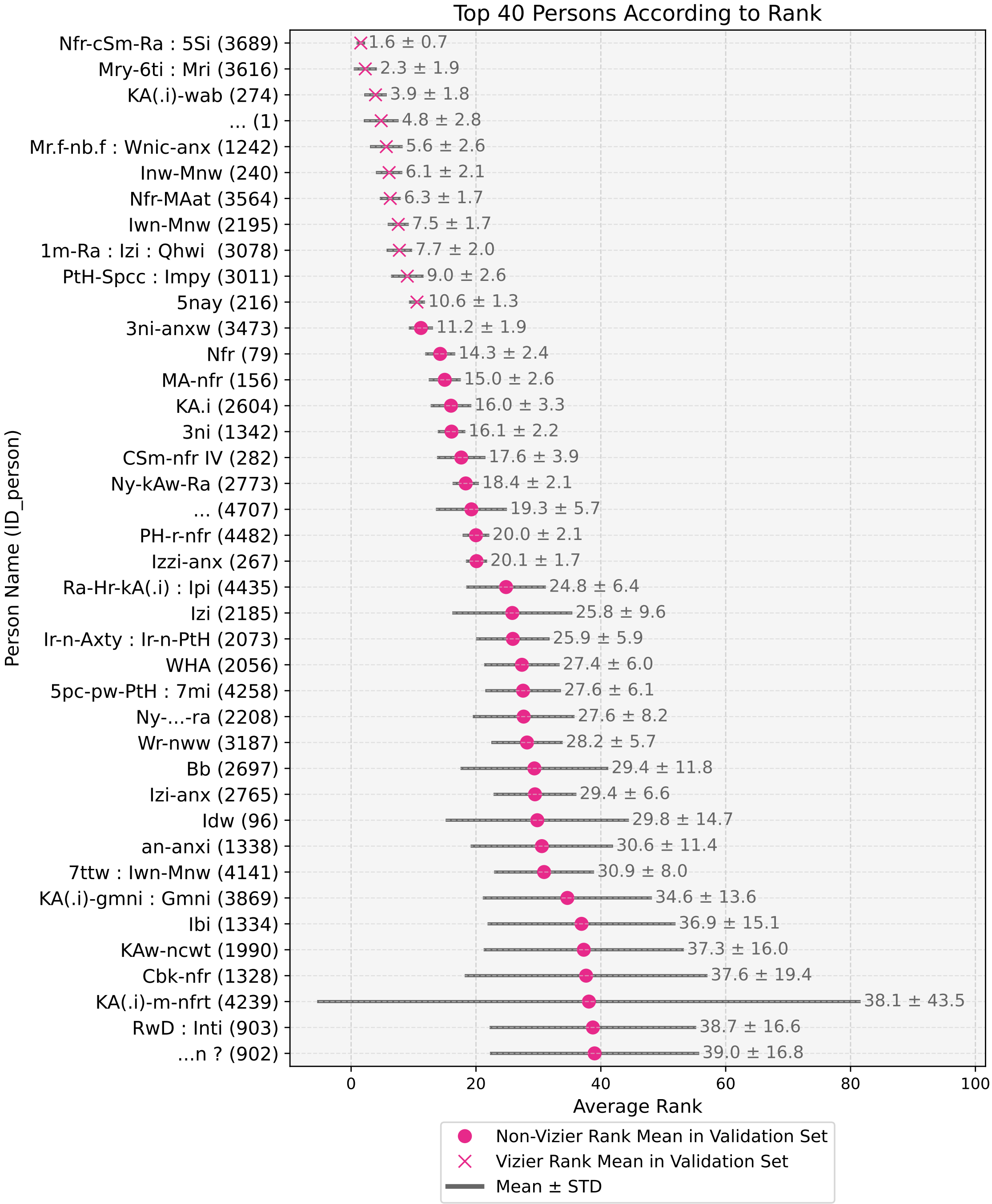


Figure 7: Overview of the **top 40 ranked persons** in the **validation set**. The rank was computed based on predictions among all persons in the corresponding set. The lower the rank, the higher the prediction of the model. The dots represent the mean rank. The vertical lines represent one standard deviation interval. This interval is given as mean minus, respectively plus standard deviation of the rank. The mean and standard deviation are also shown on the right of the deviation interval. Due to lack of space, the names on horizontal axis are shown only up to 20th character.

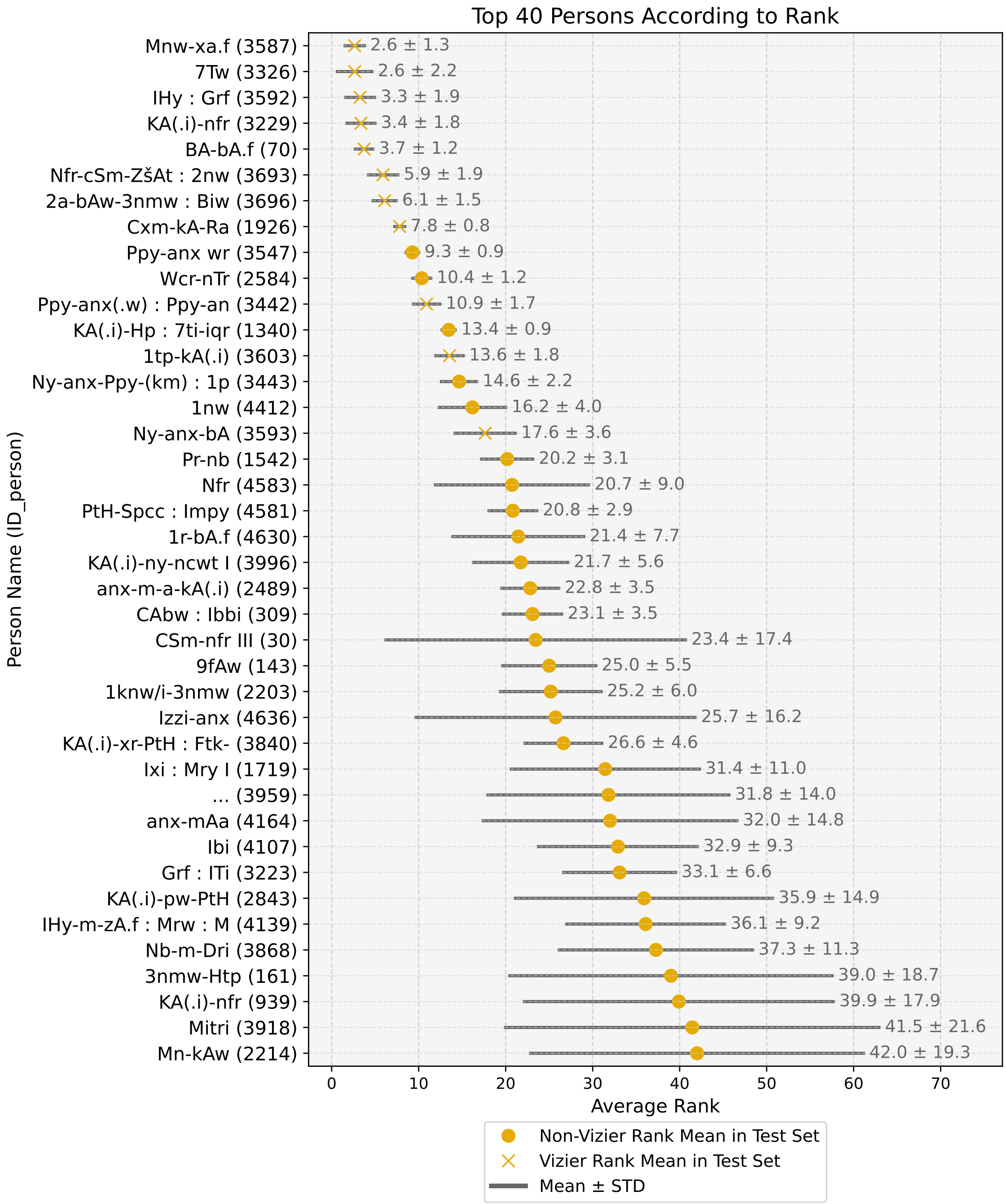
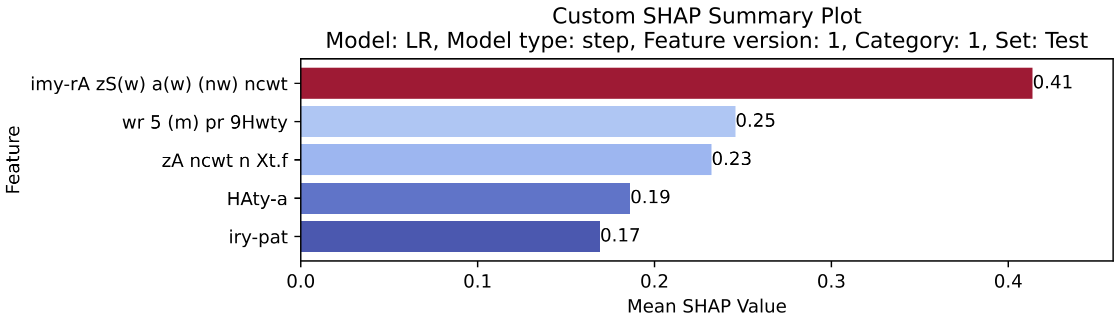
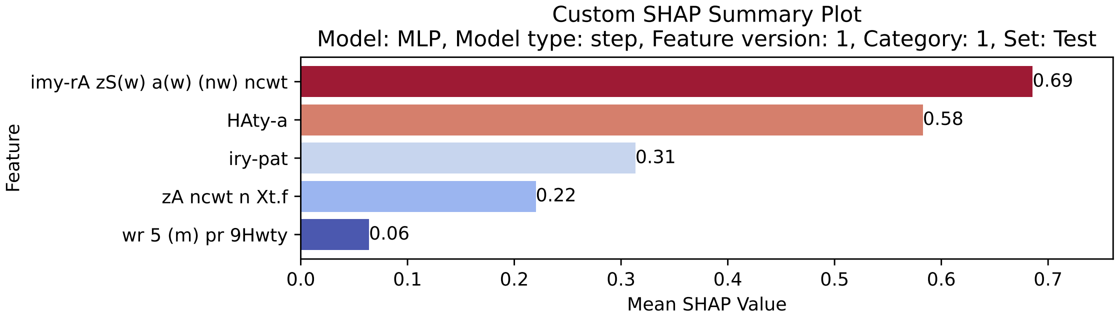


Figure 8: Overview of the **top 40 ranked persons** in the **test set**. The rank was computed based on predictions among all persons in the corresponding set. The lower the rank, the higher the prediction of the model. The dots represent the mean rank. The vertical lines represent one standard deviation interval. This interval is given as mean minus, respectively plus standard deviation of the rank. The mean and standard deviation are also shown on the right of the deviation interval. Due to lack of space, the names on horizontal axis are shown only up to 20th character.

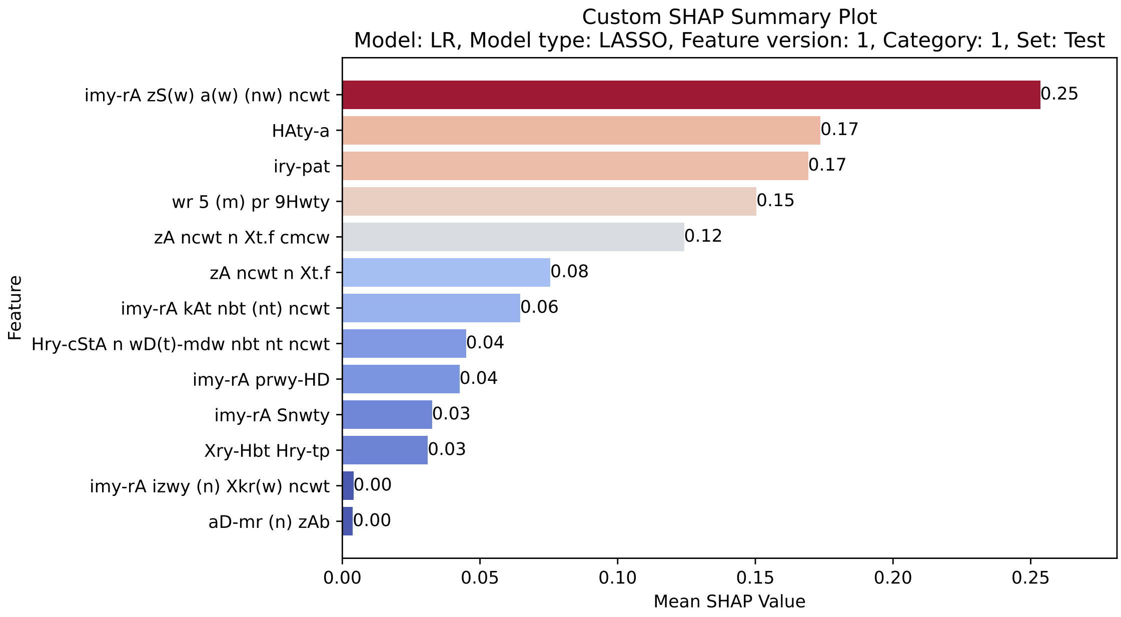


(a) Step LR model, feature version 1

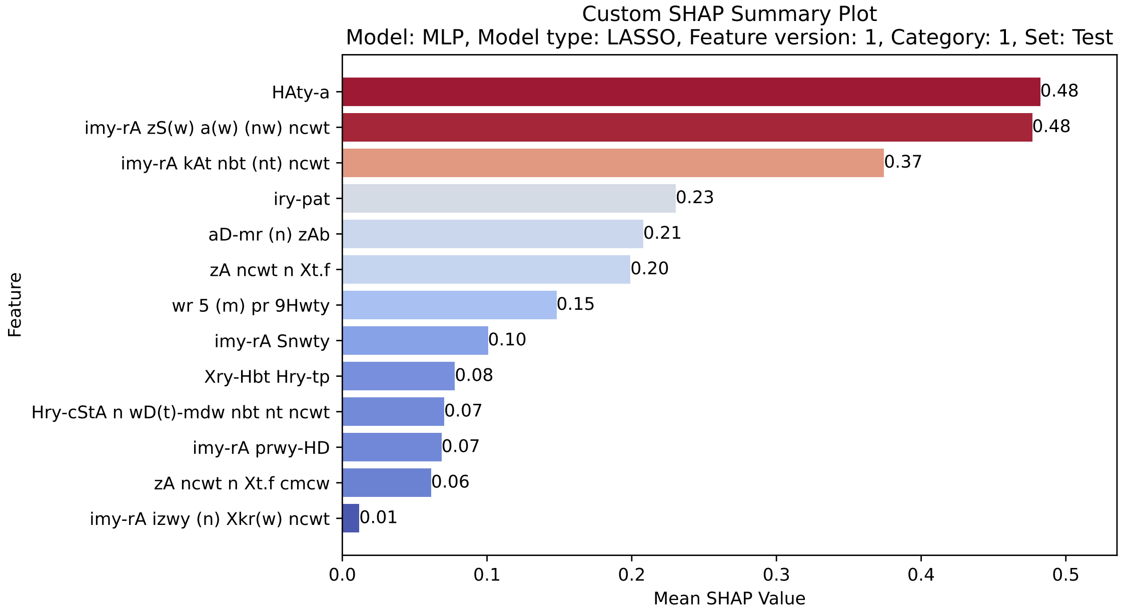


(b) Step MLP model, feature version 1

Figure 9: Custom plot of SHAP mean values for **step LR and MLP** model using **feature version 1**. This plot was defined in Section 2.5. The SHAP means were computed on test set using only observations where the given feature was present, see Figure 17a.



(a) LASSO LR model, feature version 1



(b) LASSO MLP model, feature version 1

Figure 10: Custom plot of SHAP mean values for **LASSO LR and MLP** model using **feature version 1**. This plot was defined in Section 2.5. The SHAP means were computed on test set using only observations where the given feature was present, see Figure 17b.

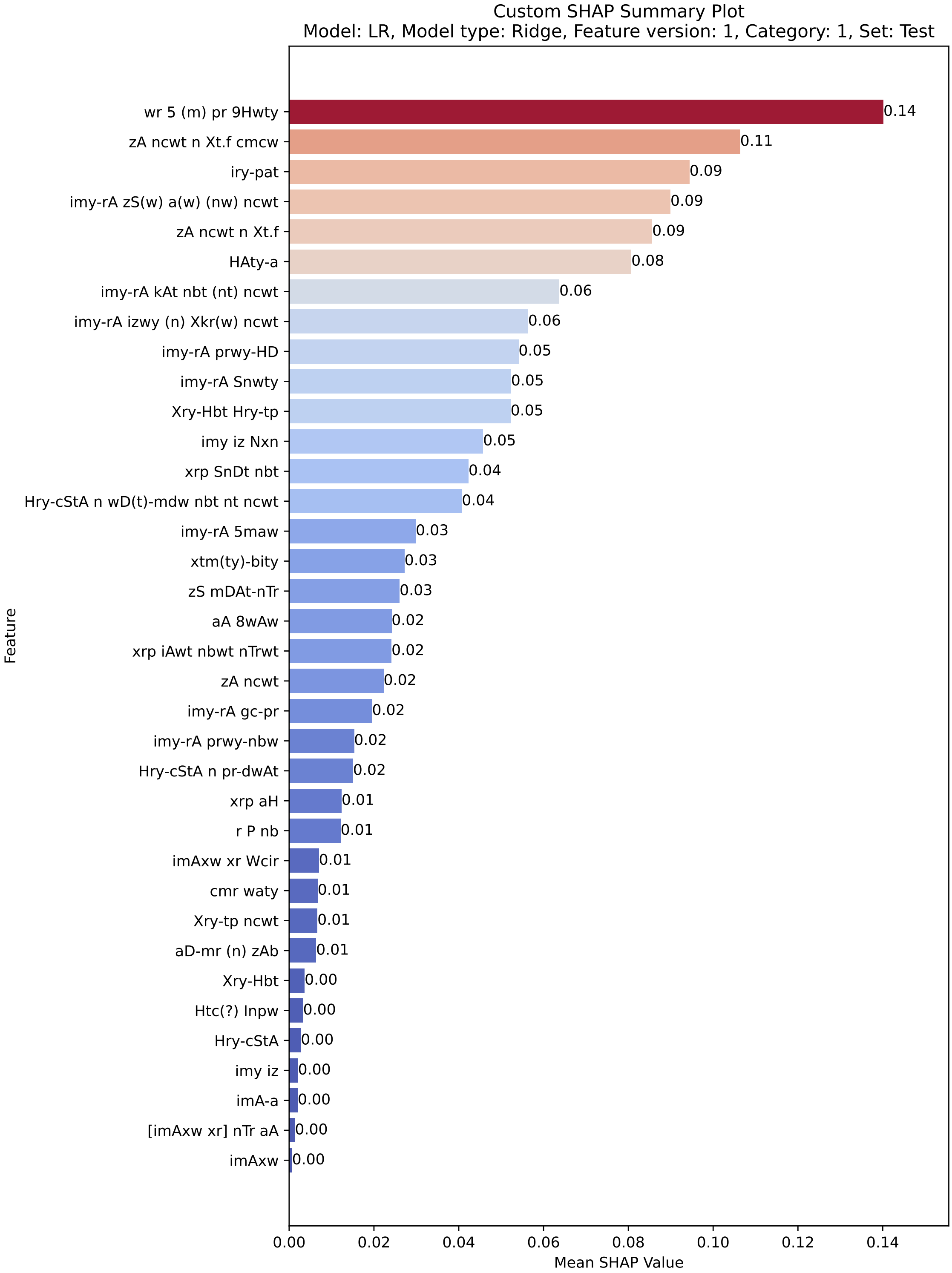


Figure 11: Custom plot of SHAP mean values for **Ridge LR** model using **feature version 1**. This plot was defined in Section 2.5. The SHAP means were computed on test set using only observations where the given feature was present.

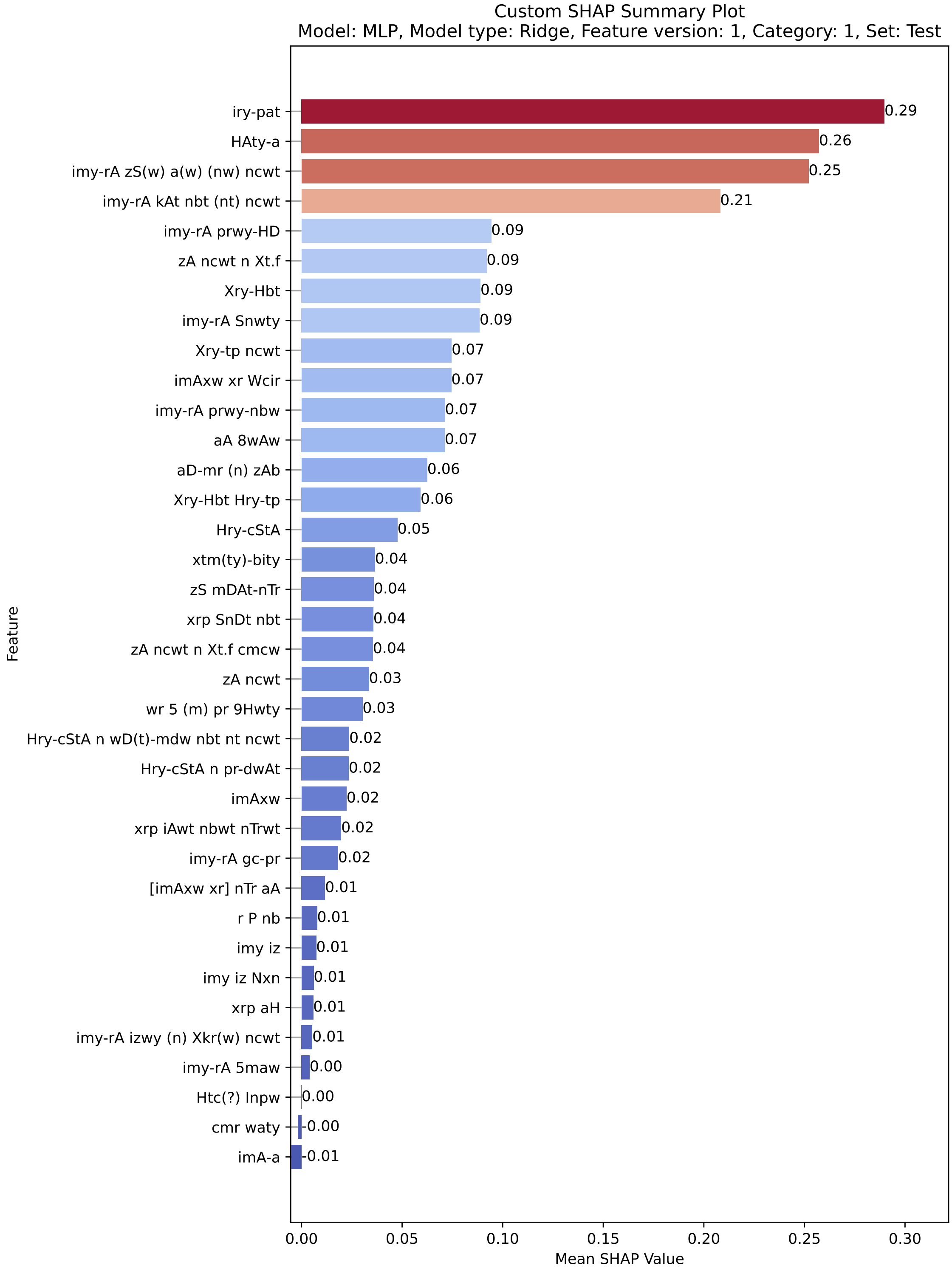
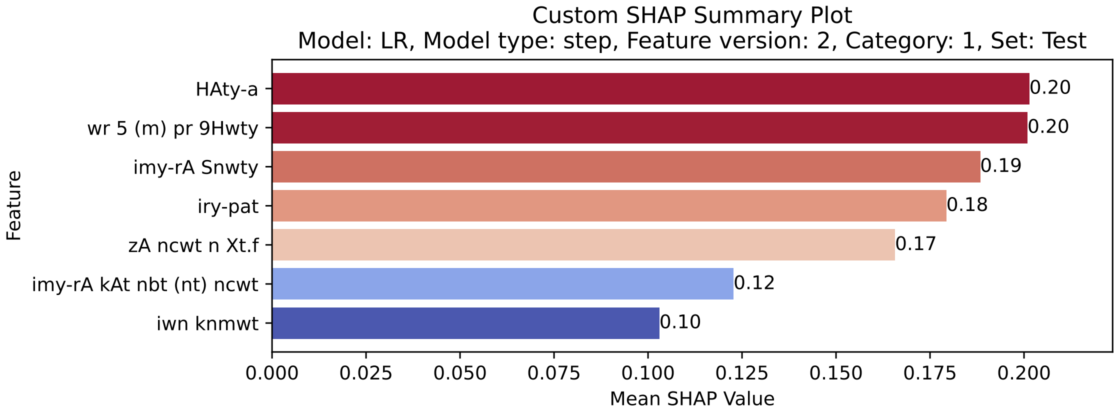
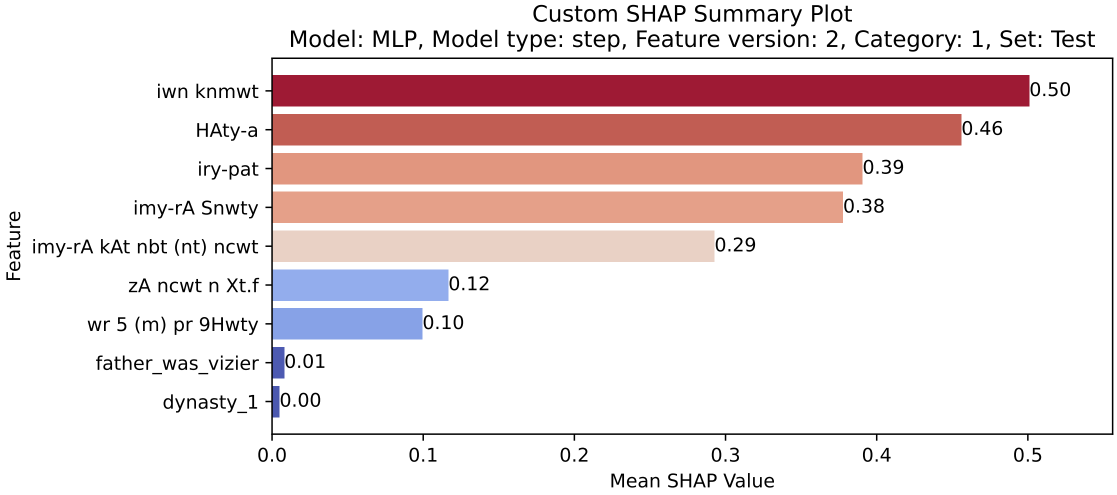


Figure 12: Custom plot of SHAP mean values for **Ridge MLP** model using **feature version 1**. This plot was defined in Section 2.5. The SHAP means were computed on test set using only observations where the given feature was present, see Figure 18.

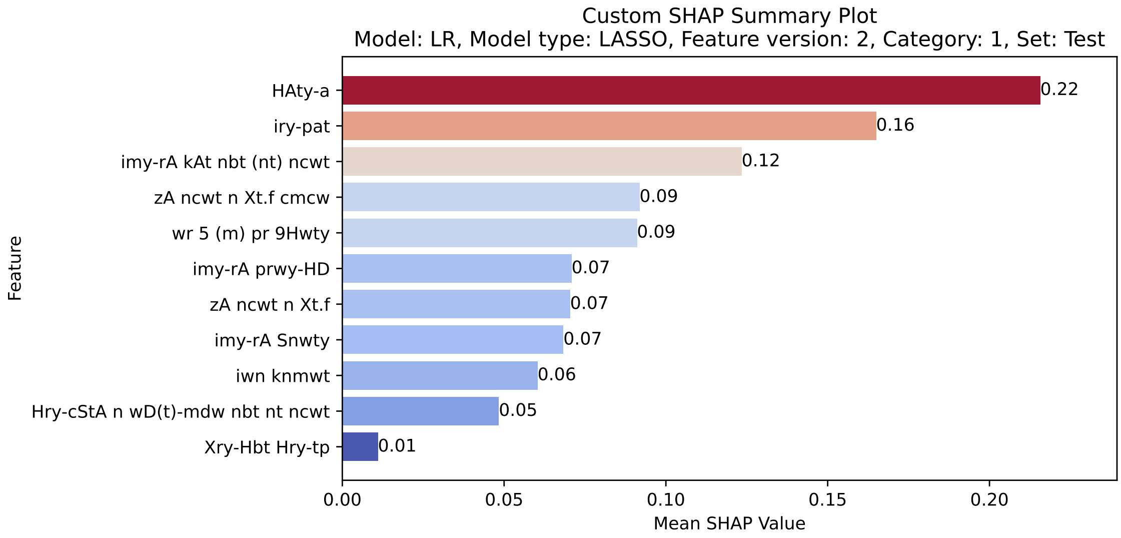


(a) Step LR model, feature version 2

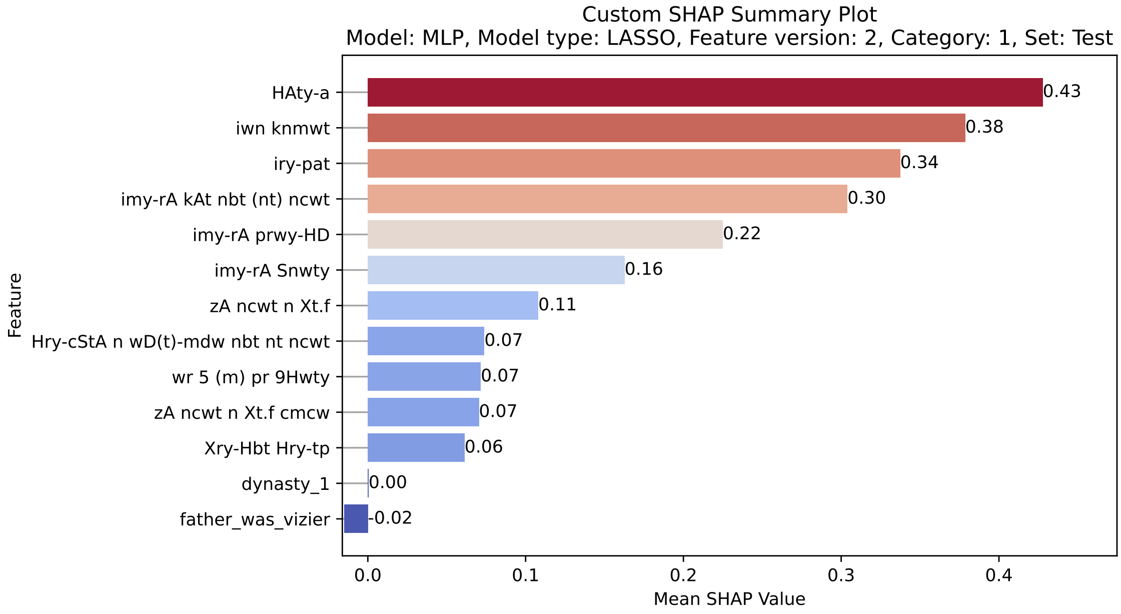


(b) Step MLP model, feature version 2

Figure 13: Custom plot of SHAP mean values for **step LR and MLP** model using **feature version 2**. This plot was defined in Section 2.5. The SHAP means were computed on test set using only observations where the given feature was present, see Figure 19a.



(a) LASSO LR model, feature version 2



(b) LASSO MLP model, feature version 2

Figure 14: Custom plot of SHAP mean values for **LASSO LR and MLP** model using **feature version 2**. This plot was defined in Section 2.5. The SHAP means were computed on test set using only observations where the given feature was present, see Figure 19b.

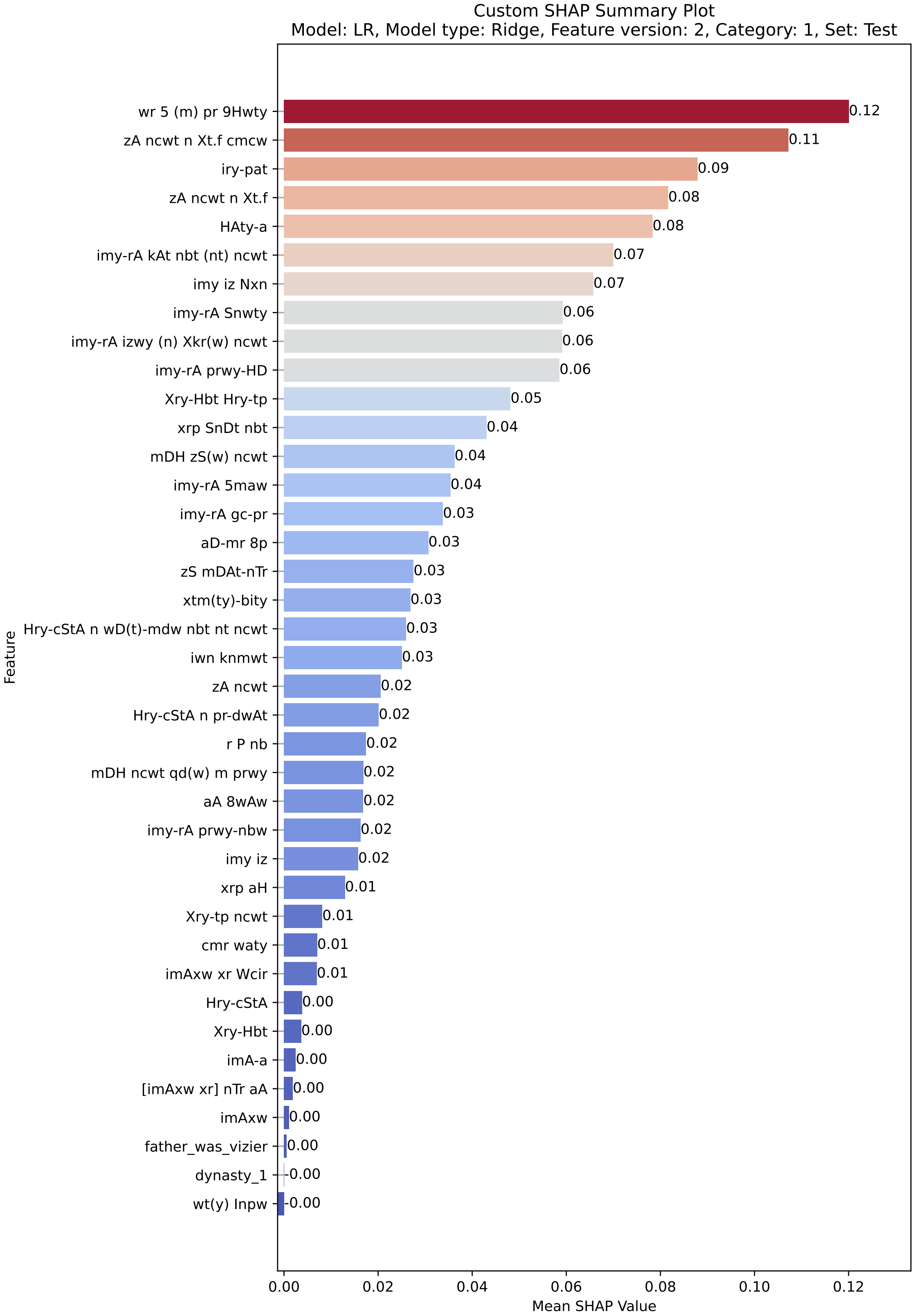


Figure 15: Custom plot of SHAP mean values for **Ridge LR** model using **feature version 2**. This plot was defined in Section 2.5. The SHAP means were computed using only observations where the given feature was present.

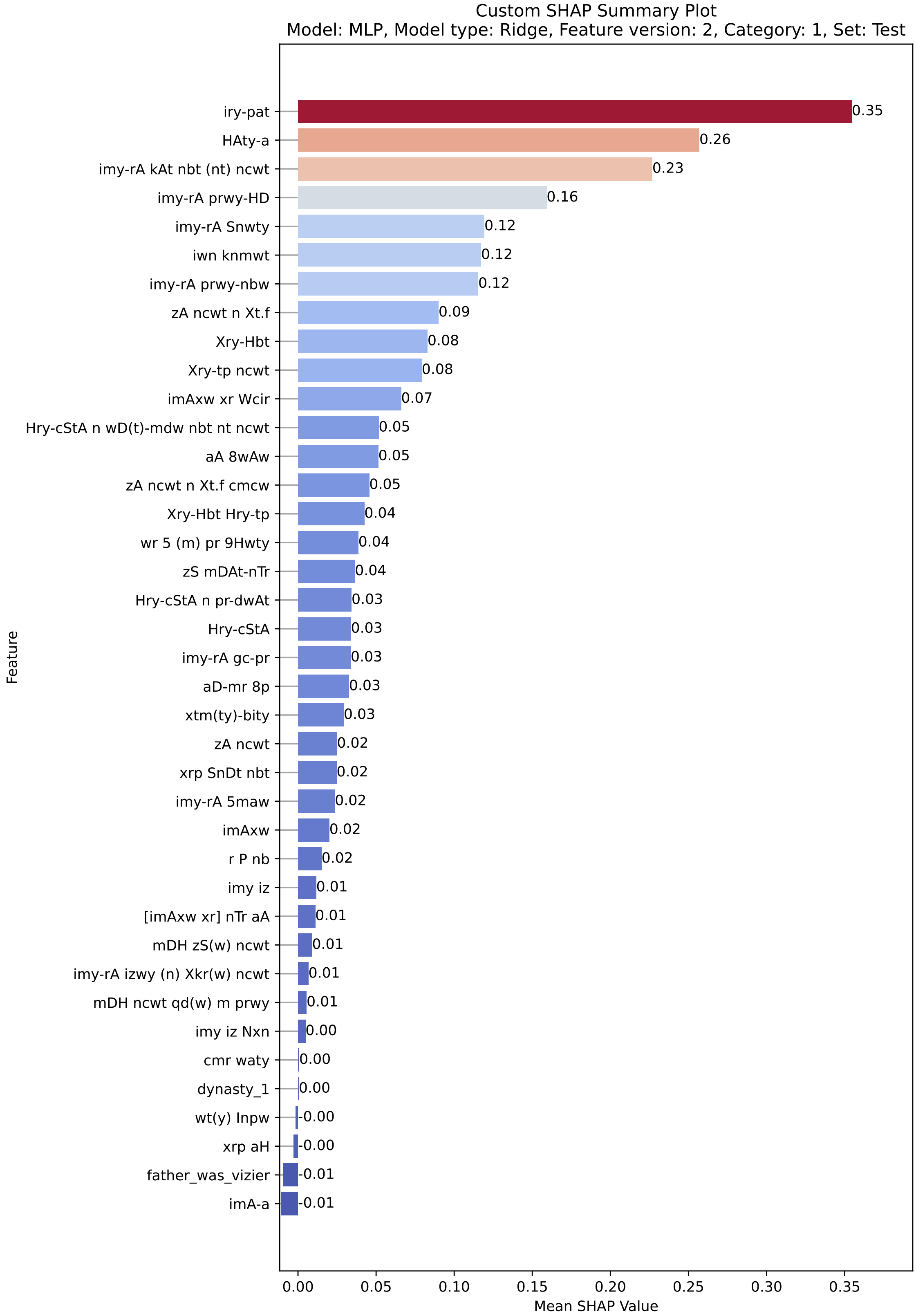
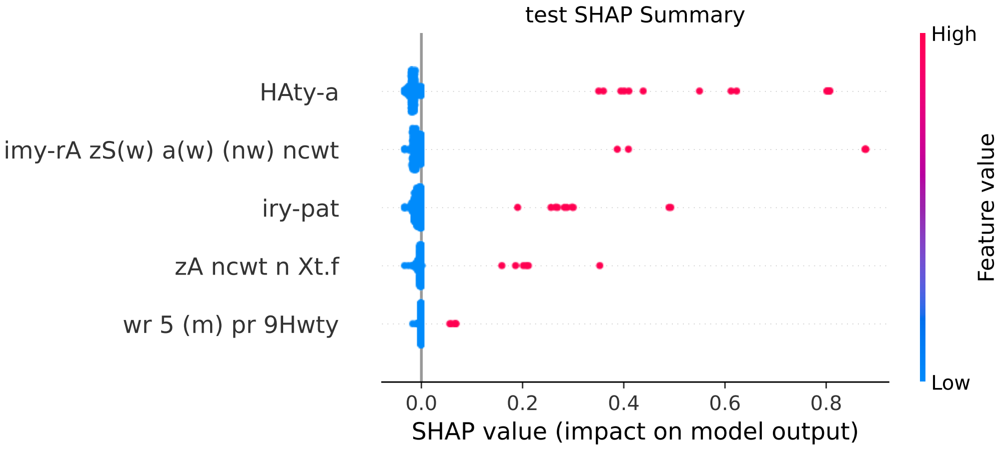
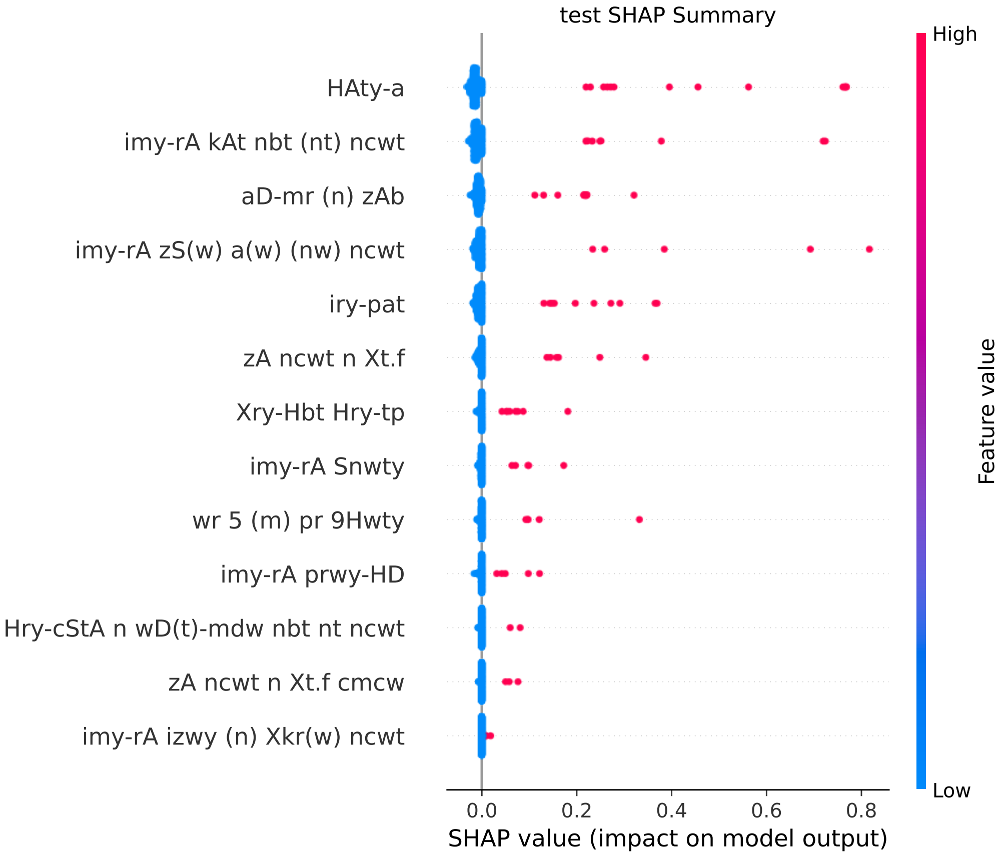


Figure 16: Custom plot of SHAP mean values for **Ridge MLP** model using **feature version 2**. This plot was defined in Section 2.5. The SHAP means were computed using only observations where the given feature was present, see Figure 20.



(a) Step MLP model, feature version 1



(b) LASSO MLP model, feature version 1

Figure 17: SHAP summary plot for **step and LASSO MLP** model using **feature version 1**. The SHAP values were computed on the test set.

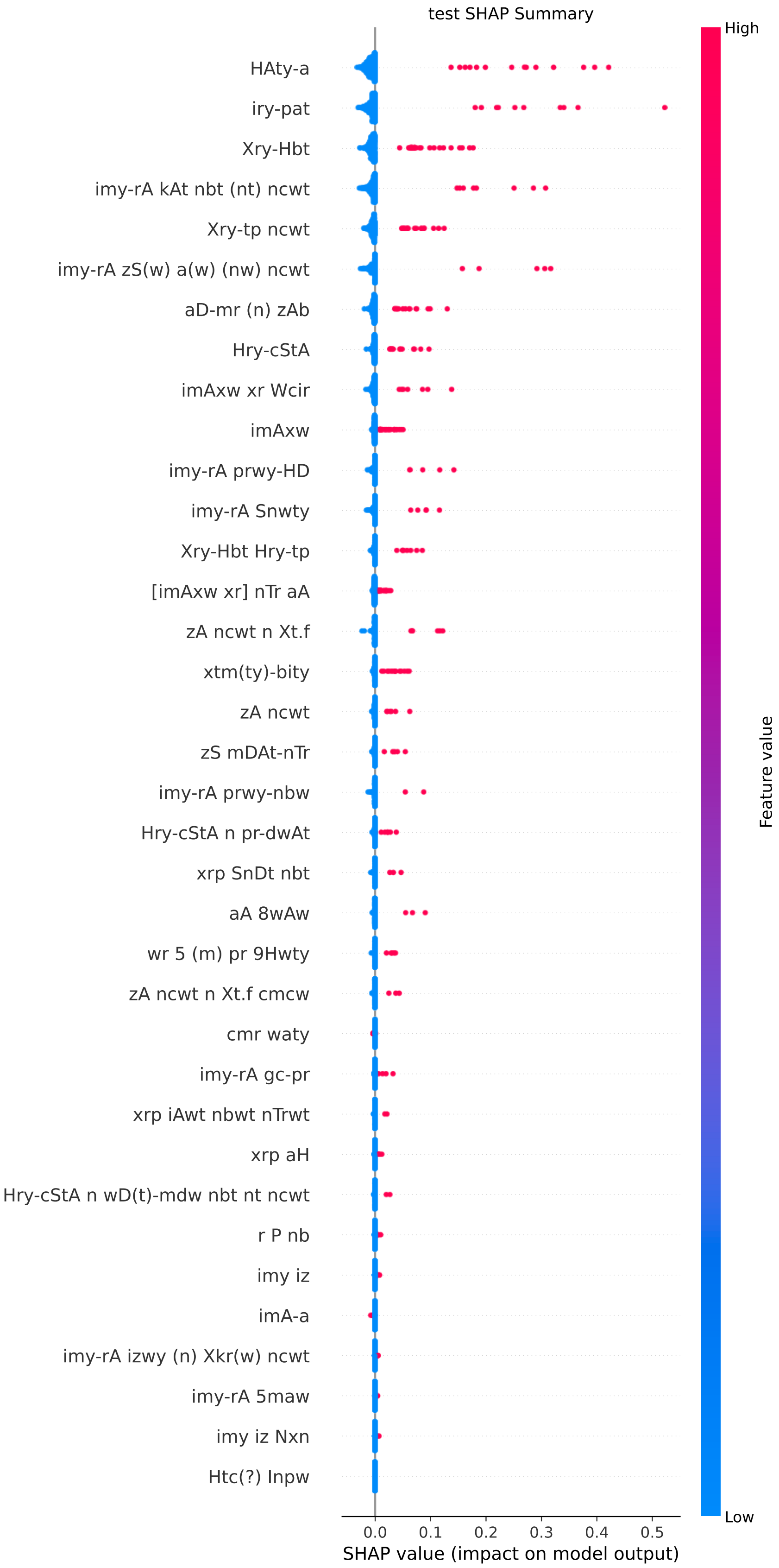
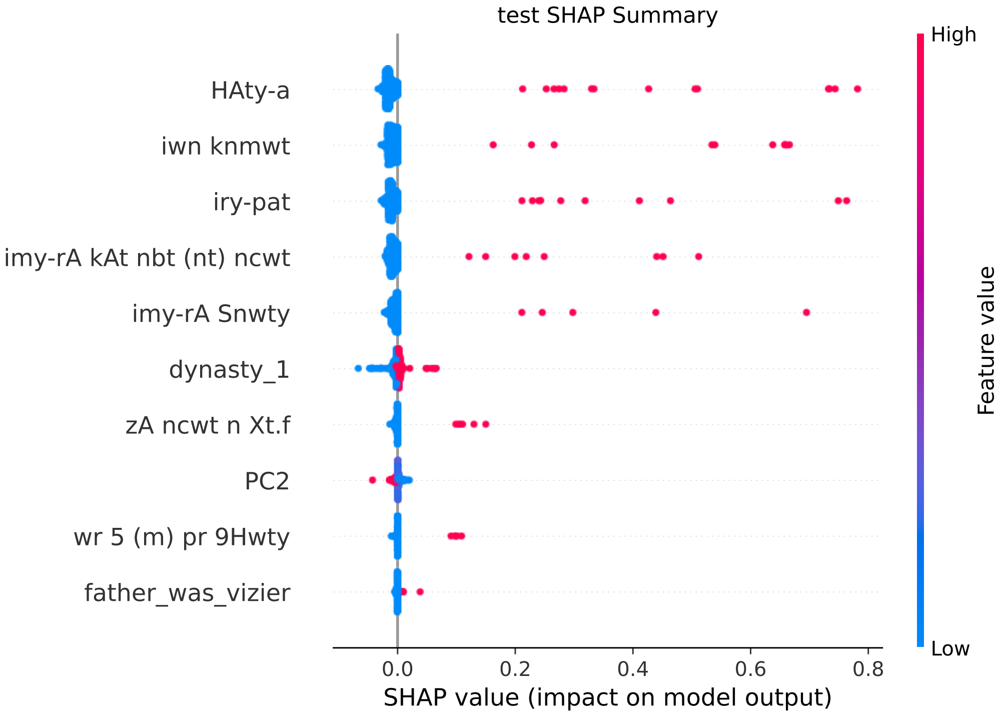
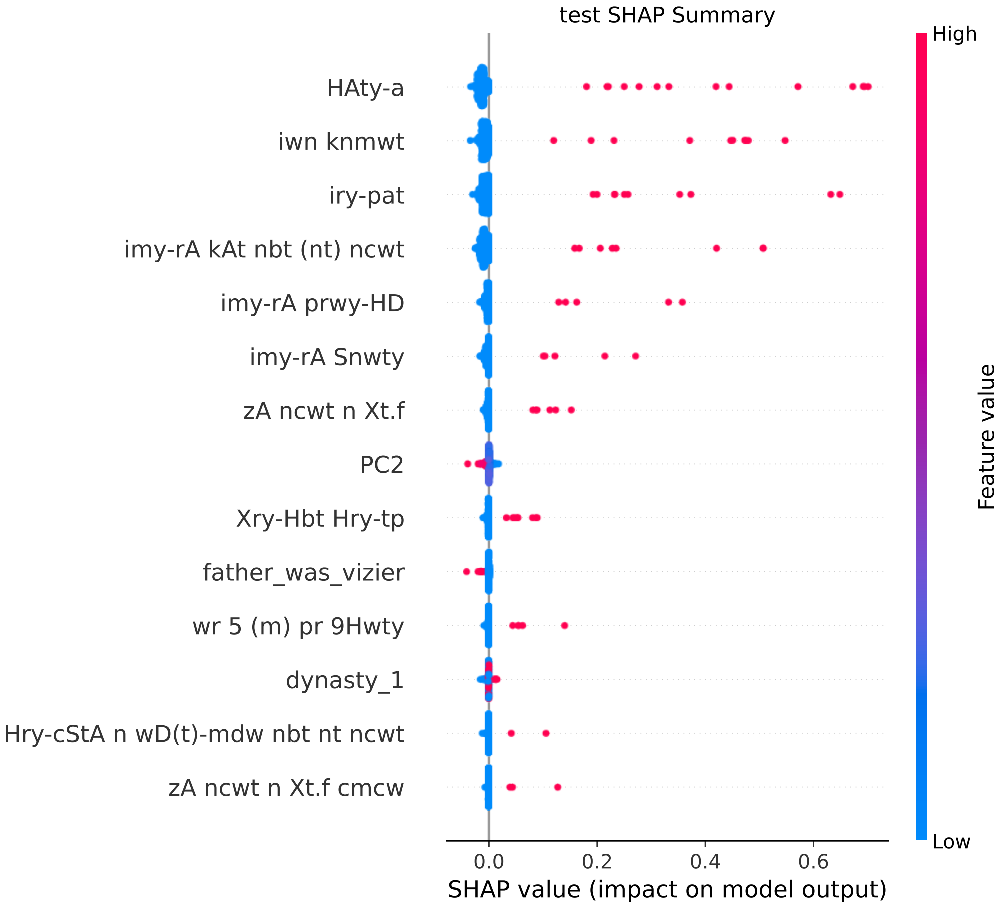


Figure 18: SHAP summary plot for **Ridge MLP** model using **feature version 1**. The SHAP values were computed on the test set.



(a) Step MLP model, feature version 2



(b) LASSO MLP model, feature version 2

Figure 19: SHAP summary plot for **step and LASSO MLP** model using **feature version 2**. The SHAP values were computed on the test set.

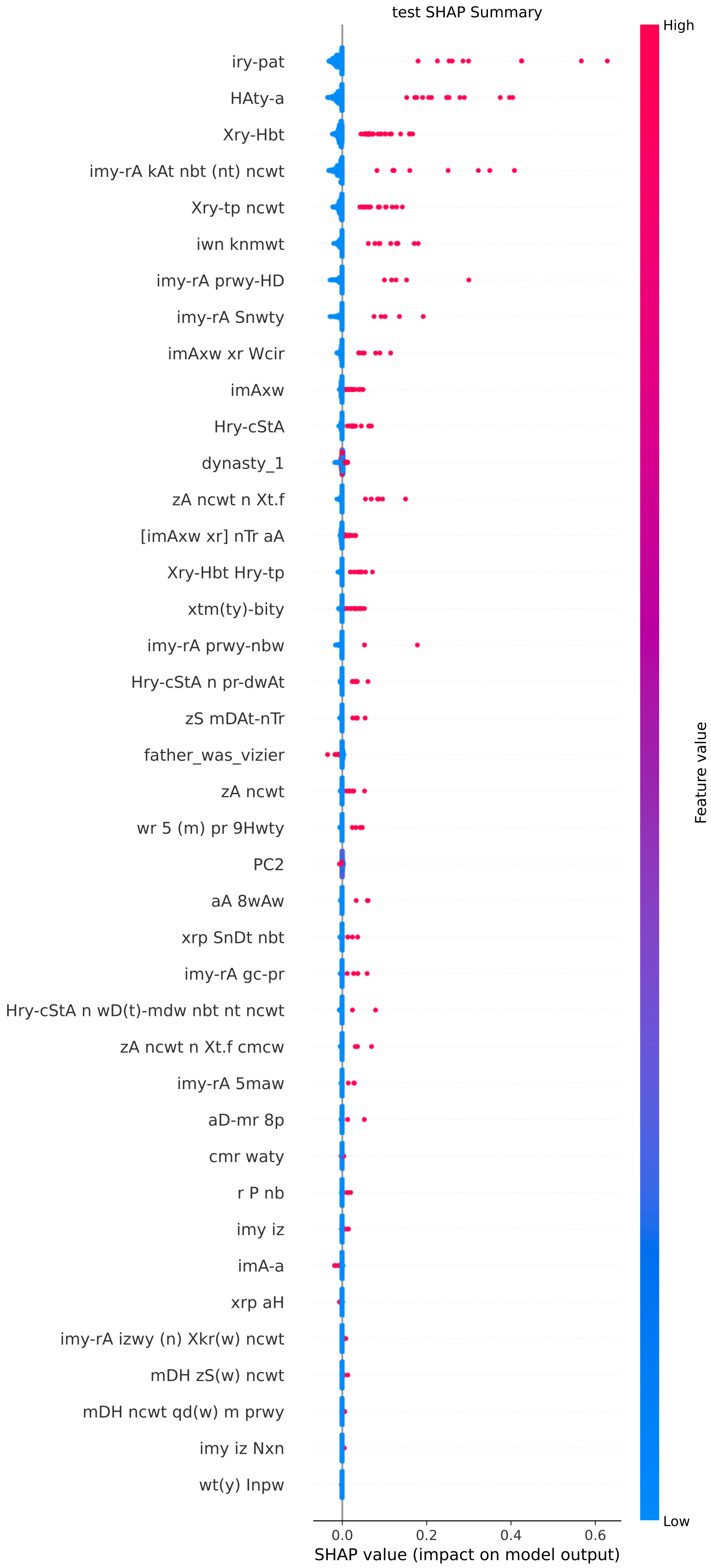


Figure 20: SHAP summary plot for **Ridge MLP** model using **feature version 2**. The SHAP values were computed on the test set.