UNIVERSIDAD POLITÉCNICA DE MADRID

ESCUELA TÉCNICA SUPERIOR DE INGENIEROS DE TELECOMUNICACIÓN

ETSIT-UPM

PREDICTIVE AND DESCRIPTIVE LEARNING

MACHINE LEARNING REPORT

Vic Bracke

2024-2025

December 31, 2024

Contents

List of Figures

List of Tables

1 Introduction

Machine learning (ML) is a powerful tool for uncovering patterns, making predictions, and classifying data. In this report, an extensive data preprocessing phase is carried out to prepare the dataset for analysis. Several ML models are then introduced and implemented for regression, classification, and clustering tasks. These models are trained using a nested cross-validation procedure, ensuring robust model evaluation and optimized hyperparameter tuning. The results of these evaluations are presented, followed by a comprehensive discussion of the findings. Finally, a conclusion is drawn based on the insights gained throughout the report.

The primary dataset used for developing the machine learning models in this report is derived from the OSA case study. However, to further demonstrate the ability to perform a comprehensive dataset description and exploratory data analysis (EDA), a secondary dataset related to breast cancer recurrence prediction is also included in the appendix. Unlike the OSA dataset, no machine learning models will be trained on the breast cancer dataset.

The decision not to use the breast cancer dataset for model training is due to its data features, which are not suitable for addressing all key types of Machine Learning tasks—regression, classification, and clustering. This distinction highlights the complementary roles of the two datasets in showcasing the analytical techniques and methodologies applied in this report.

2 Developing Machine Learning models

2.1 Problem formulation

Obstructive sleep apnea (OSA) is the most common form of sleep-disordered breathing and is thought to affect almost 1 billion people worldwide. [1] It is characterized by repetitive interruptions in breathing during sleep due to partial or complete collapse of the upper airway, leading to fragmented and non-restorative sleep. [2] With a global prevalence of almost 1 billion people and 425 million adults experiencing moderate to severe cases, OSA poses a significant public health challenge. Despite its prevalence, around 80% of cases remain undiagnosed, highlighting the need for improved diagnostic approaches. [4]

A key tool in diagnosing and classifying OSA is the apnea-hypopnea index (AHI), which quantifies the average number of apneas (complete airflow cessation) and hypopneas (partial airflow reduction) per hour of sleep. According to the American Academy of Sleep Medicine, OSA severity is categorized as mild (5–15 events/hour), moderate (15–30 events/hour), or severe (30 events/hour). [2, 3] While AHI provides a standardized framework for diagnosis, it has limitations in capturing the broader impact of OSA. Factors such as the duration of apnea or hypopnea events and their physiological consequences are not accounted for, making it difficult to fully assess disease severity or predict treatment outcomes.

The hallmark symptoms of OSA include loud, disruptive snoring, witnessed apneas during sleep, and excessive daytime sleepiness. [2] Daytime drowsiness can impair concentration and increase the risk of accidents, particularly motor vehicle crashes, by 2 to 10 times. [5]

OSA arises from a complex interplay of anatomical and non-anatomical factors. Anatomical contributors include large neck circumference, soft tissue abnormalities, and craniofacial structure, which promote pharyngeal narrowing during sleep. Non-anatomical risk factors encompass obesity, central fat distribution, advancing age, male sex, and supine sleeping position [2]. Epidemiological studies underscore the heightened prevalence of OSA among men and individuals with obesity, with odds ratios of 4.1 and 10.5, respectively. [6] This is also confirmed by the systematic review led by Senaratna et al. (2016), where the overall body of evidence demonstrated that advancing age, male sex, and higher body-mass index increase OSA prevalence. [7] Additionally, it was shown in a study by Bixler et al.

(1998) that the prevalence of sleep apnea tends to increase with age, although the clinical significance (severity) of apnea tends to decrease. [8]

Further insights into OSA risk factors highlight a range of clinical and anthropometric predictors. Male sex is consistently associated with a higher prevalence of OSA, as demonstrated in multiple studies. The relationship between age and OSA, however, remains inconsistent: some studies report a significant increase in prevalence with advancing age, while others show conflicting results. Waist-to-hip ratio is another important anthropometric factor, with larger ratios linked to increased OSA risk. Neck circumference has also been correlated with OSA risk, though thresholds of 43 cm in men and 47 cm in women did not show consistent significance. [9] Furthermore, another study led by Hoffstein et al. (1993) identified age, sex, BMI, bed partner observation of apnea, and pharyngeal examination as significant predictors, explaining 36% of variability in apnea-hypopnea index (AHI) outcomes. [10] Together, these findings underscore the multifactorial nature of OSA risk, with both anatomical and non-anatomical contributors playing critical roles.

OSA has far-reaching consequences on health, quality of life, and economic burden. The intermittent oxygen desaturation and arousals during sleep lead to fragmented, non-restorative sleep, resulting in daytime fatigue [2]. Beyond personal health, undiagnosed OSA significantly increases medical costs, with affected individuals coping with considerably higher medical costs than their age- and sex-matched controls. [11] These findings highlight the critical need for early diagnosis and management to mitigate its broad-reaching effects.

The gold standard for diagnosing OSA is nocturnal polysomnography (PSG), an overnight sleep study conducted in a hospital or sleep unit. However, accessibility to PSG is limited due to its labor-intensive nature and correlated high costs. [12] Alternative diagnostic tools include the STOP-BANG questionnaire, which evaluates key risk factors, including snoring, daytime tiredness, high BMI, age over 50, male sex, high blood pressure, and neck circumference over 40 cm. A score of 3 or more indicates a high risk of OSA. Additional tools, such as the Epworth Sleepiness Scale (ESS) and Fatigue Severity Scale (FSS), help assess daytime sleepiness and fatigue, common OSA symptoms. [2] Despite these tools, many cases remain undiagnosed, highlighting the need for accessible diagnostic methods.

Effective management of OSA involves a combination of lifestyle modifications and medical interventions. Continuous positive airway pressure (CPAP) therapy remains the cornerstone of treatment, preventing airway collapse during sleep. Other evidence-based treatments include the use of oral appliances or surgical interventions [2]. Addressing risk factors, such as obesity and supine sleeping position, can further enhance treatment outcomes.

2.2 Machine Learning approach

Despite advancements in diagnostic procedures and increased awareness of the health risks associated with obstructive sleep apnea (OSA), a high rate of missed diagnoses persists. [12] To address this, researchers have been exploring various predictive models to identify patients at risk of OSA more efficiently. For instance, Rowley et al. (2000) tested several clinical prediction models using factors such as snoring, gasping, witnessed apneas, BMI, age, sex, hypertension, and neck circumference. While these models showed potential, they lacked sufficient accuracy to reliably distinguish between patients with and without OSA, though they could help prioritize individuals for more comprehensive tests like split-night polysomnography. [13] Given these limitations, there is an ongoing need for improved prediction methods.

One promising approach involves the use of machine learning (ML) models, which can be trained on a broader range of features that are known to influence OSA. By leveraging large datasets, ML models have the potential to more accurately predict the apnea-hypopnea index (AHI) and help in early identification and management of OSA. This approach not only enhances diagnostic accuracy but also holds the promise of developing personalized treatment strategies for patients at risk of OSA.

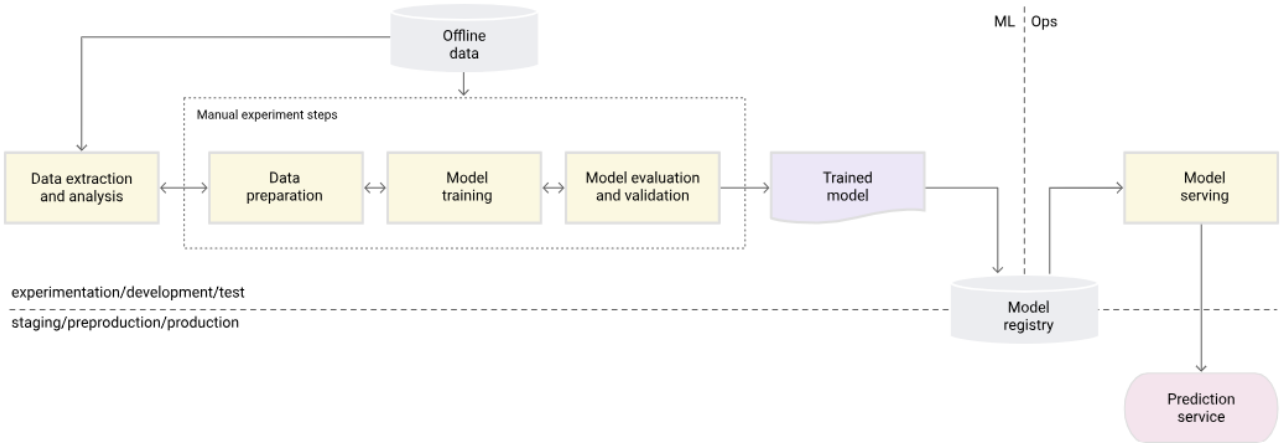

Figure 1: Machine Learning pipeline [14]

The development of a machine learning (ML) model for predicting the apnea-hypopnea index (AHI) follows a structured pipeline, as visualized in Figure 51. This process incorporates both foundational steps for ML development and principles for delivering a robust model to production. The steps include:

- **Problem Definition**: The success of an ML project starts with clearly defining the problem to align with business goals. In this case, the objective is to predict the apnea-hypopnea index (AHI), a key diagnostic measure for obstructive sleep apnea (OSA).
- **Data Collection**: Data relevant to the prediction task is collected from various sources such as databases or clinical records. This step is time-intensive and requires ensuring that the data is representative of the problem being solved.
- **Data Description**: Understanding the dataset is essential. This involves exploring the dataset schema and characteristics to determine the meaning of the features and target variable. Exploratory data analysis (EDA) employs visual tools like histograms and scatter plots to uncover patterns, relationships, and anomalies in the data, which informs subsequent feature engineering.
- **Data Preparation**: The data is cleaned and preprocessed for ML tasks. This step includes handling missing values, normalizing the data, splitting it into training, validation, and test sets, and applying transformations and feature engineering. Well-prepared data improves the model's ability to learn effectively.
- **Data Analysis**: Exploratory data analysis (EDA) employs visual tools like histograms and scatter plots to uncover patterns, relationships, and anomalies in the data, which informs subsequent feature engineering.
- **ML Model Selection**: Selecting the appropriate ML algorithm is critical. This involves evaluating different algorithms to balance accuracy and interpretability. The goal is to identify the most optimal model for the problem at hand.
- **Model Training and Hyperparameter Tuning**: The prepared data is used to train the selected ML models. During training, hyperparameter tuning is performed to optimize performance, ensuring the model learns the relationship between inputs and the target variable (AHI) effectively. This is done based on a holdout validation dataset.
- **Model Evaluation**: The trained model is evaluated using a holdout test set to assess its quality through performance metrics.
- **Model Deployment and Serving**: The model is deployed into a real-world environment to serve predictions. This deployment can take various forms, such as a REST API for online predictions, integration into edge or

mobile devices, or use in batch prediction systems. Effective deployment ensures scalable and efficient model use.

- **Model Monitoring and Maintenance**: Post-deployment, the model's predictive performance is continuously monitored to detect issues like data drift or reduced accuracy. This step may invoke retraining or iteration of the ML pipeline to maintain long-term effectiveness.

This pipeline follows the principles of MLOps, an ML engineering culture and practice that unifies ML system development (Dev) and ML system operation (Ops). It adapts DevOps principles to machine learning, enabling the reliable and efficient deployment and maintenance of ML systems in production. Like traditional software development, MLOps emphasizes continuous integration (CI) and continuous delivery (CD) but incorporates additional practices tailored to the unique characteristics of ML workflows. [14]

In MLOps, **continuous integration (CI)** involves not only testing and validating code but also ensuring the quality of data, data schemas, and ML models. This step is critical for maintaining consistency across the pipeline. **Continuous delivery (CD)** automates the deployment of entire ML training pipelines, streamlining the release of model prediction services and integrating them into broader system infrastructures. A unique feature of MLOps is **continuous training (CT)**, which focuses on automatically retraining models as new data becomes available. This ensures that models remain accurate and adaptable in dynamic environments.

By advocating for automation and monitoring across all stages—data integration, model training, testing, releasing, and deployment—MLOps transforms the challenge of building isolated ML models into the development of scalable and continuously operating ML systems. This unified approach bridges the gap between development and operation, making ML systems more reliable and efficient at scale.

3 Dataset description

3.1 Data access

The dataset for this project is derived from clinical data collected at Quirónsalud Málaga hospital. These data consist of detailed records of patients who underwent polysomnography (PSG) following clinical suspicion of obstructive sleep apnea (OSA). Relevant clinical details were extracted from these patients and compiled into an Excel file for analysis. The dataset is handled with strict adherence to data privacy and ethical guidelines to ensure patient confidentiality.

3.2 Initial data exploration

The dataset consists of 28 clinical parameters collected from 683 patients who underwent evaluation for obstructive sleep apnea (OSA). These parameters are organized as tabular data within an Excel file and are detailed below:

- **Patient**: A unique identifier for each patient, represented by a number.
- **Comentarios (Comments)**: Free-text field providing additional observations or notes about the patient.
- **Audios tumbado (Audio while lying down)**: Audio recordings captured while the patient is lying down, useful for snoring analysis.
- **Fotos (Photos)**: Facial images of the patient for craniofacial photographic analysis.
- **Audio fs KHz**: Sampling frequency of the audio recordings in kilohertz.
- **Gender**: Specifies whether the patient is male or female.
- **EPWORTH**: The Epworth Sleepiness Scale score, measuring the patient's level of daytime sleepiness.
- **IAH (Apnea-Hypopnea Index)**: The overall AHI value, indicating the severity of sleep apnea based on polysomnography.
- **IAH Supino**: AHI recorded specifically when the patient is lying on their back (supine position).
- **IAH Lateral**: AHI recorded when the patient is lying on their side (lateral position).
- **Peso (Weight)**: The weight of the patient in kilograms.
- **Talla (Height)**: The height of the patient in centimeters.
- **IMC (BMI)**: The Body Mass Index, calculated using weight and height, providing insight into obesity.
- **Edad (Age)**: The age of the patient in years.
- **PerCervical (Cervical perimeter)**: The circumference of the patient's neck, measured in centimeters.
- **Fumador (Smoker)**: Indicates whether the patient smokes (Yes/No).
- **Roncador (Snorer)**: Indicates whether the patient reports snoring during sleep (Yes/No).
- **Enfermedades (Illnesses)**: Information about any pre-existing conditions or comorbidities reported by the patient.
- **Sala/Ruidos (Noise in room)**: Describes the noise levels in the patient's sleeping environment.
- **Imagen (Image)**: Refers to any images linked to the patient's clinical data.
- **Dialecto (Dialect)**: Describes the patient's dialect or linguistic characteristics, which may influence audio data.
- **DIST EXT OJOS (Distance between outer edges of eyes)**: A facial measurement in centimeters.

- **DIST BARB-LOB (Distance between chin and earlobe)**: A facial measurement in centimeters.
- **Cansancio (Tiredness)**: Indicates if the patient reports feeling tired.
- **Concentrarse (Concentration)**: Reports whether the patient experiences difficulties concentrating.
- **PerdRespNoche (Breathing pauses at night)**: Indicates whether the patient reports episodes of breathing cessation during sleep.
- **HiperT (Hypertension)**: Indicates whether the patient has a history of high blood pressure.
- **EstHOSP (Hospital stay)**: Information about the patient's hospitalization status or history.

While the dataset provides a rich set of clinical features, it has several challenges that need to be addressed during the analysis:

- **Incomplete data**: Some entries have missing values, which can affect model training and predictions. For example, the varialbes DIST EXT OJOS, DIST BARB-LOB, Cansancio, Concentrarse, PerdRespNoche, HiperT and EstHOSP rarely contain any value.
- **Unstructured text**: The dataset includes unstructured textual information that is not directly usable for predictive modeling and requires additional preprocessing.
- **Non-relevant variables**: Not all features are directly relevant to predicting the AHI, and feature selection will be crucial for optimizing the model's performance and making the model reliable.

These issues highlight the need for careful data cleaning, exploratory data analysis, and feature engineering before proceeding to machine learning model development.

3.3 Data cleaning

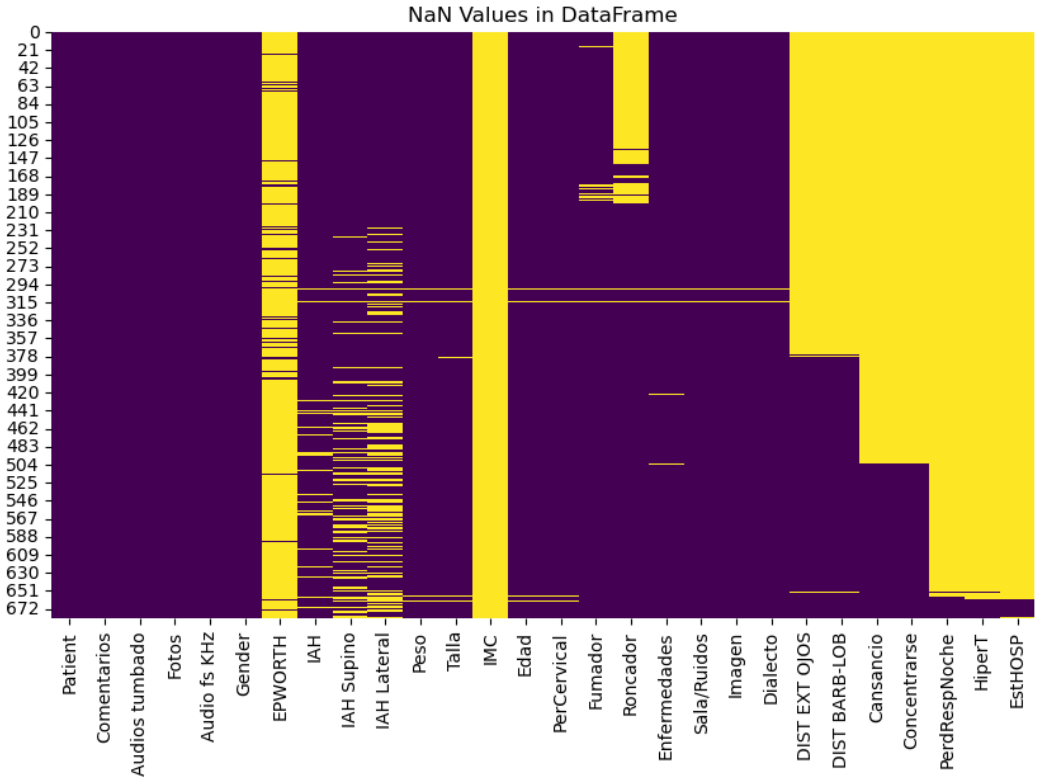

Figure 2: Visualization of NaN (Not a Number) values in the original database

Feature extraction analysis

To ensure effective preprocessing, feature selection was based on the availability of data in each column, visualized in figure 3 and the predictive relevance of the variables. The following clinical parameters were chosen for further analysis, after translating them into English.

- **Patient**: Identifier for each patient, with the following structure: P000X.
- **Gender**: Indicates whether the patient is male (hombre) or female (mujer). A mapping is done such that male corresponds to 1 and female corresponds to 0. This encoding is needed to be able to use this clinical variable as feature for model training.
- **IAH**: Apnea-Hypopnea Index, the primary target variable for prediction.
- **Weight**: Patient's weight, in kilograms.
- **Age**: Patient's age, in years.
- **Height**: Patient's height, in centimeters.
- **Cervical**: Patient's neck circumference, in centimeters.
- **BMI**: Body Mass Index of the patient, calculated by dividing a the patient's weight (in kg) by his or her height (in m) squared. [15]

While the EPWORTH feature, which measures daytime sleepiness, could be valuable for future analysis, it has been excluded due to a high rate of missing data (628 out of 683 values are unavailable).

Data type analysis

The 'dtype' of the "Weight" column was initially set to "object" due to the presence of non-numeric text entries, such as "no se saben" and "no quiere saberlo". To ensure data consistency, rows containing these text values were excluded.

Similarly, the 'dtype' of the "Patient" and "Gender" columns was also "object". These columns were then converted to the "category" data type, which offers several advantages. One key benefit of using categorical data is improved memory efficiency. Categorical data is stored more efficiently than object data, as pandas stores the unique values and uses integer codes to represent them, reducing memory usage, particularly for columns with repeated values.

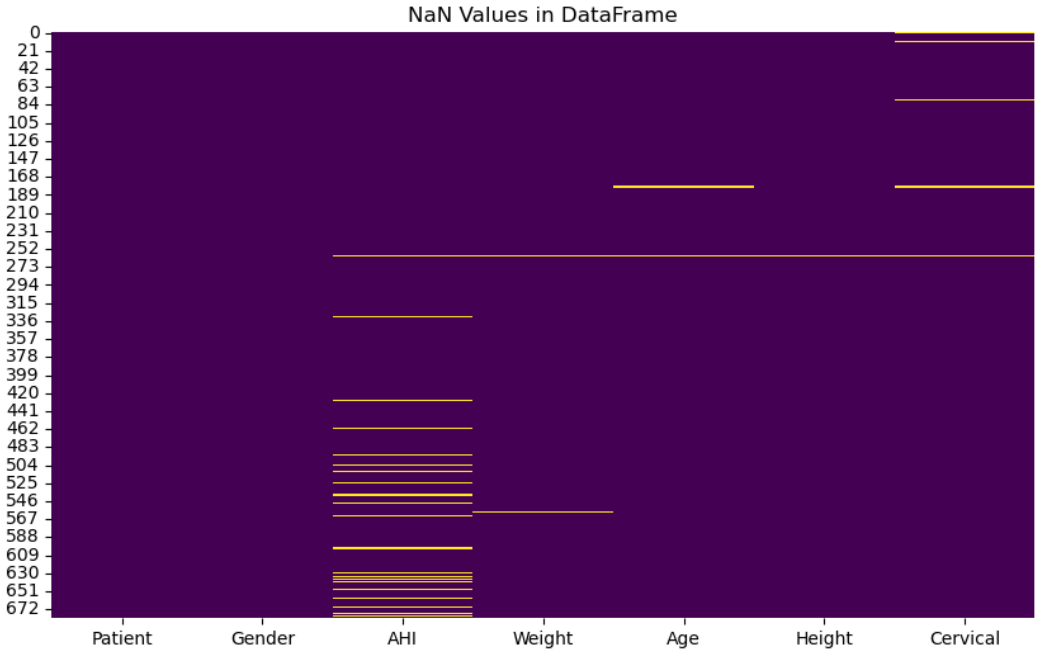

Figure 3: Visualization of NaN (Not a Number) values in the database after an initial feature extraction

NaN Value analysis

First, all '-1' values in the dataframe are converted to 'NaN', as these represent placeholders for missing or invalid data. Following this, the missing values in key independent variables, such as Weight, Age, Height, and Cervical, are replaced with the median of the respective column using a univariate imputer. The median is preferred over the mean for imputation in cases where the data distribution is skewed or contains outliers. [16] Since the exact distribution of these features is not yet clear, the median provides a more robust and reliable alternative for imputation at this stage.

Data imputation is an essential statistical technique used to retain most of the dataset's information by substituting missing values with alternative values. This is particularly useful as removing rows with missing data can lead to a significant reduction in dataset size, potentially introducing bias and impairing analysis. Various imputation techniques exist, including using the next or previous value, K-nearest neighbors, maximum or minimum value, model-based predictions, or simpler methods like the most frequent value, average, median, or linear interpolation. [17]

However, caution is required when imputing missing values in the target (dependent) variable, as this can introduce bias and would falsely influence the model performance. In this dataset, while missing values in independent variables were addressed through imputation, the 34 missing values in the target variable were directly removed to avoid bias.

Finally, after cleaning and preprocessing, the refined dataframe was saved to an Excel file for subsequent analysis, ensuring it is well-prepared for the next stages of the ML pipeline.

4 Exploratory data analysis

This section focuses on exploratory data analysis (EDA) of the cleaned dataset. The independent variables in this dataset are Gender, Weight, Age, Height, Cervical, and BMI, while the dependent variable is the AHI. Among these, Gender is the only categorical variable (stored as an integer - 1 for male and 0 for female), and the other columns contain continuous numerical values represented as floats. After the data cleaning and imputation steps outlined in the previous section, the dataset consists of a total of 649 patients.

	Gender	AHI	Weight	Age	Height	Cervical	BMI
0	1	71.0	82.000000	39.0	168.000000	40.617735	29.053288
1	1	29.6	119.000000	56.0	174.000000	48.000000	39.305060
2	1	56.0	87.797037	46.0	171.399408	43.000000	29.885548
3	1	19.7	78.000000	39.0	168.000000	42.000000	27.636054
4	1	9.0	80.000000	32.0	173.000000	40.000000	26.729927

Figure 4: Table with first five rows of the dataset

Figure 4 displays the first five rows of the dataset, providing an overview of its structure and content. Additionally, Figure 5 presents the basic statistical properties of the numerical columns.

	AHI	Weight	Age	Height	Cervical	BMI
count	648.000000	648.000000	648.000000	648.000000	648.000000	648.000000
mean	20.388673	87.686706	49.446788	171.323453	40.637438	29.834179
std	18.697199	18.245221	12.393654	9.501878	3.925383	5.584132
min	0.000000	45.000000	19.000000	144.000000	30.000000	18.289895
25%	6.300000	75.000000	40.000000	165.000000	38.000000	26.038459
50%	14.250000	86.000000	49.000000	171.000000	41.000000	28.733209
75%	30.000000	97.250000	59.000000	178.000000	43.000000	32.662431
max	108.600000	165.000000	88.000000	197.000000	53.000000	63.654952

Figure 5: Table with description of statistical properties of the numerical features

The dataset no longer contains any missing values due to the preprocessing and data imputation procedures. However, it does include one duplicate row. Specifically, two patients, identified by Patient IDs P0125 and P0201, share identical clinical variables. While such cases are rare, they are plausible in a clinical context. Therefore, the duplicate row will not be removed from the dataset, as it reflects realistic scenarios rather than errors.

4.1 EDA for regression

4.1.1 Data Distribution

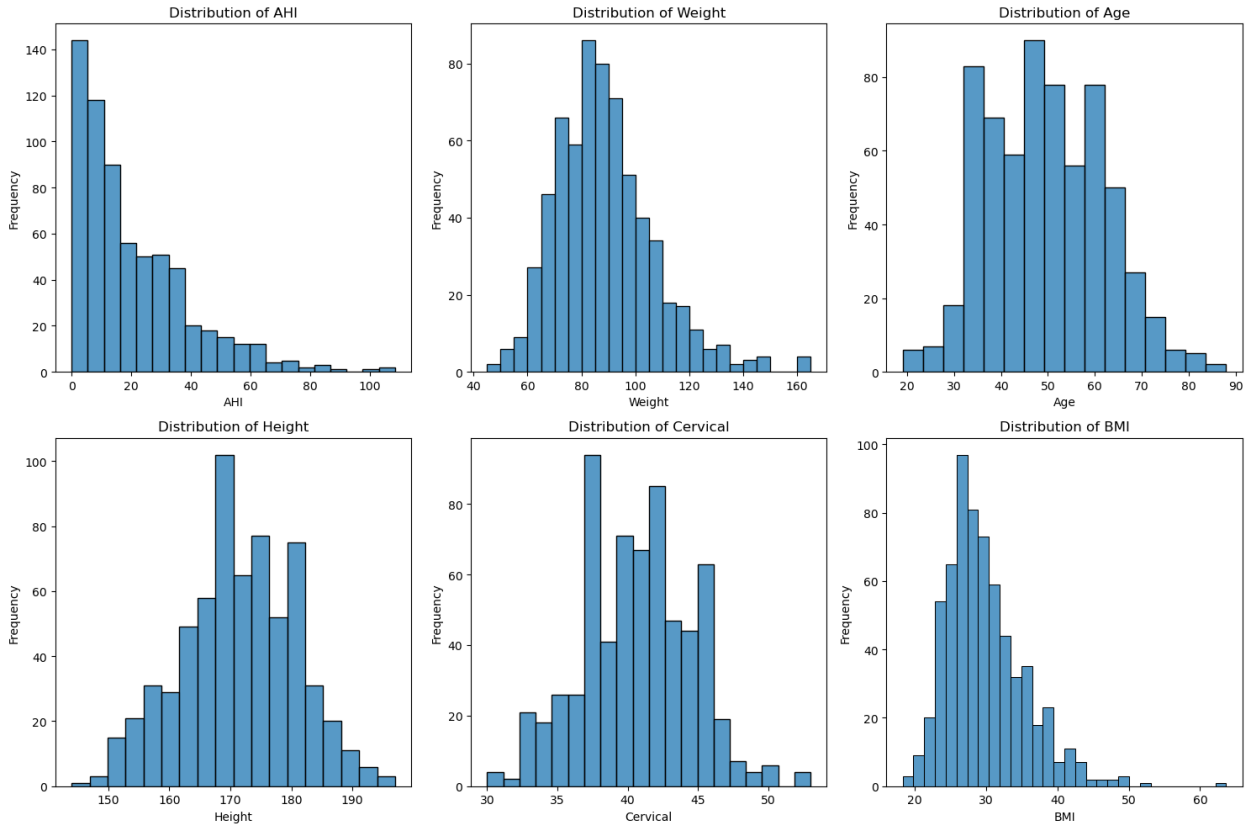

Figure 6: Plot with data distributions of the continuous features

Figure 55 displays the histograms of the numerical variables in the dataset. To determine whether these variables follow a normal distribution, the Shapiro-Wilk normality test is applied. The null hypothesis of this test states that "the data is normally distributed." If the p-value is less than 0.05, the null hypothesis is rejected, indicating that the data does not follow a Gaussian distribution. Based on the results (shown in Figure 7), none of the features in the dataset exhibit a normal distribution.

	Shapiro-Wilk Test Statistic	p-value
AHI	0.868548	6.643517e-23
Weight	0.960238	3.083062e-12
Age	0.989504	1.378645e-04
Height	0.993931	1.058192e-02
Cervical	0.990703	4.157560e-04
BMI	0.937381	7.312283e-16

Figure 7: Table with results of the Shapiro-Wilk normality test

To identify the distribution that best aligns with each feature, a Kolmogorov-Smirnov (KS) test is conducted. This test compares the empirical distribution function $F(x)$ of a sample to a specified theoretical distribution $G(x)$. The tested distributions include ['norm', 'expon', 'uniform', 'chi2', 'gamma']. The results of the KS test are as follows:

- **AHI**: Best fits an exponential distribution with a p-value of 0.0824.
- **Weight**: Fits both a chi-square distribution ($p = 0.2965$) and a gamma distribution ($p = 0.2964$).

- **Age**: Fits a chi-square distribution ($p = 0.0708$) and a gamma distribution ($p = 0.0588$).
- **Height**: Does not fit any of the tested distributions.
- **Cervical**: Does not fit any of the tested distributions.
- **BMI**: Best fits a gamma distribution with a p-value of 0.3654.

These findings highlight the variability in the distributions of different features, emphasizing the need for careful consideration during preprocessing and model development. For example, transformations like logartihms may be necessary for features such as Weight and AHI to address skewness or non-Gaussian characteristics.

4.1.2 Relationship continuous variables and target variable

To analyze the data for predicting AHI, the focus is put on exploring the relationships between the predictors (Age, BMI, and Cervical) and the outcome variable (AHI). The aim is to assess the predictive power of these features and evaluate their correlations to detect multicollinearity. While individual predictive power is important, even variables with low predictive strength can contribute to the model if they are uncorrelated with others.

The features Weight and Height were excluded as they are encapsulated within the BMI variable, making them redundant. Additionally, the Gender feature was temporarily removed since it is categorical.

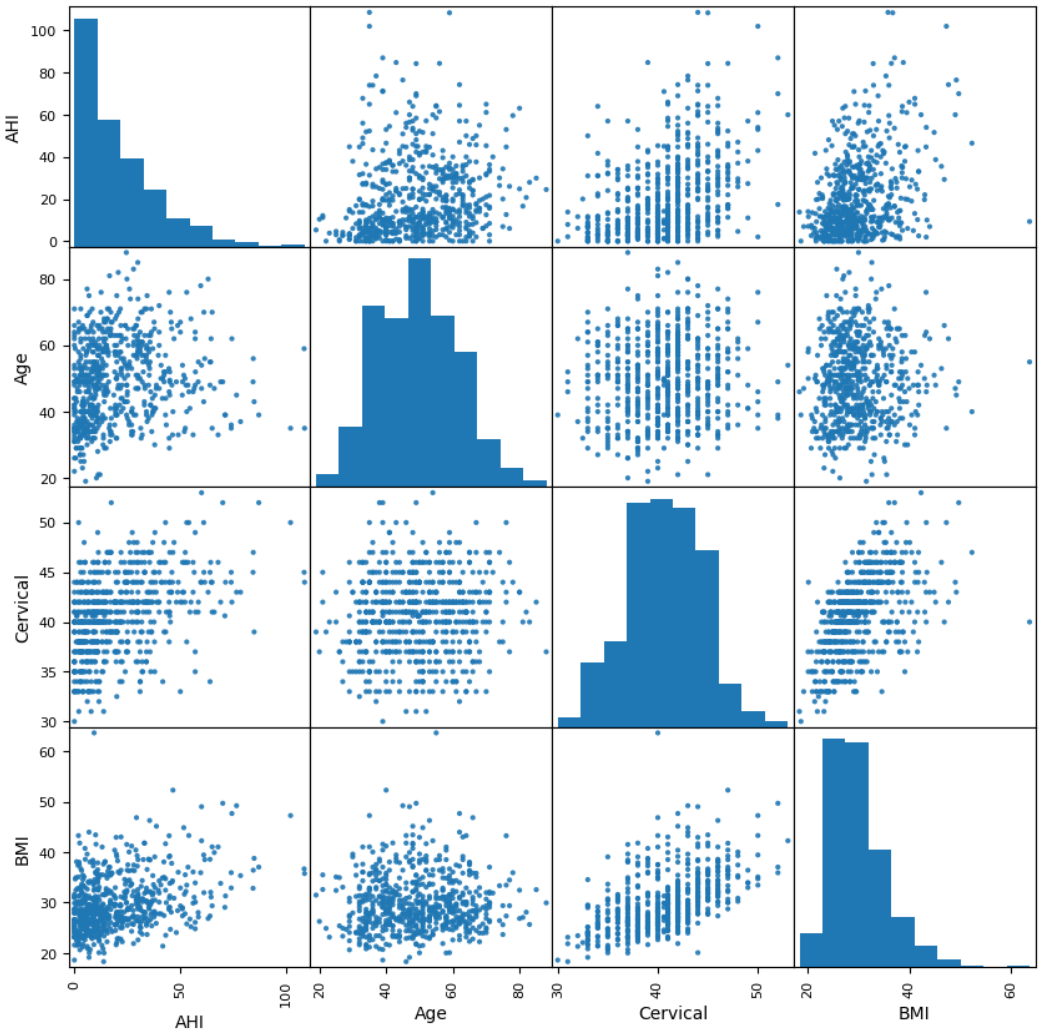

Figure 8: Scatter matrix of all continuous features

The scatter plot matrix, as shown in figure 8, visualizes all pairwise scatter plots between the variables in the dataset, arranged in the form of a matrix. [18] This tool provides valuable insights into the dataset by addressing the following questions:

- Are there pairwise relationships between the variables? If so, what is the nature of these relationships (e.g., linear, non-linear)?
- Are there any outliers present in the dataset that deviate significantly from the patterns observed?
- Is there evidence of clustering or grouping within the data based on certain variables?

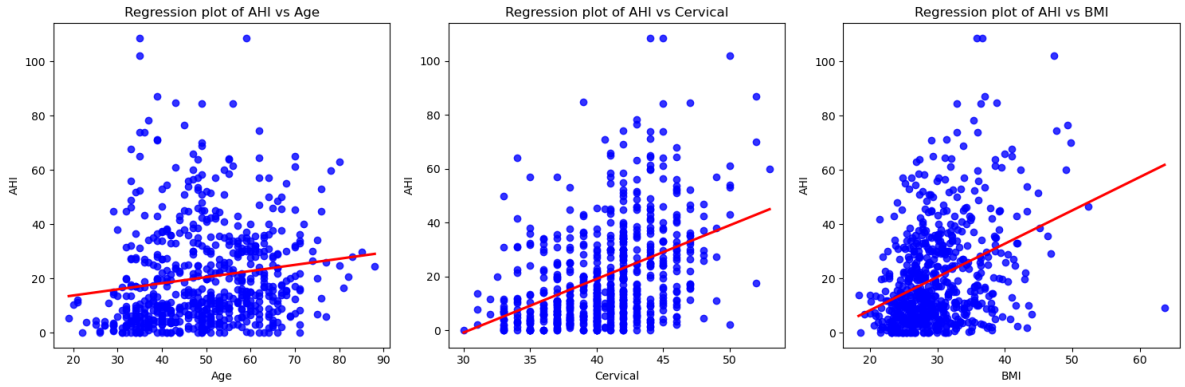

Figure 9: Regression plots of the predictors and the outcome variable

To have even more insight into the pairwise relationships that are present between the continuous features, regression plots can be made, visualized on figure 9, on which the regression line is visualized in red. Regression plots visually highlight linear trends between continuous features. However, they do not quantify relationship strength. For this, a correlation matrix is more suitable, providing exact values for the strength and direction of relationships. The choice between regression and correlation depends on the goal: prediction and trend visualization versus assessing relationship strength. [19]

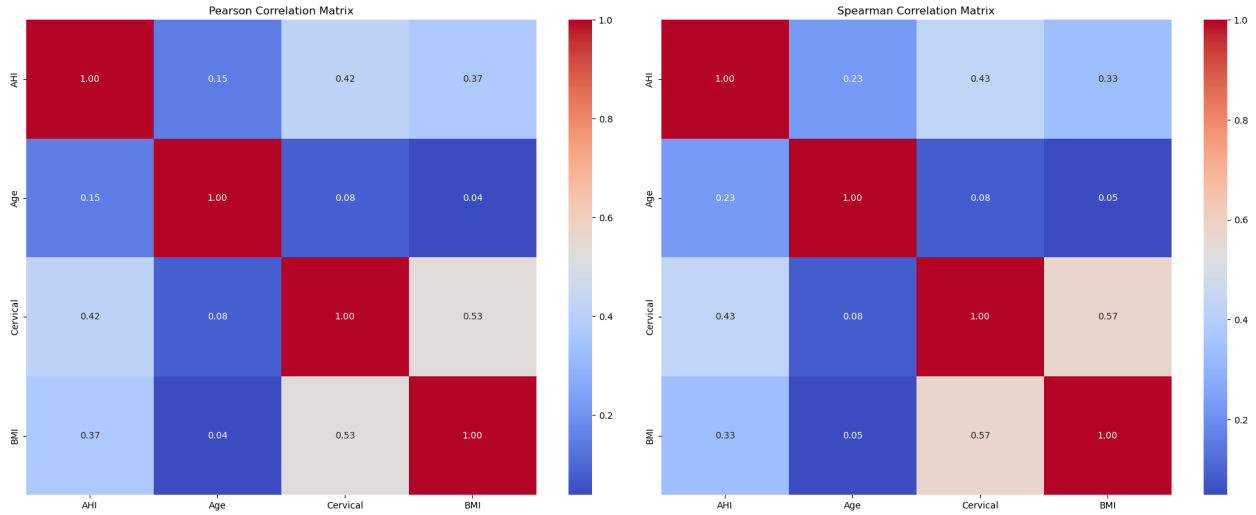

Figure 10: Pearson (left) and Spearman (right) correlation matrices

To accurately determine the strength of linear or non-linear relationships, correlation matrices are essential. Two commonly used methods are Pearson and Spearman correlations, which results are present in figure 10.

The Pearson correlation measures the linear association between two normally distributed continuous variables, while the Spearman rank correlation assesses monotonic relationships between variables. Spearman correlation is particularly useful for (1) non-normally distributed continuous data, (2) ordinal data, and (3) datasets with outliers, as it is relatively robust to their effects. As in this case, based on the results of the Shapiro-Wilk normality tests, the features do not follow a normal distribution, the Spearman correlation matrix is more reliable.
Like the Pearson coefficient, the Spearman coefficient ranges from -1 to $+1$, indicating no association ($\rho = 0$) to a perfect monotonic relationship ($\rho = -1$ or $\rho = +1$).

However, correlations are frequently misunderstood and misapplied. [19] It is essential to understand that an observed correlation (i.e., association) does not imply a causal relationship between two variables. Additionally, a common misconception is that a correlation coefficient near zero signifies no relationship between variables. In reality, correlation specifically measures linear or monotonic associations and is not suitable for capturing more complex or non-linear relationships.

4.1.3 Influence of gender on continuous variables

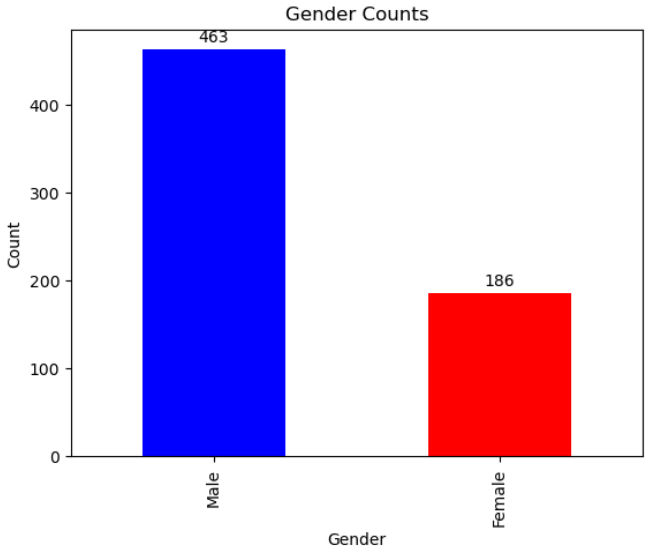

Figure 11: Barplot of the independent variable: Gender

For the categorical feature Gender, a bar plot is shown in Figure 11. Based on the figure, it is evident that more male patients underwent PSG (Polysomnography) than female patients.

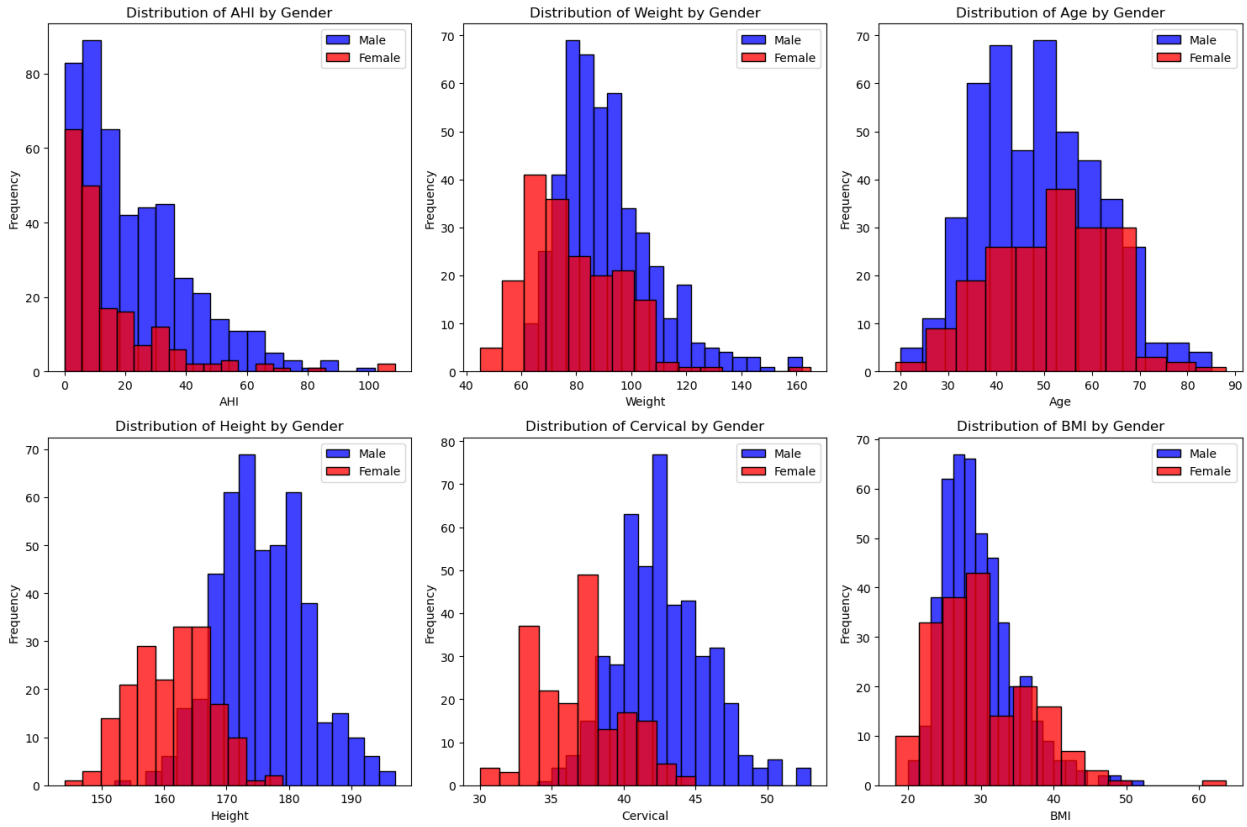

Figure 12: Plot with data distributions of the continuous features grouped by gender

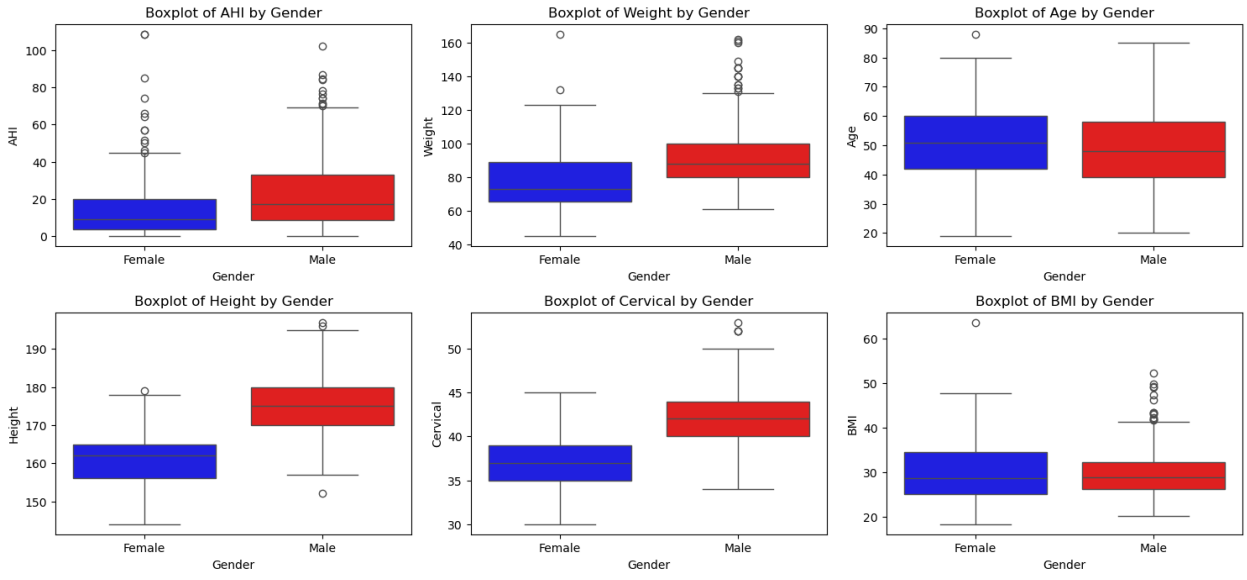

Figure 13: Boxplots of the continuous features grouped by gender

To investigate the influence of gender on the other clinical variables, separate histograms and boxplots are generated for each numerical feature, visualized in figure 12 and 13. However, to determine whether the gender-based differences in these distributions are statistically significant, appropriate statistical tests are required.

A common approach would be to use a t-test, but this test relies on two key assumptions:

- **Normality**: The data should follow a normal distribution. This has already been tested using the Shapiro-Wilk test, which revealed that none of the features follow a normal distribution.
- **Homogeneity of Variances**: The variance between groups (male and female) should be equal. This assumption is checked using Levene's test, where the null hypothesis states that the variances of the two groups are equal (homoscedasticity). A p-value below 0.05 would reject this null hypothesis.

	Levene Test Statistic	p-value
AHI	7.508407	0.006311
Weight	0.393216	0.530835
Age	0.769053	0.380836
Height	2.106412	0.147168
Cervical	0.146870	0.701671
BMI	17.176085	0.000039

Figure 14: Table with results of Levene's test of homoscedasticity

The results of the Levene's test are visualized in Figure 14. It shows that for Weight, Age, Height, and Cervical, the null hypothesis of equal variances holds true, meaning the variances are statistically equal. However, for AHI and BMI, the null hypothesis is rejected, indicating significant differences in variance between genders for these features.

Given the failure to meet the normality assumption and, for some features, the homogeneity of variance assumption, it is more appropriate to use a non-parametric test such as the Mann-Whitney U-test. This more robust test does not assume normality or equal variances, making it suitable for comparing the distributions of clinical variables between male and female patients.

	U-statistic	p-value
AHI	56764.5	1.544469e-10
Weight	62520.0	1.161370e-19
Age	37547.0	1.192891e-02
Height	80154.0	8.564080e-67
Cervical	76167.5	7.527134e-54
BMI	43399.5	8.408131e-01

Figure 15: Table with results of the Mann-Whitney U-test

An overview of the Mann-Whitney U-test results is presented in Figure 15. The results demonstrate that, for all features except BMI, the differences between genders are statistically significant with a p-value < 0.05. This finding highlights the need to account for gender-specific variations when analyzing clinical variables.

4.2 EDA for classification

4.2.1 Data distribution

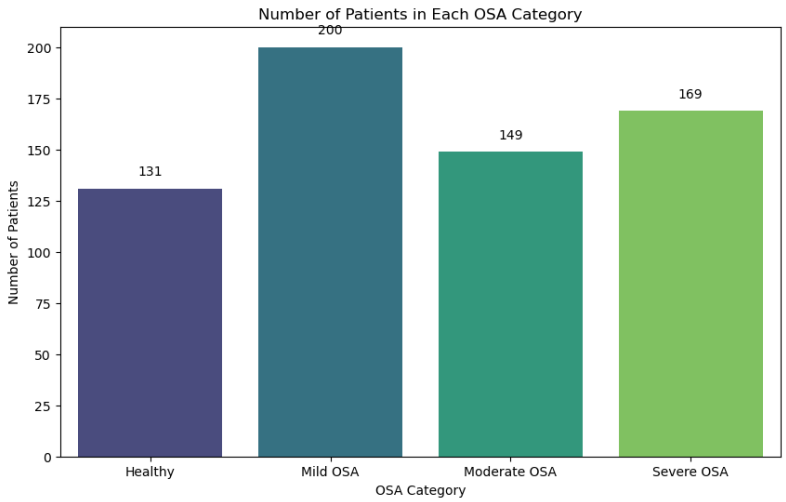

Figure 16: Barplot with severity levels of OSA

The target variable, AHI (Apnea-Hypopnea Index), can be categorized into four distinct groups to classify the severity of Obstructive Sleep Apnea (OSA) based on the number of apnea events per hour of sleep [2, 3]:

- **Healthy**: AHI $<$ 5 events/hour
- **Mild**: 5 $\leq$ AHI $<$ 15 events/hour
- **Moderate**: 15 $\leq$ AHI $<$ 30 events/hour
- **Severe**: AHI $\geq$ 30 events/hour

The distribution of these OSA severity levels is visualized in Figure 16, providing a clear overview of the dataset's composition in terms of the prevalence of each severity category. This categorization is crucial for further analysis and modeling, as it allows for distinguishing between varying levels of disease progression.

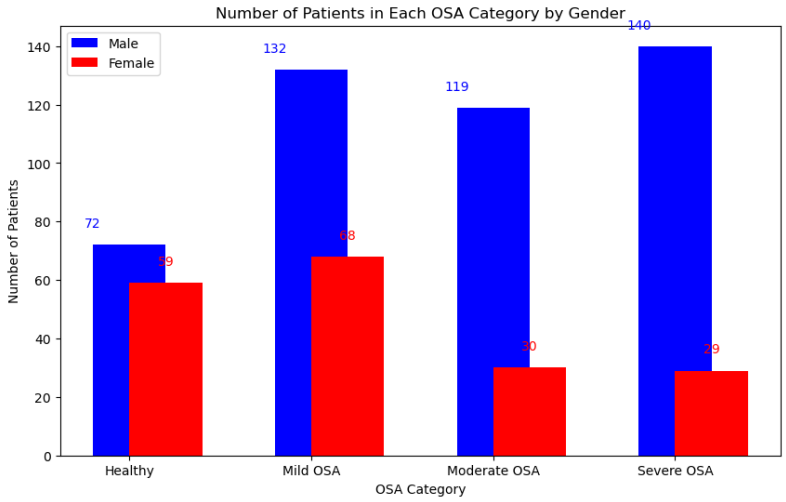

Figure 17: Barplot with severity levels of OSA grouped by gender

When grouping the OSA severity distributions by gender, visualized in figure 17, it is clear that there are significant differences in the different severity levels of OSA between men and women, where men seem to be more affected by the disease.

For further analysis, only two severity levels are considered in a newly defined outcome variable called OSA that has two possible values: healthy (AHI < 5) or severe (AHI ≥ 30). Data samples with AHI values between 5 and 30 are excluded from the analysis. In addition, the The resulting class distributions are visualized in figure 18.

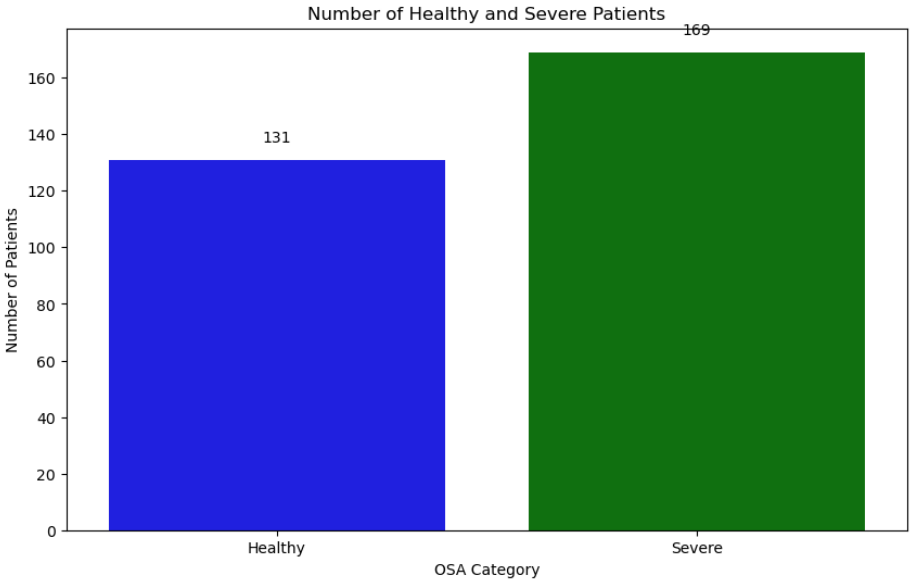

Figure 18: Barplot with class distributions of Healthy and Severe

Next, the distributions of the continuous variables (excluding the gender column) are plotted and grouped by the dichotomous outcome variable, OSA. These distributions are illustrated in figure 19.

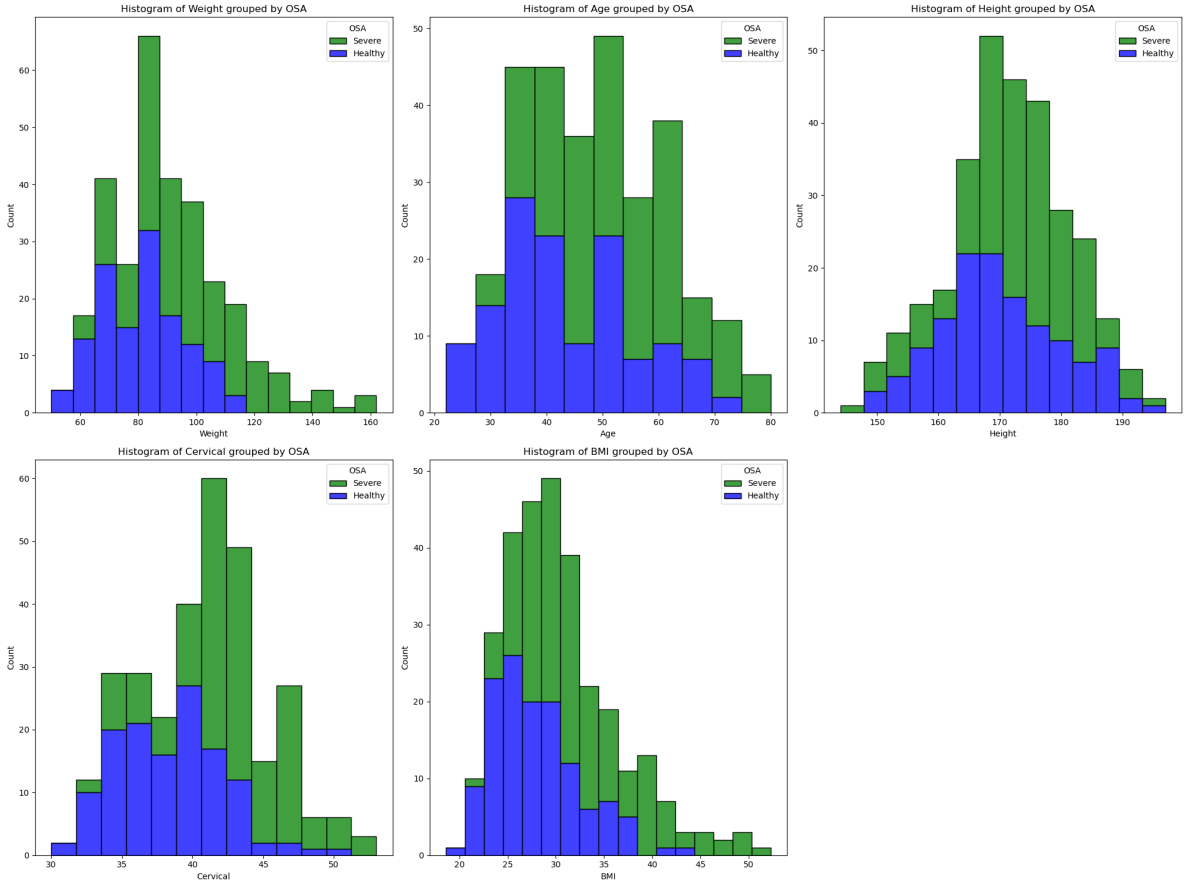

Figure 19: Plot with data distributions of the continuous features grouped by OSA

To assess whether significant differences exist in the continuous features (weight, age, height, cervical, and BMI) between healthy and severe OSA patients, a non-parametric Mann-Whitney U test was performed. The results, displayed in figure 20, indicate that all p-values are below the threshold of 0.05. Consequently, the null hypothesis is rejected, confirming that statistically significant differences exist between the two groups.

	Variable	U-statistic	p-value
0	Age	6829.0	6.282771e-09
1	Cervical	4763.0	1.082571e-17
2	BMI	6089.5	1.177343e-11
3	Weight	6110.0	1.401728e-11
4	Height	9116.0	4.362173e-03

Figure 20: Table with results of the Mann-Whitney U-test for difference between Healthy and Severe

4.2.2 Dealing with unbalanced dataset

As shown in figure 18, there is a significant disparity in the number of data samples between the healthy and severe groups. This phenomenon, referred to as class imbalance, arises when a dataset contains an unequal distribution of examples across different classes. [20] Such imbalances can have a profound impact on the performance of machine learning models. In particular, they can result in biased learning, where the model becomes disproportionately tuned to the majority class. Additionally, using accuracy or precision as a performance metric in such cases can be misleading, as it fails to appropriately evaluate the model's performance on the minority class. Therefore, it would be better to use the confusion matrix or the area under the ROC (Receiver Operating Curve) curve as metric to evaluate the performance. [21]

To address the issue of class imbalance, several techniques can be employed:

1. **Random Under-Sampling**: This technique balances the dataset by randomly removing samples from the majority class until the class distributions are equal. This reduces model complexity, runtime, and storage requirements. However, it risks losing valuable information and may lead to a reduced representation of the population, thereby impacting the model's generalization ability.

2. **Random Over-Sampling**: In this approach, instances of the minority class are increased by replicating existing samples. It prevents information loss and often outperforms under-sampling in practice. Nevertheless, it may increase the risk of overfitting. Care should be taken to apply over-sampling after dataset splitting to prevent uneven test and training sets.

3. **Synthetic Minority Over-sampling Technique (SMOTE)**: SMOTE generates synthetic samples for the minority class instead of duplicating them, illustrated in figure 21. This approach avoids overfitting linked to random over-sampling and maintains the original data distribution. Despite its advantages, SMOTE can introduce noise by increasing class overlap or generating irrelevant samples, and it is less effective for high-dimensional datasets. An improved variant, MSMOTE (Modified SMOTE), addresses these issues by selectively generating synthetic samples to minimize noise.

   SMOTE-related techniques from the imbalanced-learn package from scikit-learn [23] include:

   - `SMOTE`: Performs standard over-sampling using the SMOTE algorithm.
   - `SMOTENC`: Combines SMOTE with nominal and continuous features for mixed datasets.
   - `SMOTEN`: Tailored for nominal data by handling categorical features specifically.
   - `ADASYN`: Uses an Adaptive Synthetic (ADASYN) algorithm to focus on harder-to-classify minority samples.
   - `BorderlineSMOTE`: Enhances SMOTE by focusing on minority samples near the decision boundary for improved performance.
   - `KMeansSMOTE`: Incorporates KMeans clustering before over-sampling to generate more meaningful synthetic samples.
   - `SVMSMOTE`: Utilizes SVM-based boundary detection before applying SMOTE for better synthetic sample generation.

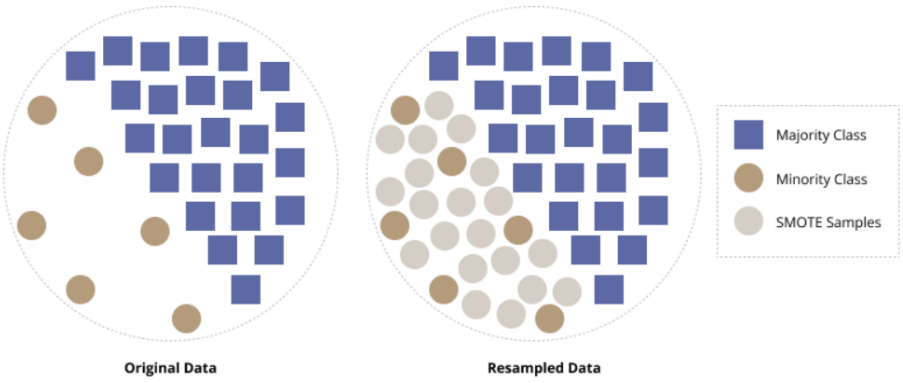

Figure 21: Graphical representation of the SMOTE algorithm

4. **Algorithm Selection**: Certain algorithms inherently handle imbalanced datasets better. Decision tree-based methods and ensemble techniques, such as bagging and boosting, are particularly effective in dealing with imbalanced classification problems.

5. **Modify the Loss Function**: Existing loss functions can be adapted by adding a penalty for misclassifying the minority class. This modification encourages the model to focus on the minority class. However, determining the optimal penalty value may require experimentation and fine-tuning.

6. **Treat as Anomaly Detection**: Reframe the classification problem as anomaly detection by treating the minority class as outliers. This method is well-suited for rare-event prediction tasks, such as fraud detection or identifying system failures.

It is crucial to emphasize that techniques for handling data imbalance should only be applied after the train/test split and should never be performed on the test dataset. If these techniques are applied to the test set, the machine learning model may appear to perform well during evaluation, but its real-world performance could be significantly worse. Finally, to address data imbalance, it is essential to split the data in a manner that maintains the class distribution, a method known as Stratified Split. This approach ensures that the relative proportions of the classes are preserved during the split. When performing cross-validation for model evaluation, it is important to use the StratifiedKFold cross-validator, as it guarantees that each fold maintains the percentage of samples for each class. Without this, there is a risk of having no samples from the minority class in some folds. [22]

Since the dataset includes one categorical variable (gender) and five continuous variables, the SMOTENC algorithm from the imbalanced-learn (imblearn) package is applied. [23] The resulting distribution is visualized in Figure 22.

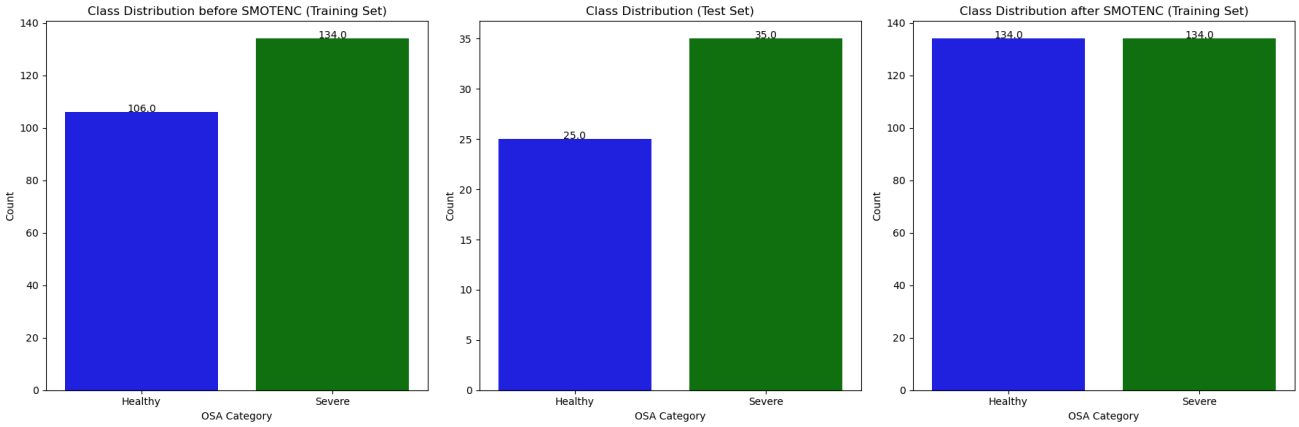

Figure 22: Barplots with the class distributions before and after applying the SMOTENC algorithm

4.3 EDA for clustering

Finally, exploratory data analysis (EDA) can be conducted without using the outcome variable: AHI for regression problems and OSA for classification problems. As a result, the continuous variables — weight, age, height, cervical, and BMI — together with the categorical variable gender will be the focus of this section.

Before proceeding with principal component analysis (PCA) or t-Distributed Stochastic Neighbouring Entities (t-SNE) analysis, it is essential to normalize these features. Since the variables may have different units of measurement, normalization ensures a consistent scale, enabling meaningful covariance analysis. Standardization, or Z-score normalization, is a commonly used feature-scaling method that transforms each feature to have a mean of 0 and a standard deviation of 1.

While tree-based models are largely unaffected by feature scaling, many other algorithms rely on normalized features for various reasons, such as improving convergence rates or achieving a more accurate model fit. Normalization can significantly influence the model's behavior, resulting in a substantially different fit compared to unscaled data. [25]

4.3.1 Principal Component Analysis (PCA)

Since classical principal component analysis (PCA) is less suitable for binary variables like gender, the gender column is excluded from further analysis [24].

Dimensionality reduction using PCA involves identifying the features that contribute most to the variance in the dataset. However, if one feature exhibits a larger variance than others due to differences in scale, PCA will disproportionately emphasize that feature, causing the principal components to be biased in its direction. To prevent this, it is essential to standardize the variables as described in previous section to ensure that all features have equal variance [25].

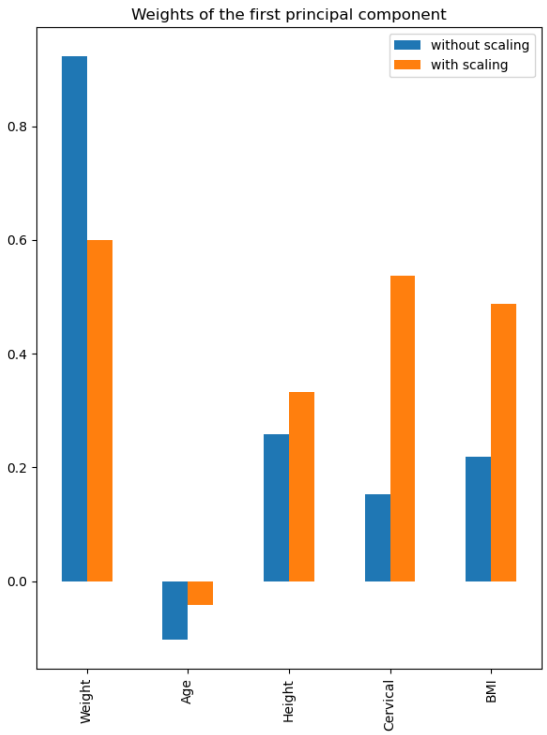

Figure 23: Barplot of the weights of the first PC for each of the continuous variables

To demonstrate the importance of scaling, PCA is performed using five principal components (PCs) - equal to the amount of features - both with and without normalization of the continuous features. As shown in figure 23, performing PCA on unscaled features (indicated in blue) results in the weight variable dominating the principal components due to its relatively high range, with values reaching up to 165 kg. In contrast, a variable like cervical circumference, with a maximum value of only 53 cm, contributes significantly less to the first PC.

However, when PCA is applied to the scaled features as seen in figure 23 in orange, the contributions of different variables are more evenly distributed, and the dominance of the weight variable is mitigated.

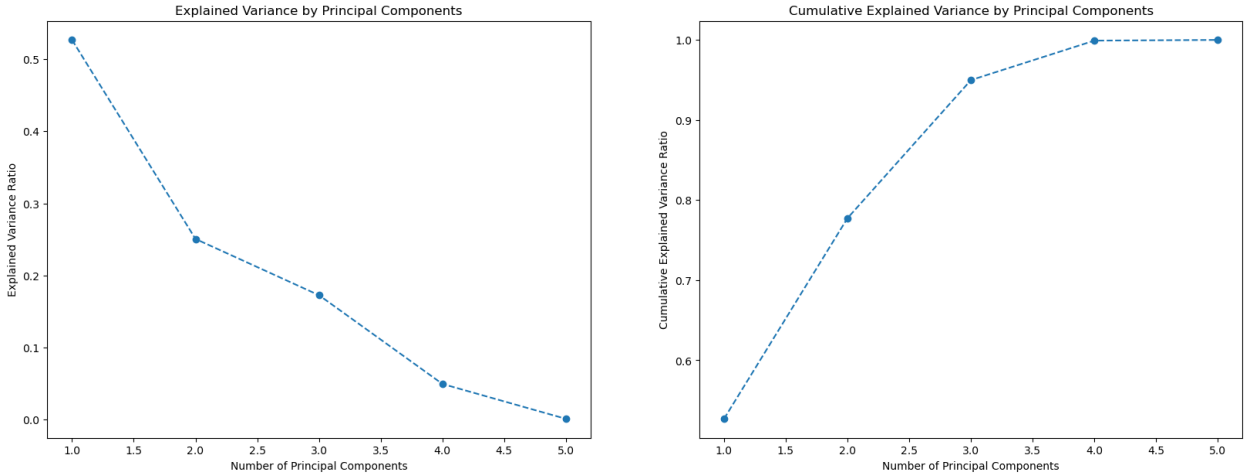

Figure 24: Elbow plot with explained variance (left) and cumulative explained variance (right) by the five principal components

By examining the (cumulative) explained variance in figure 24, the principal components (PCs) that capture the majority of the variance in the data can be identified, typically located at the "elbow" of the plot. Since the first three PCs collectively account for nearly 95% of the total variance, it was decided to retain only these three PCs for further analysis.

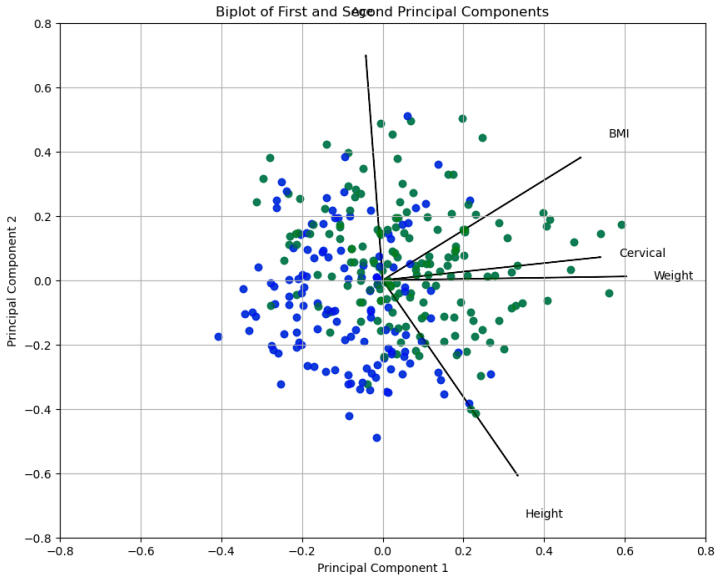

Figure 25: Biplot of first and second PC

The first two principal components (PCs) can be visualized in a scatterplot, as visualized in figure 25. To facilitate interpretation, the supervised labels of the OSA variable (e.g., "Healthy" or "Severe") can be used to annotate the scatter points, indicating their corresponding class. This PCA score plot reveals clusters of samples based on their similarity.

The scatterplot can be further enhanced by combining it with a loading plot, which illustrates how strongly each characteristic (vector) contributes to the principal components. [26] These loading plots are particularly useful for understanding the relationships between variables:

- When two vectors are close together, forming a small angle, the variables they represent are positively correlated.
- When vectors intersect at approximately 90°, the corresponding variables are likely uncorrelated.
- When vectors diverge to form a large angle (close to 180°), the variables are negatively correlated.

This combined visualization provides valuable insights into both the clustering of samples and the relationships between the contributing variables. For instance, it reveals that lower values of PC1 and PC2 are associated with a higher concentration of datapoints labeled as "Healthy". Additionally, the features cervical circumference and weight are shown to be positively correlated, while BMI and age appear to be almost uncorrelated.

These observations align with the results of a Spearman correlation analysis, as illustrated in Figure 10. Furthermore, increasing values of PC1 and PC2 correspond to rising values of BMI, cervical circumference, and weight, whereas age contributes minimally to PC1.

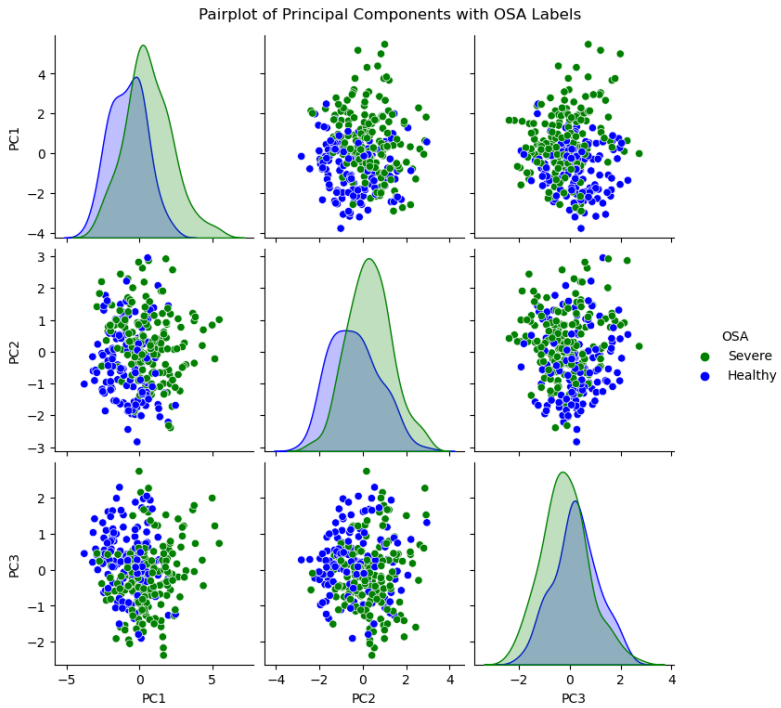

Figure 26: Pairplot of the first three PCs

Finally, a pairplot of all three principal components (PCs), along with the OSA labels, can be created, as shown in Figure 26. This visualization allows for a deeper exploration of the relationships between the PCs, and similar reasoning can be applied as with the previous plot to interpret the clusters and patterns.

Finally, the Gender feature is incorporated into the unsupervised exploratory data analysis (EDA) to evaluate how well PCA distinguishes between two categorical variables: OSA (Healthy or Severe) and Gender (Male or Female). The resulting pairplots are shown in Figure 27. Both plots are identical, as PCA is deterministic, with the only difference being the color coding: in the right plot, red dots represent females, and blue dots represent males.

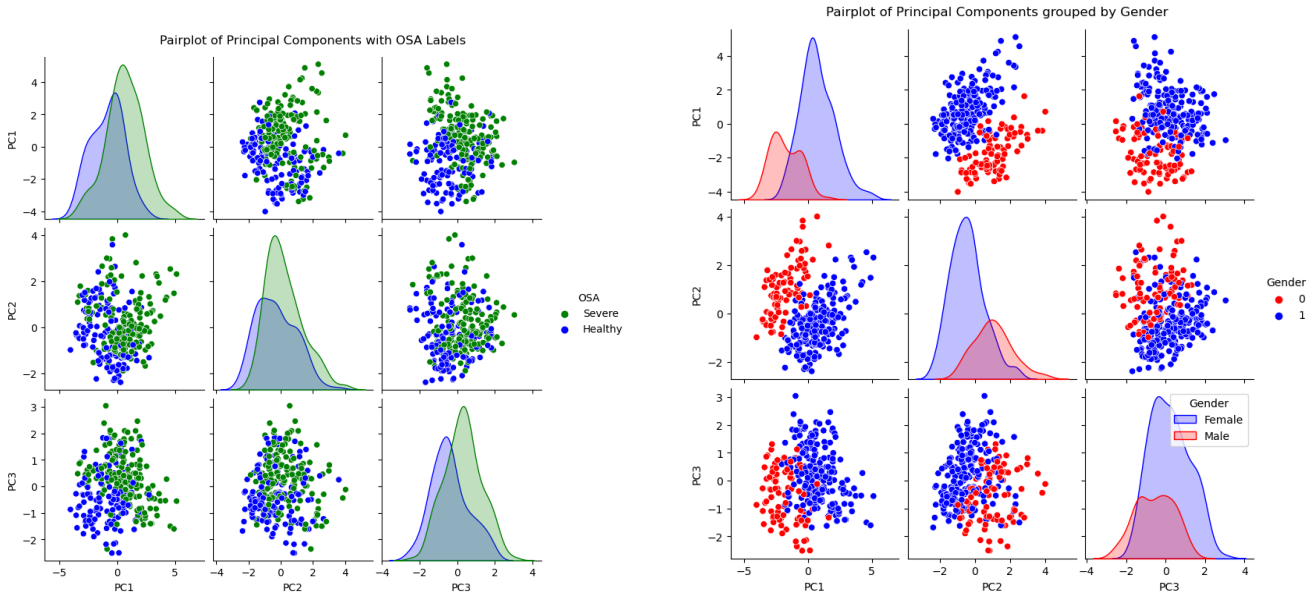

Figure 27: Pairplot of the first three PCA components including the Gender feature, grouped by OSA (left) and Gender (right)

4.3.2 t-Distributed Stochastic Neighbouring Entities (t-SNE)

As an alternative for dimensionality reduction, t-Distributed Stochastic Neighbor Embedding (t-SNE) minimizes the divergence between two distributions: one that measures the pairwise similarities of the input objects and another that measures the pairwise similarities of the corresponding low-dimensional points in the embedding. It is able to separate data that cannot be separated by a straight line, making it a non-linear dimensionality reduction technique.

t-SNE is an excellent tool for gaining insights into high-dimensional datasets. However, while it is highly effective for visualizing complex data, its plots can sometimes be difficult to interpret or even misleading. Moreover, t-SNE is non-deterministic and iterative, meaning that each run can produce a different result, in contrast with PCA which yield fixed PC results. [27]

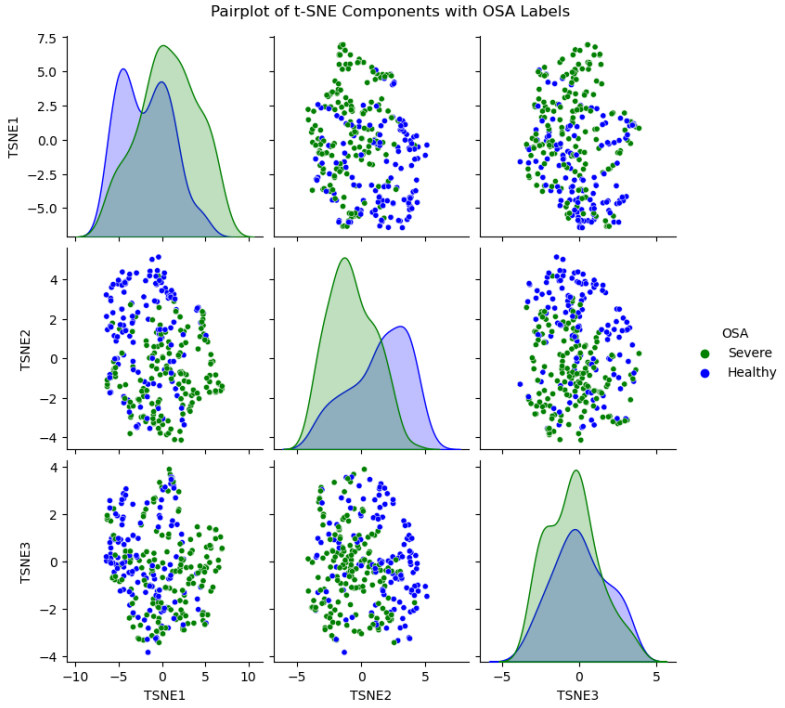

Figure 28: Pairplot of the first three t-SNE componenets

A similar pairplot can now be created, as shown in Figure 28.

To further evaluate the effectiveness of t-SNE, the Gender column is included to assess its ability to distinguish between male and female patients. As illustrated in figure 28 and compared with the PCA-based pairplot in figure 26, t-SNE demonstrates a clear advantage in separating patients by gender. In the t-SNE plot, red dots represent females, while blue dots represent males, showing a distinct separation between the two groups. This indicates that t-SNE is more suitable for this specific analysis than PCA.

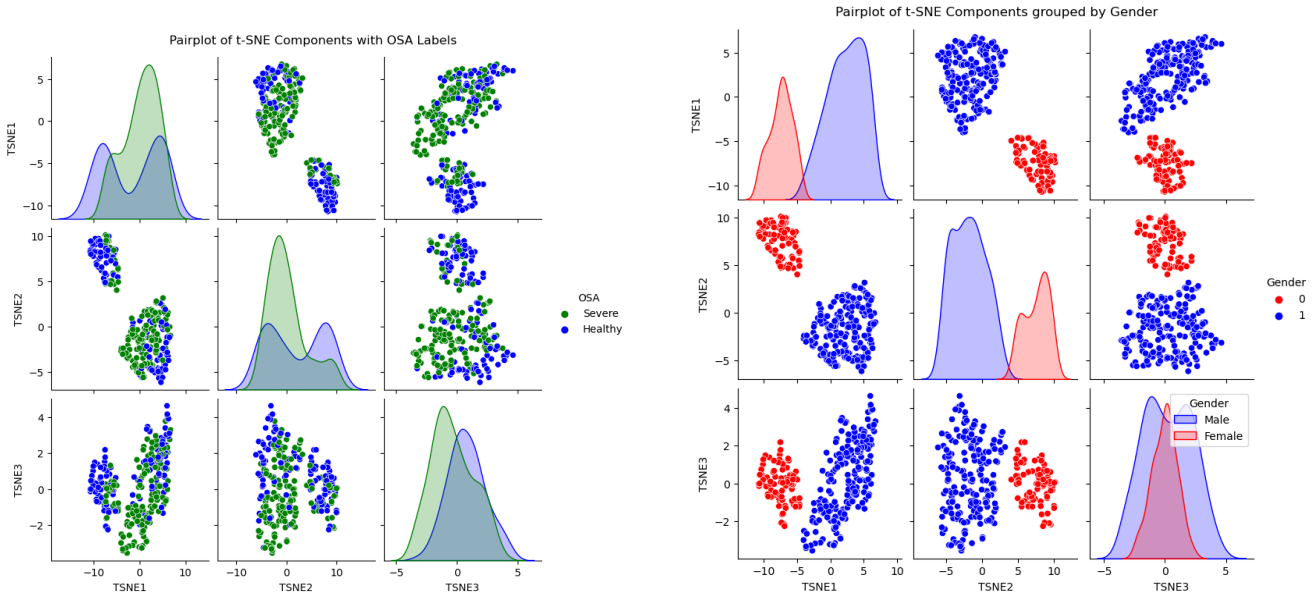

Figure 29: Pairplot of the first three t-SNE components including the Gender feature, grouped by OSA (left) and Gender (right)

5 Machine learning models

In this section, a description will be given of the pipeline that is followed in order to train and test different ML models.

5.1 Pipeline for model training and testing

5.1.1 Feature scaling

Feature scaling can be achieved through two primary methods: normalization and standardization. The primary purpose of feature scaling is to prevent any single feature from disproportionately influencing the gradient of the loss function during training of the ML models. While some algorithms such as tree-based models, handle unscaled features better than others, significant differences in feature scales can lead to suboptimal performance and computational inefficiencies. [28, 29, 31]

Normalization involves rescaling each feature to fit within a fixed range, typically between 0 and 1 or -1 and 1. [30] This is achieved by subtracting the minimum value of the feature and dividing it by the range, which is the difference between the maximum and minimum values. Normalization is particularly effective when the distribution of the data is unknown or does not follow a Gaussian (normal) pattern. It is also essential for distance-based algorithms, such as K-Nearest Neighbors (K-NN), where unscaled features with larger magnitudes could dominate distance computations and skew results. Finally, there exist different types of normalization, such as min-max, log or mean normalization.

Standardization or Z-score normalization, on the other hand, transforms features to have a mean of 0 and a standard deviation of 1 by subtracting the mean and dividing by the standard deviation of each feature. [31] This method is particularly beneficial for gradient-based algorithms like Support Vector Machines (SVMs) and neural networks (NNs), where balanced feature scales improve optimization and convergence. While models like linear regression and logistic regression do not require standardization by design, it can still enhance performance when features have varying magnitudes. Standardization is also crucial for dimensionality reduction techniques like Principal Component Analysis (PCA), which aims to identify directions of maximum variance in the data. Unlike mean normalization, standardization accounts for both the mean and variance, ensuring that features with large magnitudes do not distort the analysis.

The difference between those two feature scaling methods can be seen in figure 30.

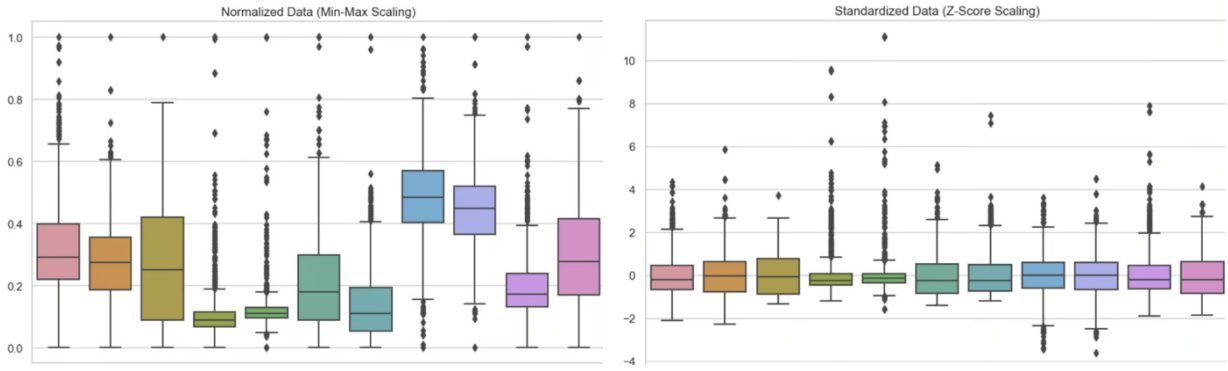

Figure 30: Graphical representation of the difference between feature normalization (left) and standardization (right) [29]

It is crucial to scale the data only after splitting it into training and testing sets to maintain the integrity of the testing set as an unbiased evaluation metric. This approach simulates the real-world scenario where the model encounters unseen data during deployment. The (Z-score) normalization process involves calculating statistical parameters, such as the minimum, maximum, mean, or standard deviation, from the training set and then applying those parameters

to normalize the testing set. This ensures that the testing set remains completely separate from the training process. Additionally, the same normalization technique used during training must be consistently applied to any new, unseen data during deployment to ensure reliable and accurate model predictions. [32]

5.1.2 Dataset splitting

The first step in training a machine learning model is to split the dataset into training and testing sets. This is to prevent data leakage from the test set into the training phase of the model. Using the same data for both training and testing would result in overly optimistic evaluations of the model's generalization abilities, resulting in a methodological error. In practice, such a model would perform poorly on truly unseen data — that is, data not encountered during training. This issue is known as overfitting. [33]

Therefore, to avoid overfitting, it is standard practice in supervised machine learning experiments to reserve a portion of the available data as a test set. The training set is used to learn the model, while the test set is used independently to evaluate its performance. This ensures that the model is assessed on data it has not encountered during training. After splitting the data, feature scaling can be applied following the procedures outlined in the previous section.

5.1.3 (Nested) Cross-validation

Even with a standard train-test split, the model's performance metrics can remain sensitive to the specific test set chosen. A fortunate selection might produce an overly optimistic estimate of the generalization error, while an unfortunate one could result in an overly pessimistic evaluation. To reduce this variability and obtain a more reliable estimate of model performance, **cross-validation** is commonly employed. This technique involves repeating the training and evaluation process multiple times with different data splits and averaging the results to achieve a robust assessment of the model's generalization capabilities. [34]

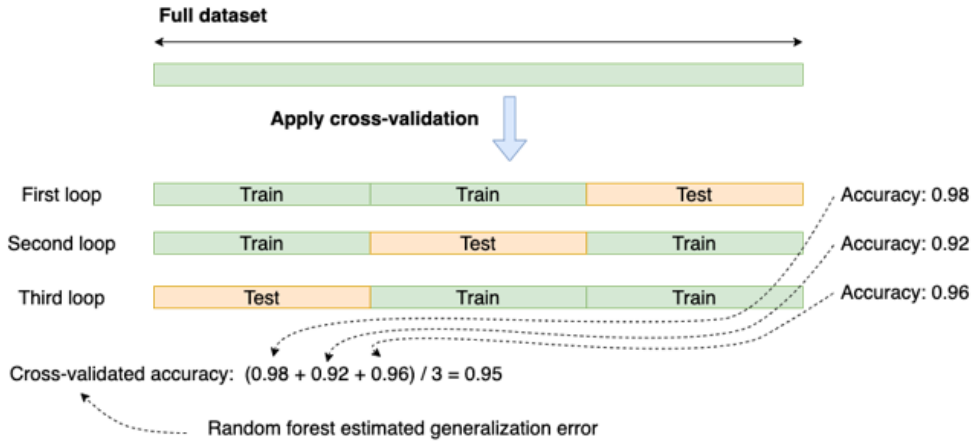

Figure 31: Schematic overview of a 3-fold cross-validation applied to the training of a Random Forest [34]

A popular method in practice is K-fold cross-validation. In this approach, the dataset is randomly divided into K disjoint folds, each containing approximately the same number of instances. Each fold is used once as a test set, while the remaining $K-1$ folds serve as the training set. This ensures that every instance in the dataset is used for both training and testing at some point. [35] K-fold cross-validation is useful for tasks such as optimizing a model's hyperparameters and comparing or selecting models for a given dataset. A graphical representation of a 3-fold cross-validation procedure applied to the training of a Random Forest can be seen in figure 31.

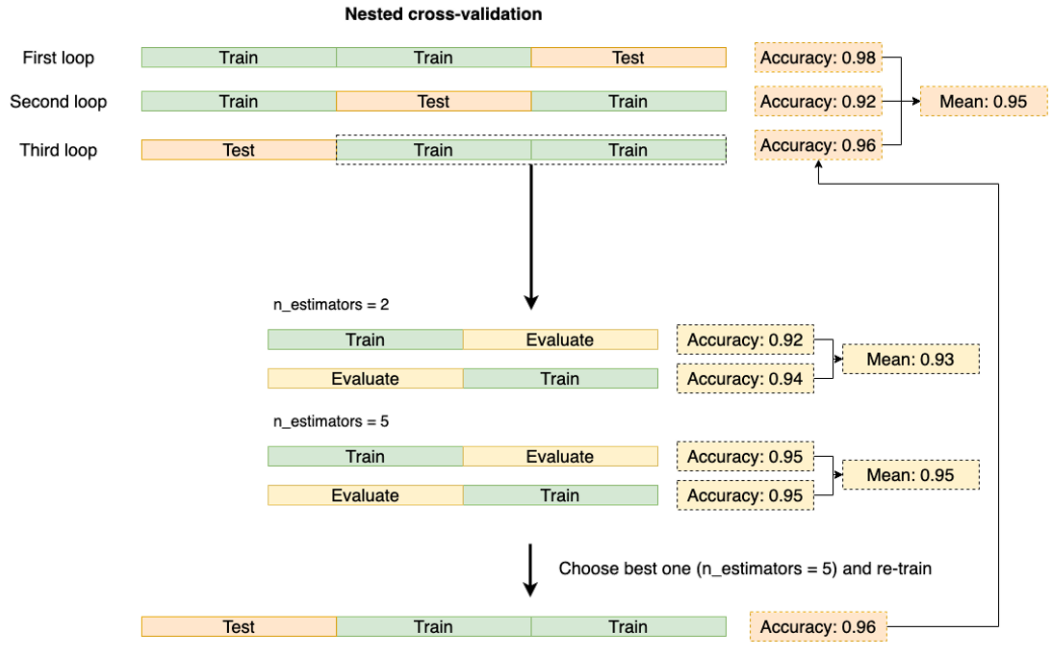

Figure 32: Schematic overview of a nested cross-validation with 3-folds in the outer CV loop and 2-folds in the inner CV loop, applied to the training of a Random Forest [34]

However, if the same cross-validation procedure and dataset are used to both tune hyperparameters and perform model selection, the evaluation may be optimistically biased and information may thus "leak" into the model. This form of selection bias results in overfitting and make the results unreliable. [36]

To address this issue, **nested cross-validation** is recommended. This method involves nesting the hyperparameter optimization process within an outer cross-validation loop. By separating the hyperparameter tuning from the model selection process, nested cross-validation provides an unbiased evaluation of tuned machine learning models and ensures a fair comparison between them. This double cross-validation approach is widely regarded as the gold standard for robust model evaluation. [37] A schematic overview of this nested CV can be seen in figure 32.

The distinction between nested and non-nested cross-validation becomes evident when training a model for tasks such as classification. In this experiment, a Support Vector Classifier (SVC) was trained on the OSA dataset to classify cases as Healthy or Severe.

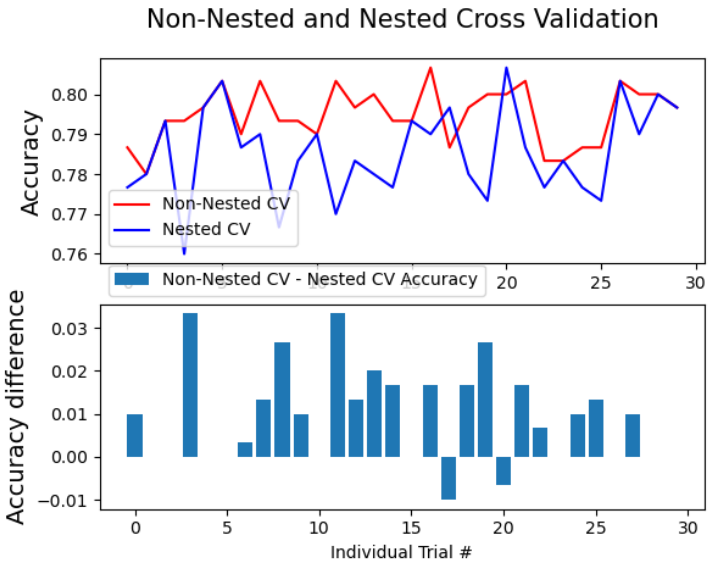

Figure 33: Plots showing the difference in accuracy between using nested versus non-nested cross-validation [38]

5.1.4 Conclusion for model training and testing

As illustrated in figure 33, the non-nested cross-validation approach (indicated in red) produces overly optimistic accuracy scores. This occurs because non-nested CV uses the same dataset to tune the model parameters and evaluate its performance. Consequently, information from the test set may inadvertently "leak" into the model during the training process, leading to overfitting and an inflated sense of the model's predictive capabilities. By contrast, nested CV mitigates this issue by ensuring a strict separation between the data used for model tuning and that used for performance evaluation, resulting in a more reliable and unbiased assessment of the model. [36, 38]

So, in the training and testing of the ML models, a nested cross-validation (CV) procedure will be employed to ensure robust evaluation and hyperparameter optimization. This procedure consists of two key components:

- **Outer CV Loop:** The outer loop, implemented using the 'KFold' function from 'sklearn' with the number of splits set to 5, is responsible for splitting the entire dataset into a training+validation set and a test set with a ratio of respectively 80% and 20%. This loop serves the purpose of model selection, as it evaluates the performance of the model on unseen data (i.e., the test set) in each iteration. The outer loop can be implemented manually with a 'for'-loop iterating over the splits provided by the 'KFold' function or automatically using the 'cross_val_score' function from 'sklearn'.
- **Inner CV Loop:** The inner loop also employs the 'KFold' function with again 5 splits, but it operates exclusively on the training+validation set, further splitting it into separate training and validation subsets in a ratio of resp. 80% and 20%. This loop is dedicated to hyperparameter optimization. Hyperparameter tuning can be performed using methods like grid search or random search, both of which are available in 'sklearn'. Random search is useful for narrowing down the range of potential values for each hyperparameter by sampling randomly from a specified distribution. Once the range is refined, 'GridSearchCV' can be employed to evaluate every possible combination of hyperparameter settings within the defined grid. This systematic approach ensures that the model is tested with all relevant parameter combinations, leading to the selection of the optimal configuration for the given dataset.

By combining these two loops, the nested CV procedure provides an unbiased estimate of the model's performance while simultaneously optimizing its hyperparameters, offering a rigorous and reliable framework for machine learning experiments executed in the next sections.

5.2 ML models for Regression

The following section explains the machine learning models used to predict the apnea-hypopnea index (AHI) based on the features of age, cervical circumference, BMI, and gender. Additionally, the results of the evaluation of these models will be presented.

5.2.1 Regression evaluation metrics

- **R^2 (Coefficient of Determination)**: Measures the proportion of the variance in the dependent variable (i.e. AHI) that is predictable from the independent variables (i.e. Age, Cervical, BMI and gender). A value close to 1 indicates a good fit.
- **MAE (Mean Absolute Error)**: Represents the average of the absolute differences between predicted and actual values. It measures the magnitude of errors without considering their direction.
- **MSE (Mean Squared Error)**: Calculates the average of the squared differences between predicted and actual values. It penalizes larger errors more than smaller ones, making it sensitive to outliers.
- **RMSE (Root Mean Squared Error)**: The square root of the mean squared error, providing an error metric in the same units as the target variable. It is more interpretable compared to MSE and emphasizes larger errors.

For the evaluation of the best configuration of the prediction models, the RMSE metric is chosen as it is more interpretable than the MSE and is sensitive to outliers.

5.2.2 Prediction models

Multiple Linear Regression/Generalized Linear Model

Multiple Linear Regression is a statistical technique that models the relationship between a dependent variable (AHI) and multiple independent variables by fitting a linear equation to the observed data.
Below, the hyperparameters to be tuned in Multiple Linear Regression can be found:

- **fit_intercept**: Whether to calculate the intercept for this model. If set to False, no intercept will be used in calculations.

The best configuration from the nested cross-validation is:

- **fit_intercept**: True

These are the corresponding performance metrics:

	R2	MAE	MSE	RMSE
Value	0.06	12.87	249.12	15.78

Table 1: Performance Metrics for the Best Multiple Linear Regression Model

The best Multiple Linear Regression model uses the `fit_intercept` parameter set to True, achieving a balanced performance across R2, MAE, MSE, and RMSE.

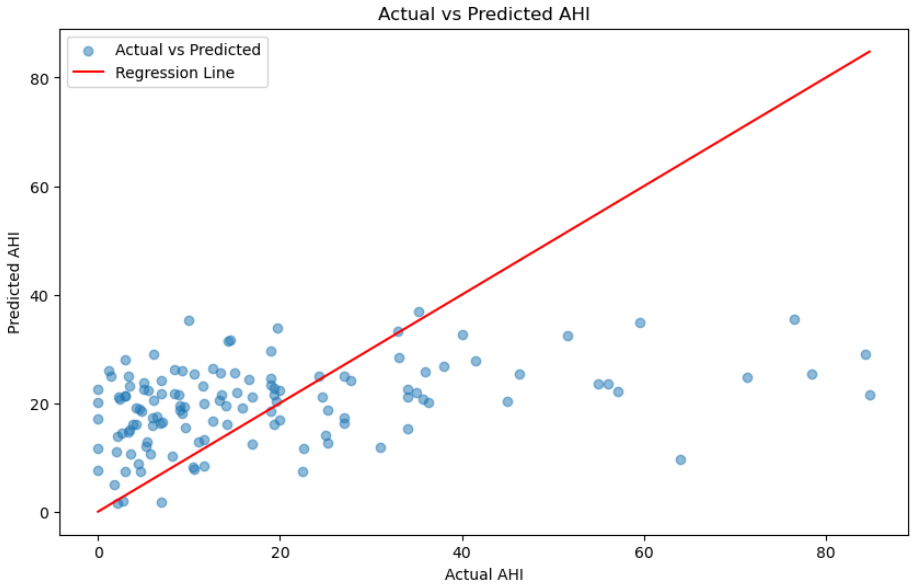

Figure 34: Plot of regression line (red) obtained from Multiple Linear Regression

In addition, a plot of the regression line is constructed that shows the predicted versus the actual AHI values. As seen in figure 34, predicting the AHI using a linear formula is not suitable in this case, as the relationship between the dependent and independent variables does not follow a linear relationship.

The linear regression equation for predicting AHI is given by:

$$\text{AHI} = 1.50 \times \text{Gender} + 12.60 \times \text{Age} + 32.45 \times \text{Cervical} + 27.85 \times \text{BMI} \tag{1}$$

Moreover, a residual plot can be generated, as shown in 35. Residuals are calculated as the difference between the observed AHI and the predicted AHI. Positive residual values (on the y-axis) indicate that the prediction was too low, while negative values suggest the prediction was too high. A residual of zero means the prediction was exactly correct. If a clear pattern or trend is observed in the residual plot, it suggests that the model could be improved, as it may not have fully captured the underlying data structure. [39]

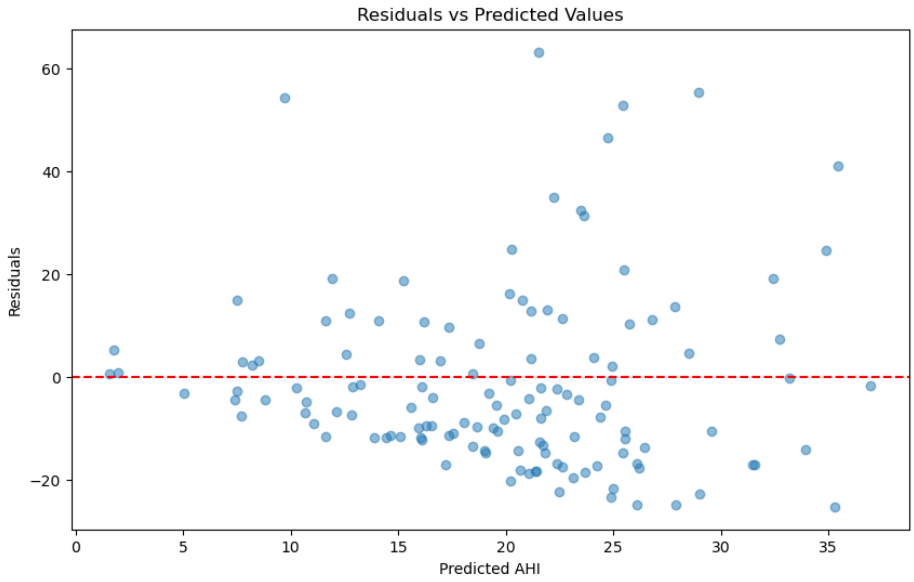

Figure 35: Plot of residuals

Now, the multiple linear regression model is extended by including interaction terms between the different independent variables. Below, the performance metrics for this linear regression model with interaction terms are shown:

	R2	MAE	MSE	RMSE
Value	0.20	12.39	270.13	16.25

Table 2: Performance Metrics for the Linear Regression Model with Interaction Terms

It is clear that the performance of these linear regression models (with and without consideration of the interaction terms) are better when comparing to a naive predictor, that results in following performance metrics:

	R2	MAE	MSE	RMSE
Value	-0.02	14.85	357.65	18.91

Table 3: Performance Metrics for the Naive Predictor

Finally, when comparing the initial linear regression model (without interaction terms) for male and female patients separately, it is evident that the model's predictions are more accurate for male patients. This is reflected in the following evaluation scores:

	R2	MAE	MSE	RMSE
Male	0.17	12.98	274.83	16.46
Female	0.02	11.58	292.30	16.71

Table 4: Performance Metrics for Best Multiple Linear Regression Model grouped by Gender

To assess whether the differences between male and female patients are statistically significant, a t-test is conducted, assuming both datasets are normally distributed and have equal variances. The two-sample t-test yields a p-value of 0.26, which suggests that we fail to reject the null hypothesis. Therefore, the gender-based differences are not statistically significant.

Lasso Linear Regression

Lasso Linear Regression is a type of linear regression that uses L1 regularization. It adds a penalty equal to the absolute value of the magnitude of coefficients, which can result in sparse models with fewer coefficients.
Below, the hyperparameters to be tuned in Lasso Linear Regression can be found:

- **alpha**: Constant that multiplies the L1 term. It controls the strength of the regularization.

The best configuration from the nested cross-validation is:

- **alpha**: 0.01

These are the corresponding performance metrics:
The best Lasso Linear Regression model uses the `alpha` parameter set to 0.01, achieving a balanced performance across R2, MAE, MSE, and RMSE.

	R2	MAE	MSE	RMSE
Value	0.06	12.86	248.48	15.76

Table 5: Performance Metrics for the Best Lasso Linear Regression Model

Decision Tree Regression

Decision Tree Regression is a non-parametric model that splits the data into subsets based on the value of input features. It creates a tree-like model of decisions and their possible consequences, including outcomes, resource costs, and utility.
Below, the hyperparameters to be tuned in Decision Tree Regression can be found:

- **max_depth**: The maximum depth of the tree. It controls the maximum number of levels in the tree.
- **min_samples_split**: The minimum number of samples required to split an internal node.
- **min_samples_leaf**: The minimum number of samples required to be at a leaf node.

The best configuration from the nested cross-validation is:

- **max_depth**: 10
- **min_samples_split**: 10
- **min_samples_leaf**: 4

These are the corresponding performance metrics:

	R2	MAE	MSE	RMSE
Value	-0.06	12.92	280.43	16.75

Table 6: Performance Metrics for the Best Decision Tree Regression Model

The best Decision Tree Regression model uses a maximum depth of 10, a minimum of 10 samples to split an internal node, and a minimum of 4 samples at a leaf node, achieving a balanced performance across R2, MAE, MSE, and RMSE. However, the negative value for the R^2 score indicates that the regression model is not able to predict the AHI from the given features.

Random Forest Regression

Random Forest Regression is an ensemble learning method that constructs multiple decision trees during training and outputs the mean prediction of the individual trees. It improves the predictive accuracy and controls overfitting.
Below, the hyperparameters to be tuned in Random Forest Regression can be found:

- **max_depth**: The maximum depth of the tree. It controls the maximum number of levels in the tree.
- **min_samples_split**: The minimum number of samples required to split an internal node.
- **min_samples_leaf**: The minimum number of samples required to be at a leaf node.
- **n_estimators**: The number of trees in the forest.

The best configuration from the nested cross-validation is:

- **max_depth**: 10
- **min_samples_split**: 10
- **min_samples_leaf**: 1
- **n_estimators**: 300

These are the corresponding performance metrics:

	R2	MAE	MSE	RMSE
Value	0.12	12.43	234.30	15.31

Table 7: Performance Metrics for the Best Random Forest Regression Model

The best Random Forest Regression model uses a maximum depth of 10, a minimum of 10 samples to split an internal node, a minimum of 1 sample at a leaf node, and 300 trees in the forest, achieving a balanced performance across R2, MAE, MSE, and RMSE.

XGBoost

XGBoost Regression is an optimized distributed gradient boosting library designed to be highly efficient, flexible, and portable. It implements machine learning algorithms under the Gradient Boosting framework.
Below, the hyperparameters to be tuned in XGBoost Regression can be found:

- **learning_rate**: Step size shrinkage used to prevent overfitting.
- **max_depth**: The maximum depth of a tree. It controls the maximum number of levels in the tree.
- **n_estimators**: The number of boosting rounds.

The best configuration from the nested cross-validation is:

- **learning_rate**: 0.01
- **max_depth**: 3
- **n_estimators**: 250

These are the corresponding performance metrics:

	R2	MAE	MSE	RMSE
Value	0.18	12.36	218.63	14.79

Table 8: Performance Metrics for the Best XGBoost Regression Model

The best XGBoost Regression model uses a learning rate of 0.01, a maximum depth of 3, and 250 boosting rounds, achieving a balanced performance across R2, MAE, MSE, and RMSE.

Support Vector Machine (SVM)

Support Vector Machine (SVM) Regression is a type of regression that uses the SVM algorithm to predict continuous values. It finds the hyperplane that best fits the data while minimizing the error within a specified margin.
Below, the hyperparameters to be tuned in SVM Regression can be found:

- **C**: Regularization parameter. The strength of the regularization is inversely proportional to C. Smaller values specify stronger regularization.
- **gamma**: Kernel coefficient for 'rbf', 'poly', and 'sigmoid'. It defines how far the influence of a single training example reaches.
- **kernel**: Specifies the kernel type to be used in the algorithm ('linear', 'poly', 'rbf', 'sigmoid').

The best configuration from the nested cross-validation is:

- **C**: 1000
- **gamma**: 0.1
- **kernel**: `rbf`

These are the corresponding performance metrics:

	R2	MAE	MSE	RMSE
Value	0.19	11.37	215.14	14.67

Table 9: Performance Metrics for the Best SVM Regression Model

The best SVM Regression model uses a regularization parameter (`C=1000`), a gamma of 0.1, and the `rbf` kernel, achieving a balanced performance across R2, MAE, MSE, and RMSE.

Multi-Layer Perceptron

Multi-Layer Perceptron (MLP) Regression is a class of feedforward artificial neural network (ANN) that consists of at least three layers of nodes: an input layer, a hidden layer, and an output layer. Each node, except for the input nodes, is a neuron that uses a nonlinear activation function.
Below, the hyperparameters to be tuned in MLP Regression can be found:

- **activation**: Activation function for the hidden layer ('identity', 'logistic', 'tanh', 'relu').
- **alpha**: L2 penalty (regularization term) parameter.
- **hidden_layer_sizes**: The number of neurons in the hidden layers.
- **solver**: The solver for weight optimization ('lbfgs', 'sgd', 'adam').

The best configuration from the nested cross-validation is:

- **activation**: `relu`
- **alpha**: 0.0001
- **hidden_layer_sizes**: (50,)

	R2	MAE	MSE	RMSE
Value	0.10	12.49	239.06	15.46

Table 10: Performance Metrics for the Best Multi-Layer Perceptron Regression Model

- **solver**: `sgd`

These are the corresponding performance metrics:
The best MLP Regression model uses the `relu` activation function, an alpha of 0.0001, 50 neurons in the hidden layer, and the `sgd` solver, achieving a balanced performance across R2, MAE, MSE, and RMSE.

5.3 ML models for Classification

The following section outlines the machine learning models used to classify the apnea-hypopnea index (AHI) into healthy (AHI < 5) and severe (AHI ≥ 30) using the features of age, cervical circumference, BMI, and gender. Additionally, the evaluation results of these models will be presented.

5.3.1 Classification evaluation metrics

The metrics that are going to be used to evaluate the performance of the ML models for classification are listed below:

- **Accuracy**: The proportion of correctly classified instances out of the total instances. It is a measure of overall correctness.
- **ROC AUC (Receiver Operating Characteristic - Area Under Curve)**: Measures the ability of a classifier to distinguish between classes by evaluating the trade-off between true positive rate and false positive rate. A higher value indicates better classification performance.
- **Precision**: The proportion of true positive predictions out of all positive predictions made by the model. It indicates how many predicted positives are correct.
- **Recall/sensitivity**: The proportion of true positives correctly identified out of all actual positives. It reflects the model's ability to detect all relevant instances.
- **F1 Score**: The harmonic mean of precision and recall, providing a single measure that balances these two metrics. It is a more robust metrics that is particularly useful for imbalanced datasets.

During the nested cross-validation, the best configuration of the classifier is determined based on the F1 Score, as it is a robust metric in case of imbalanced datasets. However, note that the SMOTENC algorithm is performed to handle the imbalance in the training data. Therefore, the precision or recall will have similar values as the F1 score.

Machine learning classifiers can be broadly categorized into discriminative, generative, non-parametric, and hybrid models. Each type has unique characteristics and applications, which are briefly explained below.

Discriminative models focus on learning the decision boundary between classes by directly modeling the conditional probability $p(y|X)$. They do not attempt to model the underlying data distribution but instead aim to minimize classification errors. These models are well-suited for tasks where the primary goal is to predict class labels accurately. Examples include logistic regression, support vector machines (SVMs), and tree-based models.

Generative models aim to capture the joint probability distribution $p(X, y)$, which allows them to generate new data points and derive the conditional probability $p(y|X)$. These models build a representation of how the data was generated, making them useful for understanding the structure of the data and handling missing values. Examples include Naïve Bayes, Gaussian Mixture Models (GMM), and Linear or Quadratic Discriminant Analysis (LDA/QDA).

Non-parametric models, such as K-Nearest Neighbors (K-NN), do not make assumptions about the underlying data distribution. Instead, they rely on the data directly to make predictions. In K-NN, the class label of a new data point is determined by the majority vote of its K nearest neighbors in the feature space. These models are simple, intuitive, and effective for problems with non-linear decision boundaries but can be computationally expensive for large datasets.

Hybrid models combine the strengths of multiple classifiers to improve overall performance. For example, the Soft Voting Classifier is an ensemble method that aggregates the predicted probabilities from multiple base classifiers and selects the class with the highest average probability. Hybrid models are particularly useful when individual classifiers have complementary strengths and weaknesses, as the ensemble often provides a more robust and accurate prediction.

5.3.2 Discriminative Models

Multiple Logistic Regression

Logistic Regression is a binary classification algorithm that models the probability of a target variable belonging to a class using the logistic function. It outputs probabilities that map to two possible discrete outcomes, in this case Healthy or Severe.
Below, the hyperparameters to be tuned in logistic regression can be found:

- **C**: Inverse of regularization strength. Smaller values specify stronger regularization.
- **penalty**: Type of regularization (`l1`/the lasso or `l2`/the ridge).
- **solver**: Algorithm for optimization (`liblinear` is suitable for small datasets).

The best configuration from the nested cross-validation is:

- **C**: 1
- **penalty**: `l2`
- **solver**: `liblinear`

These are the corresponding performance metrics:

	Accuracy	ROC AUC	F1 Score	Precision	Recall
Value	0.85	0.85	0.86	0.88	0.85

Table 11: Performance Metrics for the Best Logistic Regression Model

Figure 36 presents the confusion matrix alongside an Age-BMI plot. The Age-BMI plot visualizes the labeled data points, with blue representing healthy patients and green representing severe patients. Additionally, the plot illustrates the decision boundary constructed by the Best Logistic Regression Model. The blue-shaded region corresponds to areas classified as "predicted healthy" by the model, while the green-shaded region represents areas classified as "predicted severe."

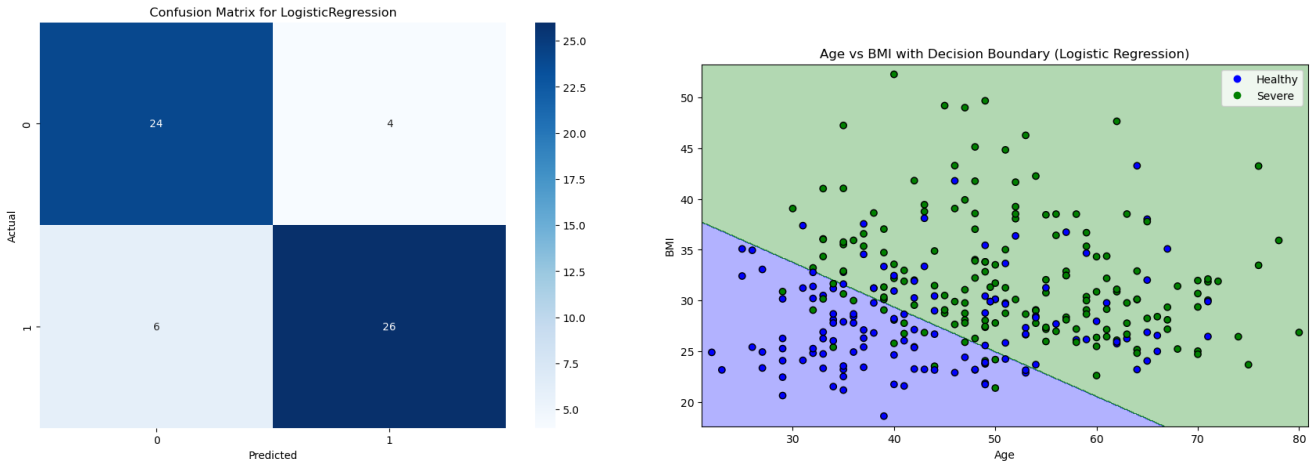

Figure 36: Confusion matrix (left) and annotated plot (Age vs BMI) with decision boundary (right) of the Best Logistic Regression Model

In addition, the Receiver Operating Characteristic (ROC) curve can be plotted, as well as the area under the ROC curve (AUC) can be calculated. This is visualized in figure 37 in orange, whereas the diagonal dotted line denotes the ROC curve of a random classifier. [40]

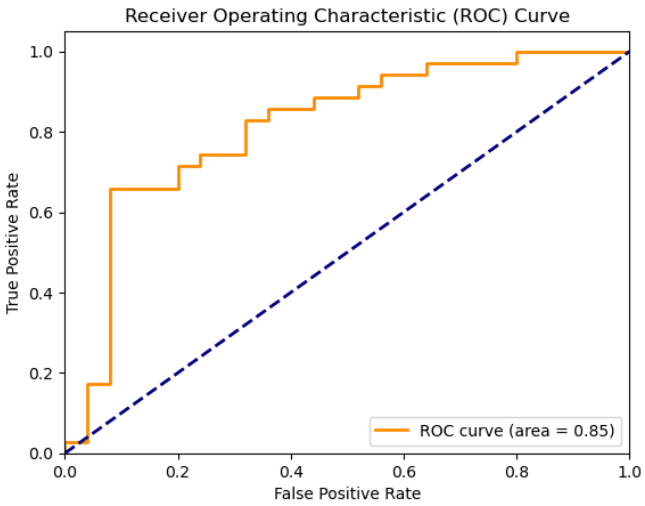

Figure 37: Receiver Operating Characteristic curve for the Best Logistic Regression Model (orange)

The best logistic regression model uses moderate regularization (`C=1`), `l2` penalty, and the `liblinear` solver, achieving a balanced performance across accuracy, precision, and recall.

Multi-Layer Perceptron

Multi-Layer Perceptron (MLP) is a class of feedforward artificial neural network (ANN). It consists of at least three layers of nodes: an input layer, a hidden layer, and an output layer. Each node, except for the input nodes, is a neuron that uses a nonlinear activation function.
Below, the hyperparameters to be tuned in MLP can be found:

- **activation**: Activation function for the hidden layer ('identity', 'logistic', 'tanh', 'relu').
- **alpha**: L2 penalty (regularization term) parameter.
- **hidden_layer_sizes**: The number of neurons in the hidden layers.
- **learning_rate**: Learning rate schedule for weight updates ('constant', 'invscaling', 'adaptive').
- **max_iter**: Maximum number of iterations.
- **solver**: The solver for weight optimization ('lbfgs', 'sgd', 'adam').

The best configuration from the nested cross-validation is:

- **activation**: `tanh`
- **alpha**: 0.0001
- **hidden_layer_sizes**: (100,)
- **learning_rate**: `constant`
- **max_iter**: 500
- **solver**: `adam`

These are the corresponding performance metrics:

	Accuracy	ROC AUC	F1 Score	Precision	Recall
Value	0.85	0.85	0.86	0.9	0.82

Table 12: Performance Metrics for the Best Multi-Layer Perceptron Model

Figure 38 displays the confusion matrix along with an Age-BMI plot containing the decision boundary generated by the Best Multi-Layer Perceptron Model. In the plot, blue represents healthy individuals (either true or predicted), while green denotes severe patients (true or predicted).

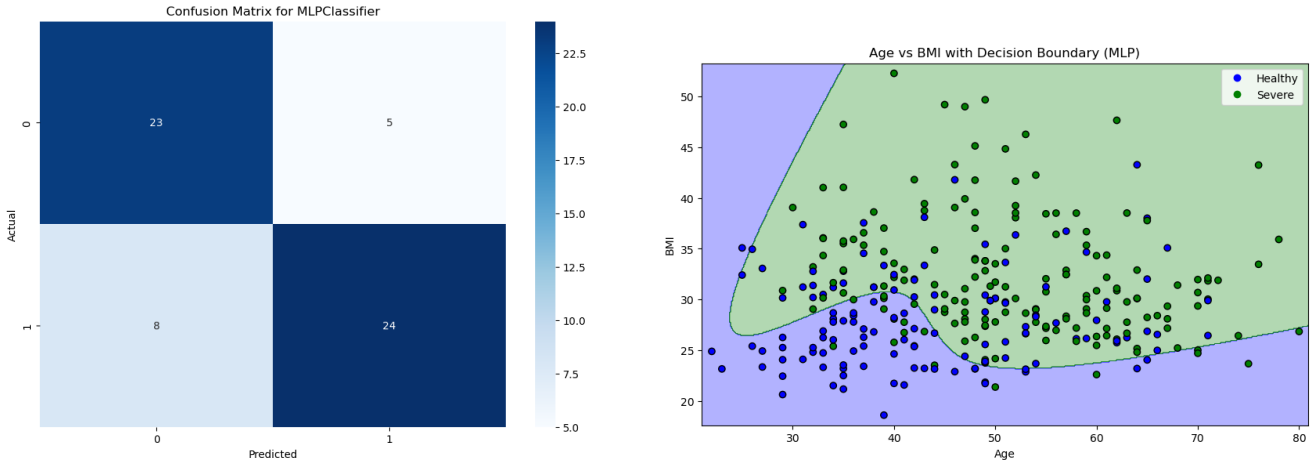

Figure 38: Confusion matrix (left) and annotated plot (Age vs BMI) with decision boundary (right) of the Best Multi-Layer Perceptron Model

The best MLP model uses the `tanh` activation function, an alpha of 0.0001, 100 neurons in the hidden layer, a constant learning rate, a maximum of 500 iterations, and the `adam` solver.

Support Vector Machine (SVM)

Support Vector Machine (SVM) is a powerful classification algorithm that finds the hyperplane that best separates the classes in the feature space. It can handle both linear and non-linear classification tasks using different kernel functions.
Below, the hyperparameters to be tuned in SVM can be found:

- **C**: Regularization parameter. The strength of the regularization is inversely proportional to C. Smaller values specify stronger regularization.
- **gamma**: Kernel coefficient for 'rbf', 'poly', and 'sigmoid'. It defines how far the influence of a single training example reaches.
- **kernel**: Specifies the kernel type to be used in the algorithm ('linear', 'poly', 'rbf', 'sigmoid').

The best configuration from the nested cross-validation is:

- **C**: 10
- **gamma**: `scale`

- **kernel**: `rbf`

These are the corresponding performance metrics:

	Accuracy	ROC AUC	F1 Score	Precision	Recall
Value	0.85	0.85	0.87	0.85	0.88

Table 13: Performance Metrics for the Best Support Vector Machine Model

Figure 39 displays the confusion matrix along with an Age-BMI plot containing the decision boundary generated by the Best Support Vector Machine Model. In the right plot, blue represents healthy individuals (either true or predicted), while green denotes severe patients (true or predicted).

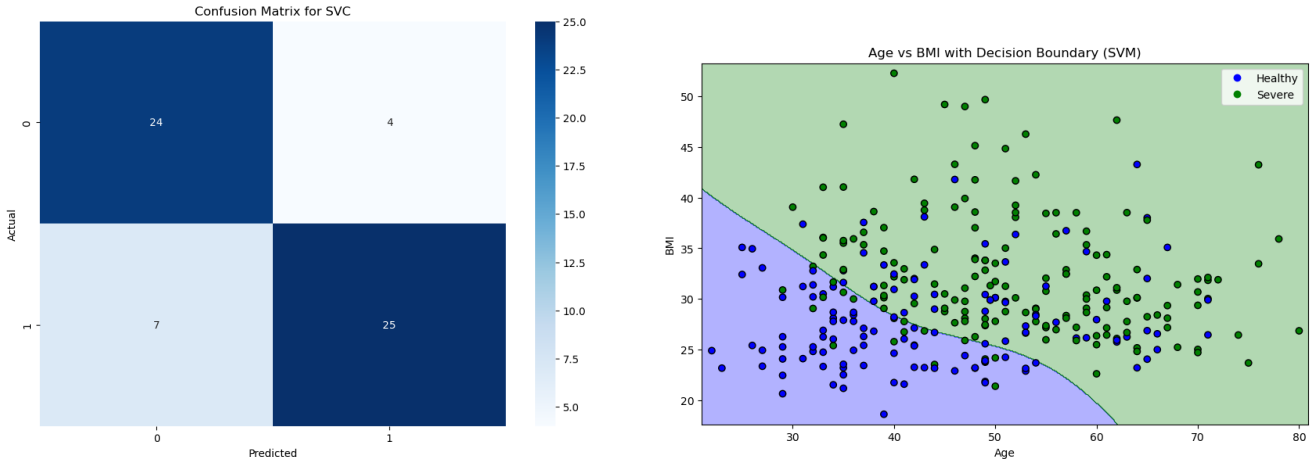

Figure 39: Confusion matrix (left) and annotated plot (Age vs BMI) with decision boundary (right) of the Best Support Vector Machine Model

The best SVM model uses a regularization parameter (`C=10`), `scale` gamma, and the `rbf` kernel, achieving a balanced performance across accuracy, precision, and recall.

Decision Trees

Decision Tree is a versatile classification algorithm that splits the data into subsets based on the value of input features. It creates a tree-like model of decisions and their possible consequences, including outcomes, resource costs, and utility.

Below, the hyperparameters to be tuned in Decision Tree can be found:

- **max_depth**: The maximum depth of the tree. It controls the maximum number of levels in the tree.
- **min_samples_split**: The minimum number of samples required to split an internal node.
- **min_samples_leaf**: The minimum number of samples required to be at a leaf node.

The best configuration from the nested cross-validation is:

- **max_depth**: 5
- **min_samples_split**: 2
- **min_samples_leaf**: 1

These are the corresponding performance metrics:

	Accuracy	ROC AUC	F1 Score	Precision	Recall
Value	0.85	0.85	0.86	0.87	0.84

Table 14: Performance Metrics for the Best Decision Tree Model

Figure 40 displays the confusion matrix along with an Age-BMI plot containing the decision boundary generated by the Best Decision Tree Model. In the right plot, blue represents healthy individuals (either true or predicted), while green denotes severe patients (true or predicted).

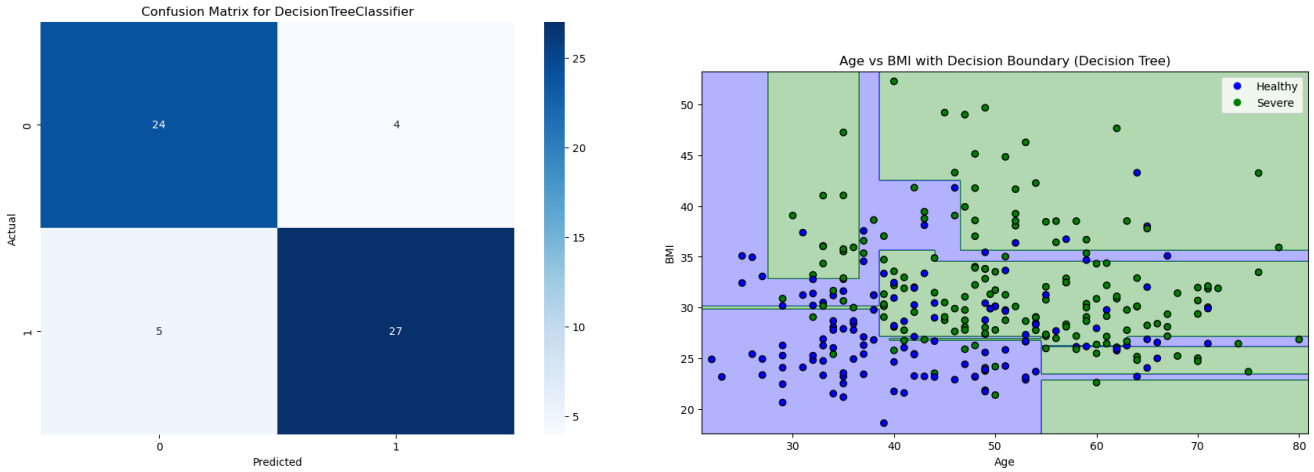

Figure 40: Confusion matrix (left) and annotated plot (Age vs BMI) with decision boundary (right) of the Best Decision Tree Model

In addition, a visualization of the decision tree can be made, illustrated in 41.

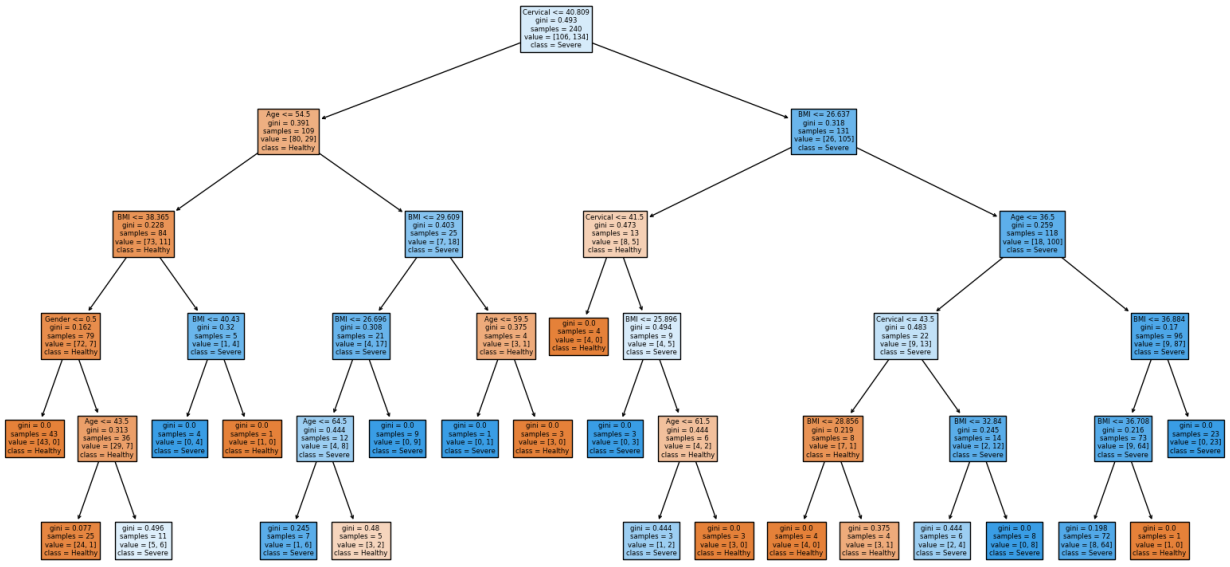

Figure 41: Graphical representation of the decision tree

The best Decision Tree model uses a maximum depth of 5, a minimum of 2 samples to split an internal node, and a minimum of 1 sample at a leaf node, achieving a balanced performance across accuracy, precision, and recall.

Random Forest

Random Forest is an ensemble learning method that constructs multiple decision trees during training and outputs the mode of the classes (classification) or mean prediction (regression) of the individual trees. It improves the predictive accuracy and controls overfitting.
Below, the hyperparameters to be tuned in Random Forest can be found:

- **max_depth**: The maximum depth of the tree. It controls the maximum number of levels in the tree.
- **max_features**: The number of features to consider when looking for the best split.
- **min_samples_split**: The minimum number of samples required to split an internal node.
- **min_samples_leaf**: The minimum number of samples required to be at a leaf node.
- **n_estimators**: The number of trees in the forest.

The best configuration from the nested cross-validation is:

- **max_depth**: 10
- **max_features**: 2
- **min_samples_split**: 5
- **min_samples_leaf**: 1
- **n_estimators**: 50

These are the corresponding performance metrics:

	Accuracy	**ROC AUC**	**F1 Score**	**Precision**	**Recall**
Value	0.88	0.88	0.90	0.88	0.91

Table 15: Performance Metrics for the Best Random Forest Model

Figure 42 displays the confusion matrix along with an Age-BMI plot containing the decision boundary generated by the Best Random Forest Model. In the right plot, blue represents healthy individuals (either true or predicted), while green denotes severe patients (true or predicted).

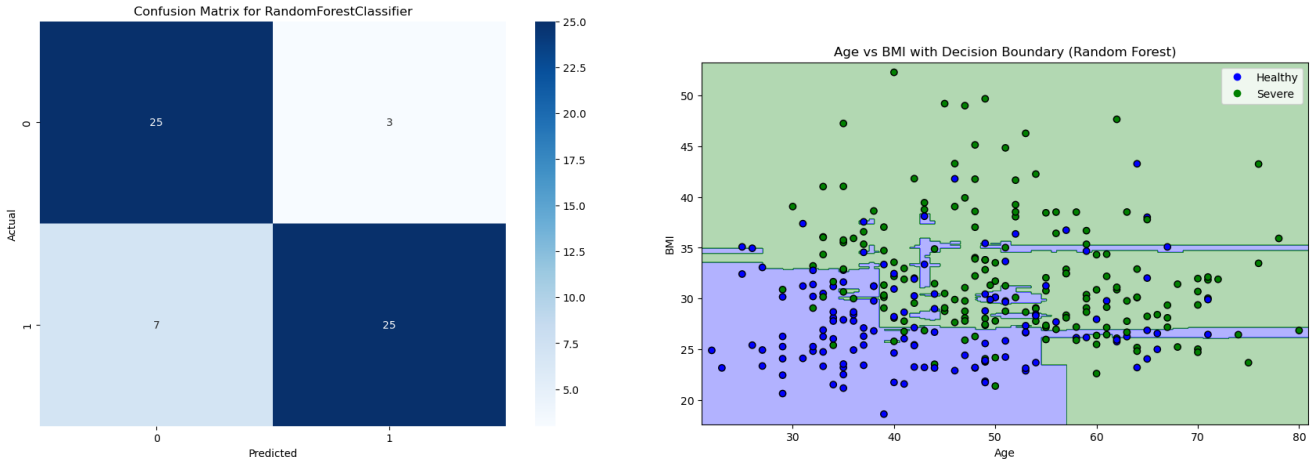

Figure 42: Confusion matrix (left) and annotated plot (Age vs BMI) with decision boundary (right) of the Best Random Forest Model

In addition, feature importance can be evaluated using two approaches: Mean Decrease in Impurity (MDI) and permutation importance. MDI computes feature importance as the mean and standard deviation of impurity decrease across all trees in the model, as shown in the left subplot of Figure 43. While MDI is effective, it has a bias towards features with high cardinality. To address this limitation, permutation feature importance can be used, as it is not affected by feature cardinality and can be computed on a hold-out test set. This provides a more robust measure of feature importance, as depicted in the right subplot of Figure 43.

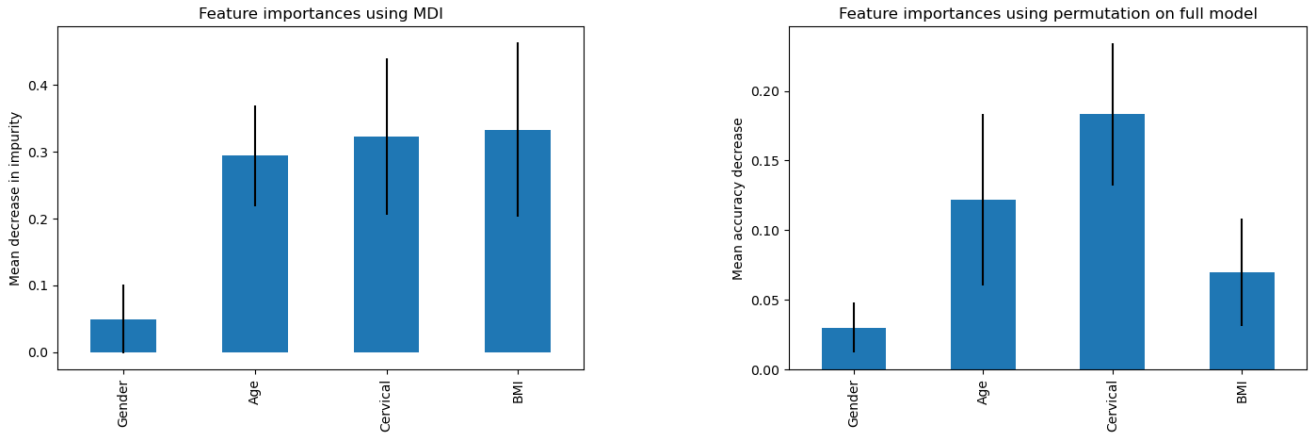

Figure 43: Comparison of feature importance measures: Mean Decrease in Impurity (left) and Permutation Feature Importance (right)

The best Random Forest model uses a maximum depth of 10, 2 features for the best split, a minimum of 5 samples to split an internal node, a minimum of 1 sample at a leaf node, and 50 trees in the forest, achieving a balanced performance across accuracy, precision, and recall.

XGBoost

XGBoost is an optimized distributed gradient boosting library designed to be highly efficient, flexible, and portable. It implements machine learning algorithms under the Gradient Boosting framework.
Below, the hyperparameters to be tuned in XGBoost can be found:

- **learning_rate**: Step size shrinkage used to prevent overfitting.

- **max_depth**: The maximum depth of a tree. It controls the maximum number of levels in the tree.
- **n_estimators**: The number of boosting rounds.
- **subsample**: The fraction of samples to be used for fitting the individual base learners.

The best configuration from the nested cross-validation is:

- **learning_rate**: 0.1
- **max_depth**: 3
- **n_estimators**: 50
- **subsample**: 1.0

These are the corresponding performance metrics:

	Accuracy	ROC AUC	F1 Score	Precision	Recall
Value	0.83	0.82	0.86	0.80	0.94

Table 16: Performance Metrics for the Best XGBoost Model

Figure 44 displays the confusion matrix along with an Age-BMI plot containing the decision boundary generated by the Best XGBoost Model. In the right plot, blue represents healthy individuals (either true or predicted), while green denotes severe patients (true or predicted).

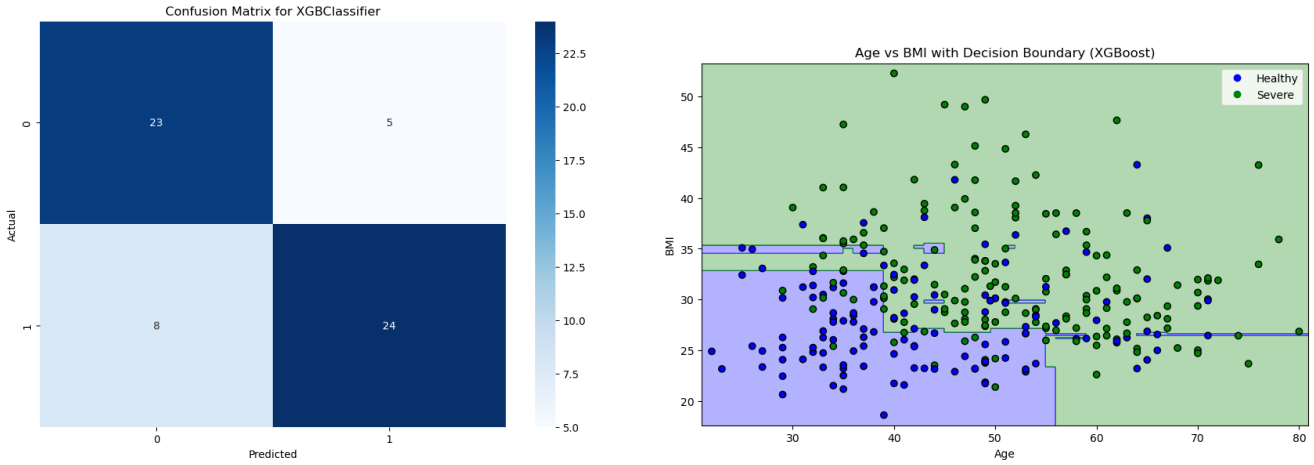

Figure 44: Confusion matrix (left) and annotated plot (Age vs BMI) with decision boundary (right) of the Best XGBoost Model

The best XGBoost model uses a learning rate of 0.1, a maximum depth of 3, 50 boosting rounds, and a subsample fraction of 1.0, achieving a balanced performance across accuracy, precision, and recall.

5.3.3 Generative Models

No implementation of generative models for classification has been carried out in VSCode, as other sections of this work have been prioritized.

Linear Discriminant Analysis (LDA)

Linear Discriminant Analysis (LDA) assumes that the data for each class is drawn from a Gaussian distribution with a common covariance matrix but different means for each class. It models the joint probability $p(X, y)$ and establishes a linear decision boundary by maximizing the separation between classes. LDA often provides similar results to Logistic Regression; however, LDA performs better when the assumption of Gaussian-distributed data with a shared covariance matrix holds approximately. On the other hand, if the Gaussian assumptions are not met, Logistic Regression may outperform LDA.

Quadratic Discriminant Analysis (QDA)

Quadratic Discriminant Analysis (QDA) extends LDA by allowing each class to have its own covariance matrix, resulting in a quadratic decision boundary. This flexibility makes QDA more suitable for datasets where the classes have distinct covariance structures. It is therefore recommended to use QDA instead of LDA if the training data is very large or if the assumption for LDA of a common covariance matrix is not achievable. It serves as a compromise between the non-parametric nature of KNN and the linear approaches of LDA and Logistic Regression. While not as flexible as KNN, QDA can outperform linear methods, especially when there is a sufficient number of training samples.

Naive Bayes (LDA for $p = 1$)

Naive Bayes simplifies LDA by assuming feature independence, hence the term "naive." It calculates class probabilities based on Bayes' theorem and is especially efficient for high-dimensional datasets with categorical or text-based features.

Gaussian Mixture Models (GMM)

Gaussian Mixture Models (GMM) model the data as a mixture of multiple Gaussian distributions, with each Gaussian representing a cluster or class. GMMs are capable of capturing complex data distributions and are useful for unsupervised learning. They can also serve as a probabilistic approach to classification when class labels are available.

5.3.4 Non-parametric models

K-Nearest Neighbours

K-Nearest Neighbors (KNN) is a simple, instance-based learning algorithm that classifies a data point based on how its neighbors are classified. It is a non-parametric method used for classification and regression.
Below, the hyperparameters to be tuned in KNN can be found:

- **metric**: The distance metric to use for the tree ('euclidean', 'manhattan', 'minkowski').
- **n_neighbors**: Number of neighbors to use.
- **weights**: Weight function used in prediction ('uniform', 'distance').

The best configuration from the nested cross-validation is:

- **metric**: `euclidean`
- **n_neighbors**: 5
- **weights**: `distance`

These are the corresponding performance metrics:

	Accuracy	ROC AUC	F1 Score	Precision	Recall
Value	0.83	0.83	0.84	0.87	0.81

Table 17: Performance Metrics for the Best K-Nearest Neighbors Model

Figure 45 displays the confusion matrix along with an Age-BMI plot containing the decision boundary generated by the Best K-Nearest Neighbors Model. In the right plot, blue represents healthy individuals (either true or predicted), while green denotes severe patients (true or predicted).

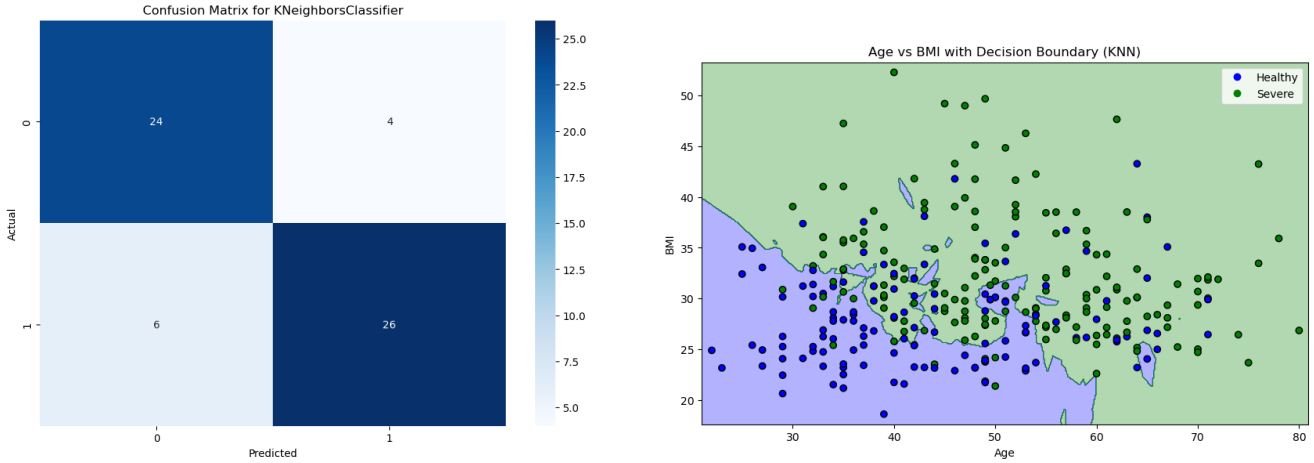

Figure 45: Confusion matrix (left) and annotated plot (Age vs BMI) with decision boundary (right) of the Best K-Nearest Neighbors Model

The best KNN model uses the `euclidean` distance metric, 5 neighbors, and `distance` weights, achieving a balanced performance across accuracy, precision, and recall.

5.3.5 Hybrid Model

Soft Voting Classifier

A Soft Voting Classifier is an ensemble method that combines multiple base classifiers to make a final prediction. In this approach, each model in the ensemble outputs class probabilities, and the final prediction is determined by averaging these probabilities. The class with the highest average probability is chosen as the predicted class. This method can improve classification performance by leveraging the strengths of multiple models, especially when they make different types of errors.

However, in this work, no machine learning models will be trained for the hybrid model due to time constraints.

5.4 ML models for clustering

The following section describes the machine learning models used to cluster the data features (weight, height, age, cervical and BMI), without considering the outcome variable (i.e., AHI). This approach offers insights into the inherent similarities and differences within the data features.

K-means clustering

K-means clustering is an unsupervised machine learning algorithm used to group data into a specified number of clusters based on the similarity of the features. The algorithm assigns each data point to the nearest cluster centroid and iteratively updates the centroids until convergence. The goal is to minimize the within-cluster variance, ensuring that data points within each cluster are as similar as possible.

Before applying the K-means clustering algorithm, it is crucial to scale the data features to ensure optimal performance. In this case, standardization is applied, transforming the features to have a mean of 0 and a standard deviation of 1.

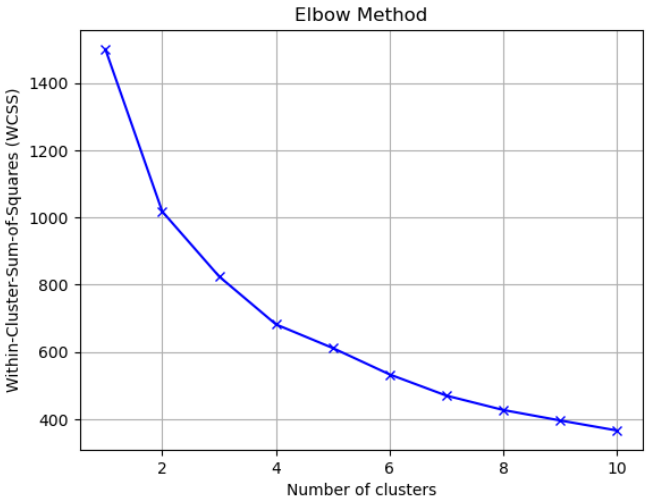

Figure 46: Plot of elbow method used in K-means clustering

To determine the optimal number of clusters, the elbow method is commonly used, as visualized in figure 46. This method involves running the K-means algorithm for different values of K (i.e. the number of clusters) and plotting the within-cluster sum of squares (WCSS) against increasing values of K. The "elbow" point on the graph, where the rate of decrease in WCSS slows down, indicates the optimal number of clusters. In this case, the elbow point corresponds to an amount of 5 clusters. This point represents a balance between a good fit and the complexity of the model.

Cluster	Weight	Age	Height	Cervical	BMI
0	105.35	41.95	176.16	43.76	34.07
1	68.05	46.53	162.85	34.98	25.75
2	85.50	62.33	168.13	42.25	30.41
3	85.89	40.69	177.43	40.34	27.36
4	136.40	47.35	176.45	46.60	43.85

Table 18: Cluster Data Table

After convergence, the cluster centroids will not change anymore. The value of each of these centroids along each feature axis is visualized in table 18. Based on the observations of these cluster centers, following characteristics can be described for each cluster:

- Cluster 0: This group consists of younger individuals who are obese (BMI 30) [41], with higher weight and BMI. They tend to have a larger cervical circumference compared to others, but their height is average.
- Cluster 1: Individuals in this cluster are at the edge of being overweight, with BMI values just above healhty (BMI 25). They are in their 40s, with a smaller stature and lower cervical circumference.
- Cluster 2: This group includes older, obese individuals. They have higher BMI values, larger cervical circumference, and average height.
- Cluster 3: These individuals are overweight with BMI values in the higher end of the normal range, are in their early 40s, and are generally taller. They also have a moderate cervical circumference, suggesting a balanced health profile relative to their height.
- Cluster 4: Characterized by severely obese individuals (BMI $\geq$ 40) who are adults, typically older. They have a very large cervical circumference and high body mass, suggesting a higher risk of obesity-related health issues. Their height is average compared to the other clusters.

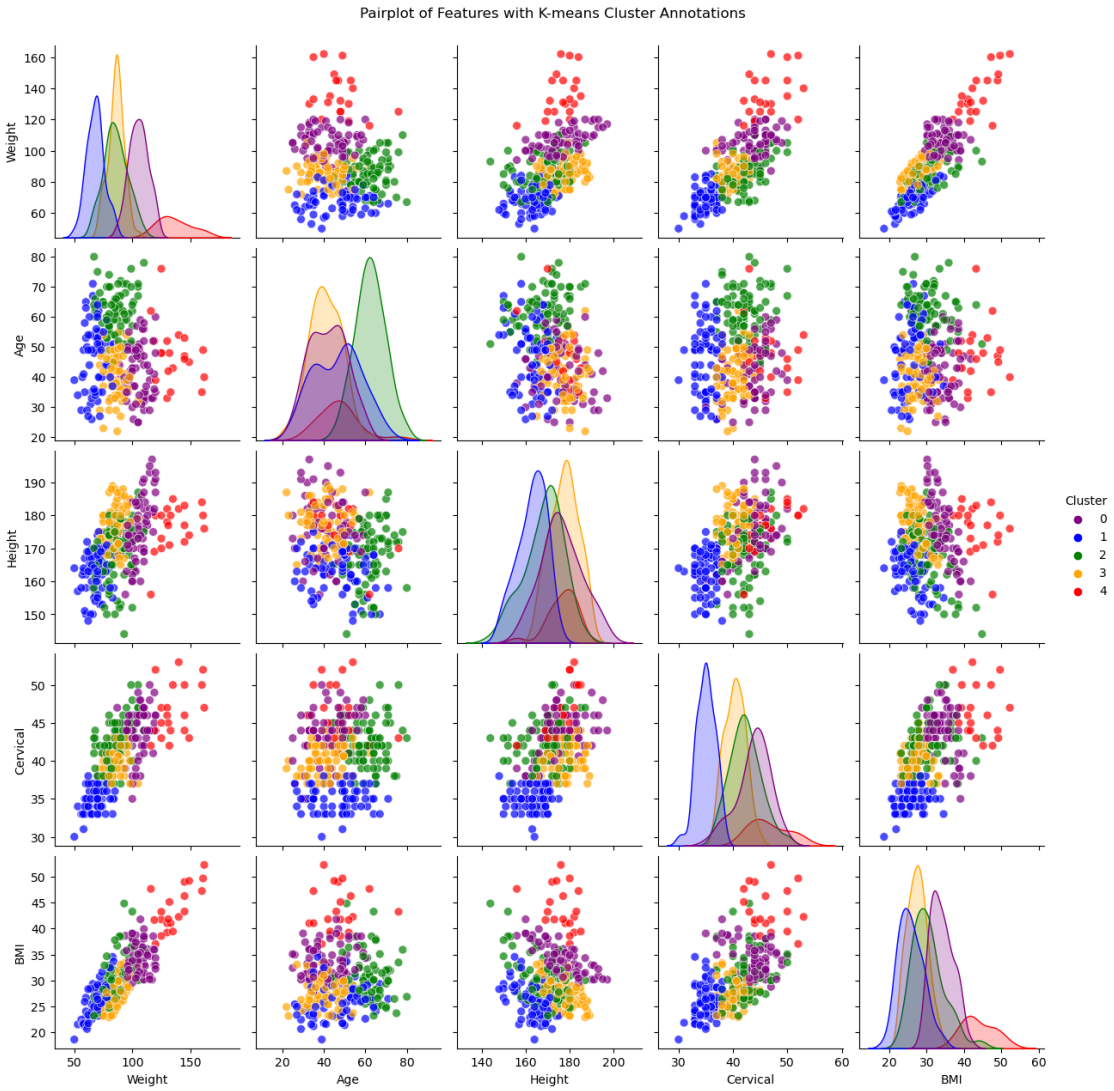

Figure 47: Pairplot of the annotated features after K-means clustering

Next, a pairplot is created to visualize the relationships between all five features, with each data point colored according to the cluster it belongs to. This provides a clear depiction of how the clusters are distributed across the feature space. The resulting visualization is shown in Figure 47.

Hierarchical clustering

Another way to perform clustering is hierarchical clustering. This is a method of cluster analysis that builds a hierarchy of clusters. It can be divided into two main approaches:

- **Agglomerative Clustering**: This is a bottom-up approach where each data point starts as its own cluster, and pairs of clusters are merged iteratively based on a chosen similarity metric. The process continues until all points are grouped into a single cluster.
- **Divisive Clustering**: This is a top-down approach where all data points start in a single cluster, and the most dissimilar points are iteratively separated into new clusters until each point forms its own cluster.

The result of hierarchical clustering can be visualized using a *dendrogram*, which is a tree-like diagram that illustrates the arrangement and distances between clusters at each step of the process. This is visualized in figure 48.

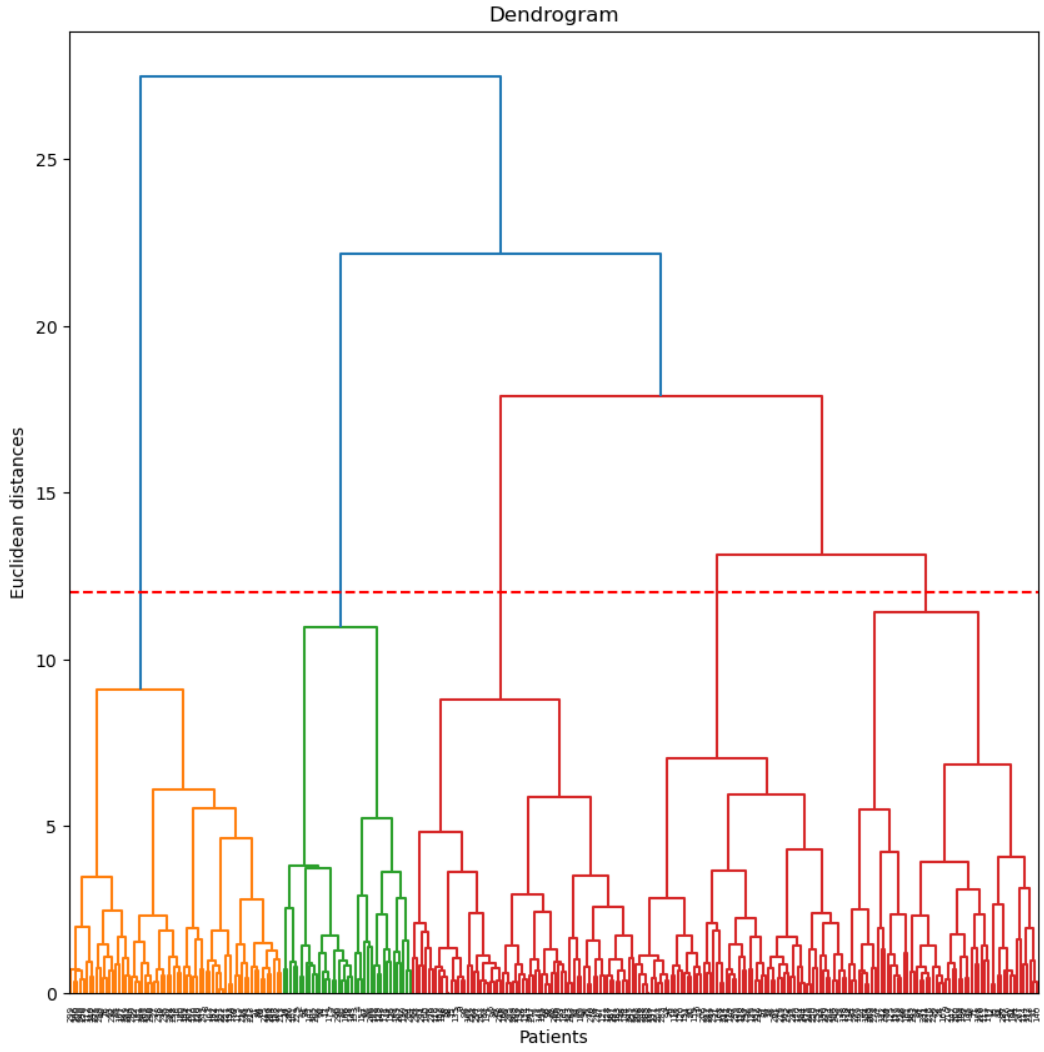

Figure 48: Dendrogram obtained by hierarchical clustering

The dendrogram offers an intuitive visualization of the relationships between clusters, allowing for a clear understanding of how clusters merge or split at various stages. Additionally, it serves as an alternative to the elbow method for determining the optimal number of clusters. By examining the point where the clusters merge most significantly, the desired level of granularity can be chosen. In this case, as indicated by the red dotted line in Figure 48, a cluster count of 5 is identified as the optimal choice, confirming the choice based on the elbow method.

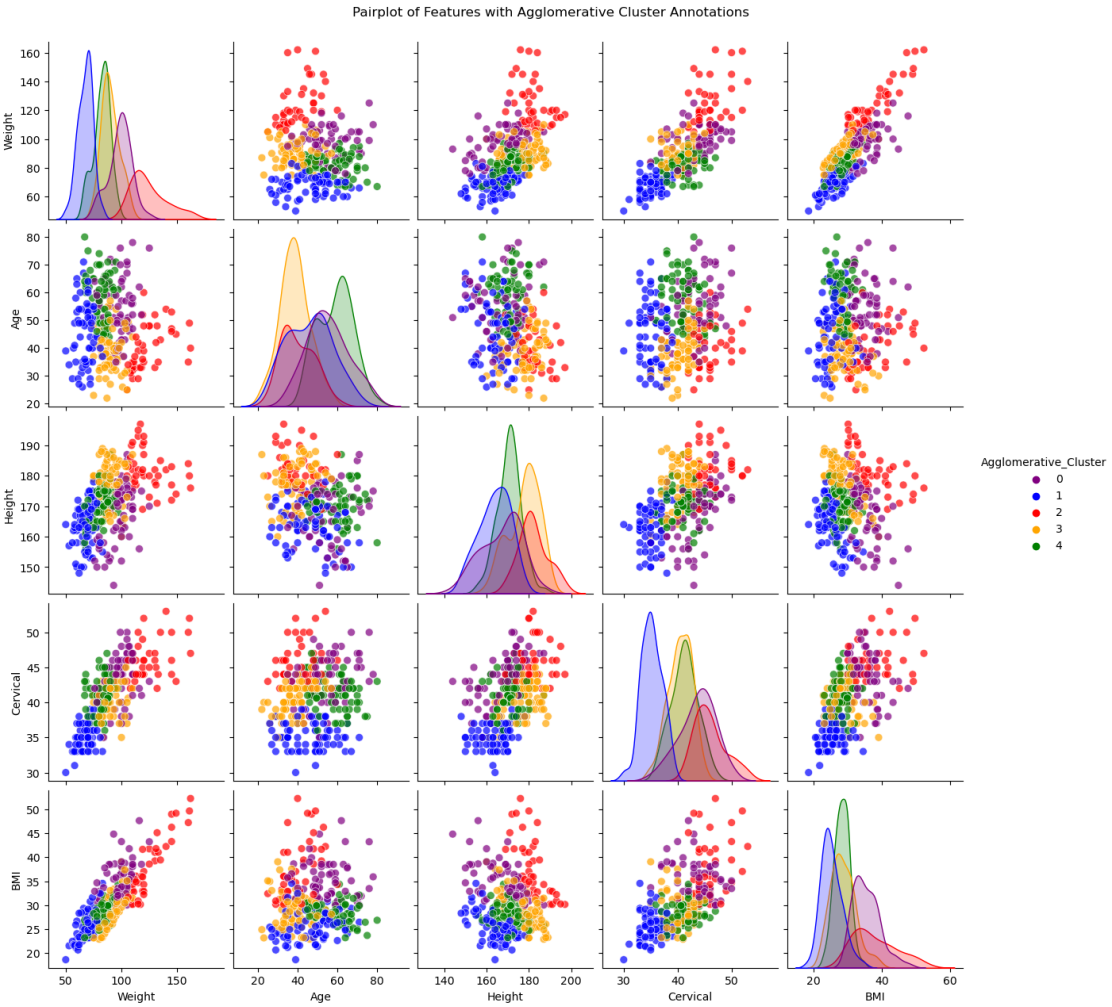

Figure 49: Pairplot of the annotated features after agglomerative clustering

After agglomerative clustering, again a pairplot can be made which is visualized in figure 49.

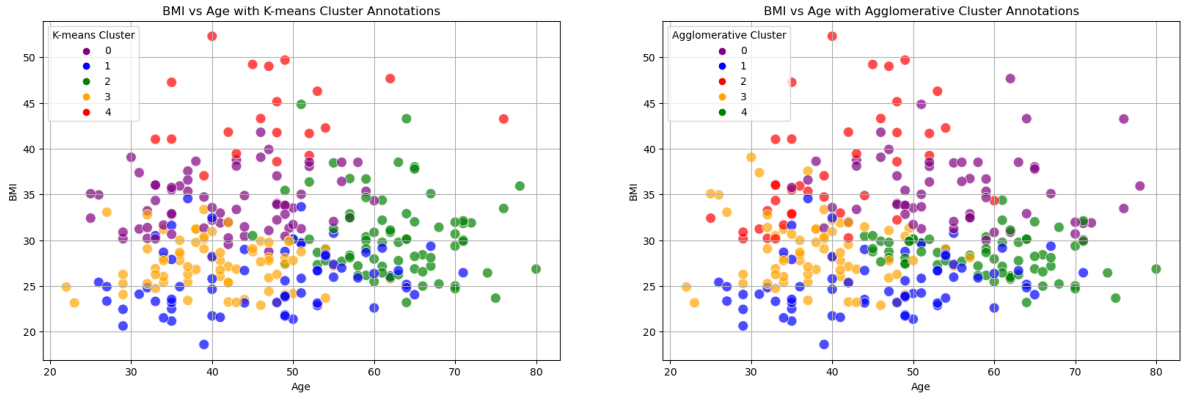

Figure 50: Comparison between K-means and Agglomerative clustering on the Age vs BMI plot

To facilitate a more precise comparison between the K-means and agglomerative clustering algorithms, two features—Age and BMI—are plotted against each other, as shown in Figure 50. The cluster colors have been carefully mapped to ensure a clear and easy comparison between the two clustering techniques.

6 Discussion

6.1 ML models for Regression

A comparison of the different regression models can be made based on their RMSE values. As shown in table 19, the Support Vector Machine model achieves the lowest RMSE, indicating its superior performance among the models evaluated.

Model	**RMSE**
Multiple Linear Regression	15.78
Linear Regression with Interaction	16.25
Lasso Linear Regression	15.76
Decision Tree Regression	16.75
Random Forest Regression	15.31
XGBoost Regression	14.79
SVM Regression	**14.67**
Multi-Layer Perceptron Regression	15.46

Table 19: Comparison of RMSE values across the trained Regression models

The regression models in this study were designed to predict the Apnea Hypopnea Index (AHI), offering a continuous measure of sleep apnea severity. Among the implemented models, ensemble methods such as Random Forest and XGBoost outperformed simpler linear models, achieving lower RMSE and higher overall predictive accuracy. These findings align with the capability of tree-based methods to capture non-linear relationships and interactions between features. For example, the importance of clinical variables such as BMI and cervical circumference in predicting AHI was effectively leveraged by these models.

However, linear regression models, including Lasso, also showed promising performance, especially when the feature space was carefully engineered. Their relative simplicity and interpretability make them valuable in clinical contexts where transparency is crucial. Nonetheless, their inability to capture complex relationships without extensive feature transformations remains a limitation.

One key insight from the regression analysis was the impact of data preprocessing, particularly the imputation of missing values. The decision to use median imputation and feature scaling significantly improved model robustness, as evidenced by consistent performance across validation sets. This emphasizes the importance of rigorous preprocessing in predictive modeling for clinical datasets, where data irregularities are common.

6.2 ML models for Classification

A similar comparison can be made for the classification models using the F1 score as the evaluation metric. As shown in table 20, the Random Forest model achieves the highest F1 score, highlighting its superior performance among the classification models.

Model	**F1 Score**
Logistic Regression	0.86
Multi-Layer Perceptron	0.86
Support Vector Machine	0.87
Decision Tree	0.86
Random Forest	**0.90**
XGBoost	0.86
K-Nearest Neighbors	0.84

Table 20: Comparison of F1 scores across the trained Classification models

For the classification task, models aimed to categorize patients as "healthy" (AHI < 5) or "severe" (AHI ≥ 30). Among the classification models, tree-based methods such as Random Forest and XGBoost again demonstrated superior performance, achieving high F1 scores. Additionally, the use of nested cross-validation ensured reliable evaluation by minimizing data leakage and providing an unbiased estimate of model performance.

Despite these successes, the classification models have limitations. The reliance on synthetic oversampling methods like SMOTENC, while effective, introduces potential risks of overfitting. Future work should explore advanced imbalance-handling techniques, such as cost-sensitive learning or adaptive sampling, to address this issue. Moreover, while predictive performance was prioritized, the interpretability of these models requires further emphasis. For example, SHAP (SHapley Additive exPlanations) values could be used to quantify feature contributions and align predictions with clinical reasoning. [42]

6.3 Ablation study

Ablation studies are a critical method in ML and DL projects used to evaluate the contribution of specific components within an AI system. By systematically removing or altering components of a model and assessing the resulting performance, researchers can better understand the importance of each part of the system. The term *ablation* is borrowed from biology, where it originally referred to experiments involving the removal of brain tissue to study behavioral changes in animals. [43]

In the context of machine learning, ablation involves removing or disabling specific elements, such as network layers in deep neural network architectures or feature layers, and then measuring the impact on performance. This iterative process helps identify which components are essential and how they contribute to the overall system's success. While ablation studies are often applied to neural architectures, they are not limited to them and can also involve feature ablation to assess feature importance.

Procedure for Ablation Studies

1. **Baseline Performance**: Train the model with all components intact and measure its performance using relevant metrics.
2. **Component Removal**: Systematically remove or disable specific components of the model.
3. **Performance Evaluation**: Reassess the model's performance after each modification.
4. **Interpretation of Results**: Analyze the results to determine the importance of the removed components. For instance, if removing a component significantly degrades performance, it indicates that the component is critical to the model's functionality.

Since removing certain components is particularly suited for architectures like convolutional neural networks, an implementation of an ablation study will be included in the Deep Learning Report.

7 Limitations

While this report effectively applies machine learning techniques to clinical data, several areas for improvement remain. One limitation is the lack of justification for the choice of algorithms. Future iterations should explicitly explain why specific models were selected based on the dataset's characteristics and problem requirements.

The evaluation metrics primarily focus on RMSE (for regression) or F1 Score (for classification), which may not fully capture performance, especially for imbalanced datasets. Incorporating metrics like precision, recall, and AUC-ROC would provide a more comprehensive assessment.

Model interpretability is another area for enhancement. Including an analysis of feature importance for every model and its alignment with domain knowledge would improve the report's practical relevance, especially in clinical settings where interpretability is critical. Similarly, the ablation study is underexplored; an implementation of a deeper analysis of how feature removal impacts performance would offer valuable insights into model robustness.

Although the secondary dataset is included for exploratory analysis, its potential remains underutilized. Applying some modeling techniques or unsupervised methods could validate the generalizability of the approaches and provide additional insights.

8 Conclusion

8.1 Conclusion related to the project

This report demonstrates the potential of machine learning (ML) models to predict and classify the Apnea Hypopnea Index (AHI), a critical metric for diagnosing obstructive sleep apnea (OSA). Through a structured pipeline, the report showcases robust data preprocessing, exploratory data analysis, and the implementation of various ML techniques, including regression, classification, and clustering. By employing advanced methodologies such as nested cross-validation and handling data imbalances with SMOTENC, the analysis ensures the reliability and generalizability of the models. These efforts highlight the value of leveraging ML to address challenges in clinical data analysis, such as missing values and complex variable relationships.

The findings of this study underscore the practical relevance of ML in clinical practice. Predicting AHI using readily available clinical parameters can complement traditional diagnostic methods like polysomnography, which are time-consuming and resource-intensive. ML models offer a scalable and cost-effective solution for identifying at-risk patients, enabling earlier interventions and reducing the burden on healthcare systems. Moreover, their ability to integrate diverse data sources and identify complex patterns can pave the way for personalized treatment strategies, enhancing patient outcomes.

However, for ML models to be effectively adopted in clinical settings, interpretability and ethical considerations must remain central. Understanding feature contributions and ensuring that models align with medical knowledge are crucial for building trust among healthcare professionals. Additionally, addressing potential biases in datasets and maintaining transparency in model development are essential to ensure equitable healthcare delivery.

In conclusion, this report not only highlights the technical capabilities of ML for predicting and classifying AHI but also emphasizes its transformative potential in clinical practice. By bridging gaps in current diagnostic approaches, ML has the potential to improve the efficiency and accuracy of OSA diagnosis and management, ultimately enhancing patient care.

8.2 Conclusion of what I learned

Data preprocessing takes significantly more time compared to training and selecting models. This is particularly true when using off-the-shelf functions from libraries like scikit-learn, such as nested cross-validation, which simplify some processes in the training of these models but still require careful handling of data in the data preprocessing stage.

Additionally, I had the opportunity to use and train models that I had previously only studied theoretically, but never implemented in practice. This was also my first experience working with a nested cross-validation loop. Initially, it was challenging to grasp, but I have since become much more comfortable with it.

While it's tempting to train many models, it's crucial to understand the underlying assumptions behind each approach. For example, understanding the assumptions of statistical tests such as normality tests, the differences between Pearson and Spearman correlations, the proper imputation techniques etc. is essential. Just because you can apply a method doesn't mean it's always the right choice. Critical thinking and a clear understanding of the methods are necessary for making sound decisions.

Achieving good results and publishing them is relatively easy, but the true reliability of these results is often not questioned. How can we be sure that the results aren't simply due to overfitting? There must be a high level of trust in the field, with everyone adhering to established practices for reporting results in a trustworthy and accurate manner.

Data is ubiquitous and holds immense power when used wisely. This is especially true in medical and hospital settings, where vast amounts of data are often collected but not fully utilized. Answers to many medical questions may already lie within these datasets; however, the challenge is identifying and uncovering the meaningful patterns within them.

Furthermore, I gained valuable experience working with GitHub repositories, which proved to be an indispensable tool for managing and tracking my code. For instance, after making some changes that did not produce the desired outcome, I was able to quickly revert to a previous version I had pushed to my local repository, saving significant time and effort. Additionally, I completed all programming using VSCode, which was a relatively new environment for me. Through this experience, I became proficient in using conda environments and executing basic terminal commands. I also developed a strong appreciation for VSCode's features, such as its syntax highlighting in different colors, automatic underlining of unused commands, and seamless integration with the Copilot tool for collaboration and code generation. These features not only streamlined my workflow but also enhanced my coding efficiency and productivity.

Finally, the use of large language models (LLMs), such as GPT (in ChatGPT) and GitHub Copilot, has been invaluable in the creation of this report. I've learned how to use them effectively to accelerate my work and improve its quality. For instance, in VSCode, I used Copilot to generate specific plots with custom color labels, quickly identify hyperparameters for ML models, execute statistical tests that I knew how to perform in SPSS but not in Python etc. Additionally, these models played a key role in improving the layout of this report on Overleaf. I used ChatGPT to construct complex tables or figures that required specific syntax, saving time that would otherwise be spent searching online, as well as to enhance the coherence of the text.

Without these models, a substantial amount of time would have been spent searching for the correct syntax or solutions online, without learning much in the process and about the fundamentals of the ML pipeline. This would have likely led to a less comprehensive report, especially given the time constraints. As a result, I have come to appreciate the value of AI tools and have learned how to use them responsibly to enhance my productivity and the quality of my work.

9 AI disclaimer

In order to make the development of this report more efficient, ChatGPT-4o is used to generate a coherent text based on provided bullet points that were self-written based on multiple references. In addition, GitHub Copilot is used to generate code based on different prompts. It is important to indicate that the output of these generative AI's is handled in a responsible way, meaning that the correctness is always checked.

A Developing Machine Learning models for breast cancer case study

A similar approach is now applied to analyze a second dataset related to breast cancer recurrence prediction. This analysis serves to demonstrate the ability to perform dataset description and exploratory data analysis (EDA). A notable distinction from the previous dataset is the absence of continuous variables (except for age) and the presence of multiple nominal and categorical variables. It is important to note that no model training will be conducted on this dataset.

A.1 Problem formulation

Breast cancer remains a global health challenge, with 2.3 million women worldwide diagnosed and 670,000 deaths recorded in 2022 alone. Of all breast cancer diagnoses, 99% affect women and 0.5-1% men, making the female gender the strongest risk factor for developing breast cancer. [45] Early detection of this disease is the most effective way to reduce mortality. Therefore, diagnostic methods are developed that distinguish between "benign" (non-cancerous) and "malignant" (cancerous) breast tumors without the need of using a surgical biopsy. On the other hand, prognostic predictions are used for focusing on the likelihood of recurrence after surgical removal of the cancer. [46]

A.2 Machine Learning approach

The diagnosis and prognosis of breast cancer remains a great challenge for the medical team. That is why a lot of effort is put into the implementation of machine learning and data mining methods that aid in the detection and prediction of breast cancer. Data mining refers to the discipline in which large volumes of data are analyzed to support decision-making processes. [47]

In this work, the emphasis is put on breast cancer recurrence, as this remains a critical concern in oncology. The use of robust predictive models is key to identify patterns and risk factors that are associated with events of recurrence. To achieve this, comprehensive analysis is made using a dataset that encompasses different attributes related to breast cancer patients. These features range from demographic information to clinical characteristics. The main focus is to investigate the impact of these different attributes on the recurrence of breast cancer.

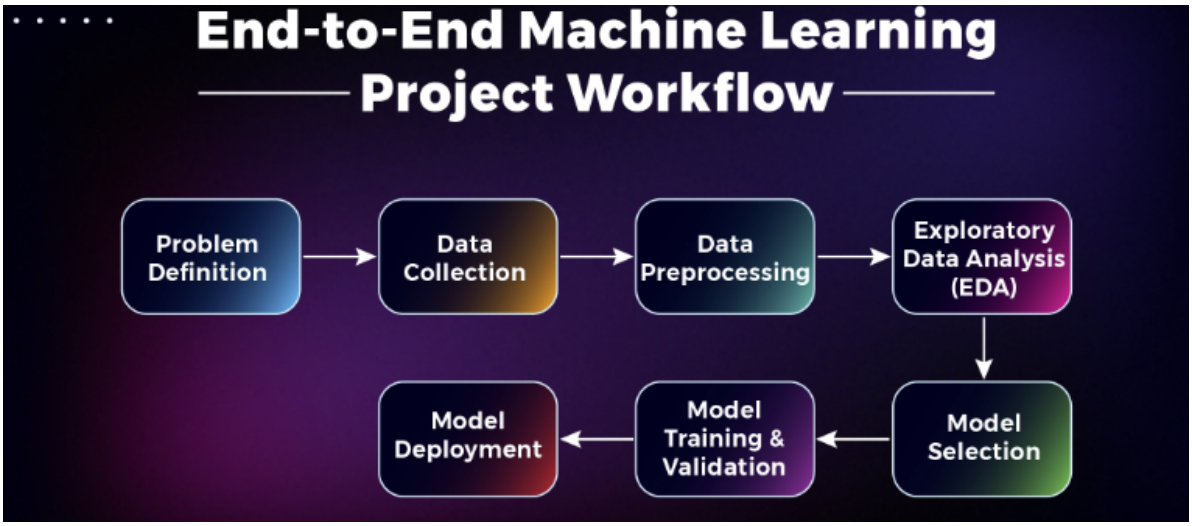

Figure 51: Machine Learning pipeline [44]

The steps that are taken on this dataset are the following, as visualized on figure 51:

- **Problem definition**: Success of a solving a data-oriented problem is clearly defining the problem, such that it aligns with the business goals - in this case, the goal of predicting breast cancer recurrence.
- **Data Collection**: Data collection, a time-intensive step, involves gathering and cleaning raw data needed for model training. This data may come from sources like databases and must be relevant and representative of the problem.
- **Data description**: An extensive dataset description is essential to know the exact meaning of the obtained features and target variable obtained upon data collection.
- **Dataset preprocessing/cleaning**: Data preprocessing is essential for ML success, involving tasks like handling missing values, removing duplicates, and normalizing data. Clean, well-prepared data improves model accuracy.
- **Exploratory Data Analysis (EDA)**: EDA uses visual tools like histograms and scatter plots to reveal data patterns, relationships, and anomalies. This step is important to obtain an intuitive feeling on which relations are present amongst the data features.
- **ML model selection**: Model selection entails choosing the best algorithm based on problem type, data, and the balance between interpretability and accuracy. By comparing different models - what will be performed in this report - the most optimal model for this specific problem will be obtained.
- **Model training and validation**: Model training uses prepared data to teach the chosen algorithm the relationship between inputs and the target variable, with hyperparameter tuning to optimize performance. Validation tests the model on unseen data, refining it before final evaluation.
- **Model deployment**: Model deployment integrates the model into real-life, ensuring efficient, scalable predictions. Post-deployment, models require real-time monitoring and maintenance to manage data drift and maintain accuracy, ensuring long-term effectiveness.

The application of different machine learning models, together with the evaluation of these models with an appropriate discussion and conclusion will not be conducted on this breast cancer dataset. A complete machine learning pipeline will be executed for the OSA use case. The reason for this is that the breast cancer data would not enable all types of ML models to be trained. The breast cancer dataset is solely chosen to serve as proof for the ability to perform an extensive dataset description and corresponding EDA.

B Dataset description for breast cancer case study

B.1 Data access

The dataset that is used to predict the recurrence of breast cancer is available on the UCI Machine Learning Repository. This Breast Cancer domain was obtained from the University Medical Centre, Institute of Oncology, Ljubljana, Yugoslavia. Thanks go to M. Zwitter and M. Soklic for providing the data. [48]

B.2 Data preparation

	age	menopause	tumor-size	inv-nodes	node-caps	deg-malig	breast	breast-quad	irradiat	Class
0	30-39	premeno	30-34	0-2	no	3	left	left_low	no	no-recurrence-events
1	40-49	premeno	20-24	0-2	no	2	right	right_up	no	no-recurrence-events
2	40-49	premeno	20-24	0-2	no	2	left	left_low	no	no-recurrence-events
3	60-69	ge40	15-19	0-2	no	2	right	left_up	no	no-recurrence-events
4	40-49	premeno	0-4	0-2	no	2	right	right_low	no	no-recurrence-events
...	...	...	...	...	...	...	...	...	...	...
281	30-39	premeno	30-34	0-2	no	2	left	left_up	no	recurrence-events
282	30-39	premeno	20-24	0-2	no	3	left	left_up	yes	recurrence-events
283	60-69	ge40	20-24	0-2	no	1	right	left_up	no	recurrence-events
284	40-49	ge40	30-34	5-Mar	no	3	left	left_low	no	recurrence-events
285	50-59	ge40	30-34	5-Mar	no	3	left	left_low	no	recurrence-events

Figure 52: Table of original imported dataset

The dataset was first fetched from the UCI repository using VSCode, displayed in figure 52. The original breast cancer dataset contains 286 instances from which 201 have the label 'no-recurrence-events' (counting for 70.3%) and 85 instances have the label 'recurrence-events' (counting for 29.7%), as illustrated in image 53.

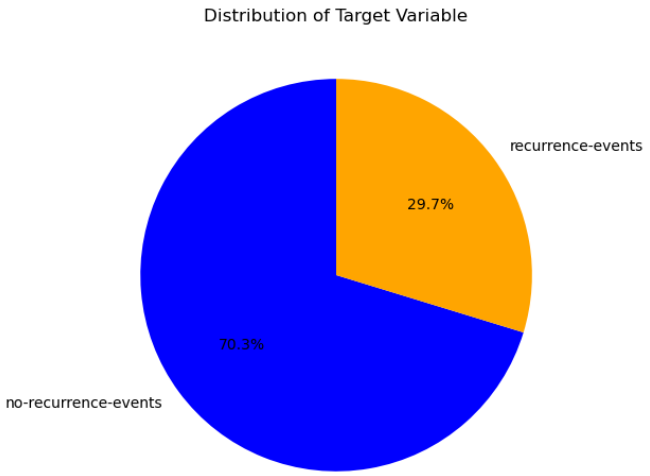

Figure 53: Distribution of the target variable, with 201 (70.3%) classified as recurring and 85 (29.7%) as non-recurring

The instances are described by 9 attributes and have as column names: age, menopause, tumor-size, inv-nodes, node-caps, deg-malig, breast, breast-quad and irradiat. The target variable has the column name 'Class'. In what follows, an extensive description of the variables is given as well as the adaptations that have been executed in order to be able to be used by the prediction models.

B.3 Precise dataset description

B.3.1 Target variable

Class: The output variable has two string values: no-recurrence-events and recurrence-events. They indicate whether or not there is a recurrence of breast cancer in the patient within a time period of 5 years after the tumor has been removed surgically. An encoding of the values has been done to map no-recurrence-events to 0 and recurrence-events to 1. In addition, the column name of this target variable has been changed to cancer_recurrence.

B.3.2 Nominal features

menopause: This feature corresponds to the menopausal status, characterized as premeno (= premenopausal), lt40 (= postmenopausal with an age lower than 40 years) or ge40 (= postmenopausal with an age greater or equal to 40 years). A new variable has been made named post_menopause that can take on the value of 0 when referring to premenopausal and a value of 1 when referring to postmenopausal. Therefore, the instances lt40 and ge40 are grouped together and mapped to a value of 1. This step is performed in order to simplify the interpretation of the variables.

node-caps: This variable refers to the presence (yes) or absence (no) of cancer cells passing through the lymph node (LN) capsules. A mapping has been performed to create a nominal feature, where 'no' is mapped to 0 and 'yes' to 1. In addition, the variable has been renamed as cancer_LN_capsule.

breast: In this column the affected breast can be identified as either left or right. Also here a mapping has been done to link the string 'left' to 1 and 'right' to 0, where the column name has also been changed to left_breast.

breast-quad: This feature refers to location (i.e. the quadrant) of the tumor in the affected breast. The possible values are left_low, left_up, right_low, right_up, central. In order to make this a nominal variable, one-hot encoding is used. In this case, five new columns have been made containing either 0 or 1. These new columns are named affected_left_low, affected_left_up, affected_right_low, affected_right_up and affected_central. The value 1 is present in e.g. the column affected_left_low if their corresponding value left_low was present in the original breast-quad column. One-hot encoding has the advantage to avoid the misinterpretation of categorical data as having some ordinal significance. However, it has the disadvantage of increasing the dimensionality of your input data. [49]

irradiat: This variable refers to whether the patient underwent irradiation during their treatment. Again a mapping is made where 'no' is mapped to 0 and 'yes' to 1. Finally, the column name has been changed to radiotherapy.

B.3.3 Categorical features

age: The patient's age has been categorized into different ranges (10–19, 20–29, 30–39, 40–49, 50–59, 60–69, 70–79, 80–89, 90–99). To create a variable that could be used as input for prediction models, a mapping has been performed to the corresponding numbers 0 up till 8. In addition, the column has been renamed to age_patient.

tumor-size: This feature describes the size of the tumor, expressed in mm and categorized into specific ranges (0–4, 5–9, 10–14, 15–19, 20–24, 25–29, 30–34, 34–39, 40–44, 45–49, 50–54, 55–59 mm). Therefore, a mapping was done to the corresponding numbers 0 up till 11. Finally, the column was given the name tumor_size.

inv-nodes: This variable refers to the number of involved lymph nodes that have been invaded by cancer cells. The values of this feature are grouped according to 0–2, 3–5, 6–8, 9–11, 12–14, 15–17, 18–20, 21–23, 24–26, 27–29, 30–32, 33–35, 36–39. Again a mapping is performed to the numbers 0 up till 12. At the end, the column was named amount_cancer_LN.

deg-malig: The final feature refers to the histological degree of malignancy of the tumor, graded as 1, 2 or 3. Here, only the column name has been changed to tumor_grade.

B.4 Data cleaning

B.4.1 Dealing with data inconsistency

By querying the unique values in each column of the pandas dataframe, some data inconsistency was observed. These were corrected before the new column (name) was produced, as described in the previous part.

For the column tumor-size, the wrong values '14-Oct' and '9-May' were present. These correspond respectively to to ranges '10-14' and '5-9'. Therefore, the correct values were created before the new mapping to the numbers 0 up till 11 was performed. For the feature inv-nodes, a similar inconsistency occurred, where '8-Jun', '11-Sep', '5-Mar' and '14-Dec' needed to be transformed to the corresponding ranges '6-8', '9-11', '3-5' and '12-14'. This was done before the mapping to the aforementioned numbers was executed.

B.4.2 Dealing with missing values

For the feature node-caps, there were 8 missing values present, indicated by a nan (not a number) value. For the column named breast-quad, there was one value missing. Instances in which a missing value was present were discarded from the subsequent analysis. Therefore, the number of instances left holds 277 from which 196 are labeled with 'no cancer recurrence' and 81 with 'cancer recurrence'.
Therefore at the end of the data cleaning and adaptation steps, the following table 54 represents the data that will be used in further analysis.

	cancer_recurrence	radiotherapy	left_breast	cancer_LN_capsule	post_menopause	tumor_grade	amount_cancer_LN	tumor_size	age_patient	affected_central	affected_left_low	affected_left_up	affected_right_low	affected_right_up
0	0	0	1	0	0	3	0	6	2	0	1	0	0	0
1	0	0	0	0	0	2	0	4	3	0	0	0	0	1
2	0	0	1	0	0	2	0	4	3	0	1	0	0	0
3	0	0	0	0	1	2	0	3	5	0	0	1	0	0
4	0	0	0	0	0	2	0	0	3	0	0	0	1	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
272	1	0	1	0	0	2	0	6	2	0	0	1	0	0
273	1	1	1	0	0	3	0	4	2	0	0	1	0	0
274	1	0	0	0	1	1	0	4	5	0	0	1	0	0
275	1	0	1	0	1	3	1	6	3	0	1	0	0	0
276	1	0	1	0	1	3	1	6	4	0	1	0	0	0

Figure 54: Table of dataset after data preprocessing steps

C Exploratory Data Analysis for breast cancer case study

C.1 Target variable and attributes

C.1.1 Cancer recurrence

In order to find out whether the dataset is balanced, the distribution of the target variable 'cancer_recurrence' is plotted, as visualized in figure 55. It is clear that the dataset is imbalanced with respect to the outcome variable. Therefore, it will be important to take this into account when training certain prediction models. If not, the model would learn to classify the majority class (in this case: no cancer recurrence) better than the minority class (cancer recurrence).

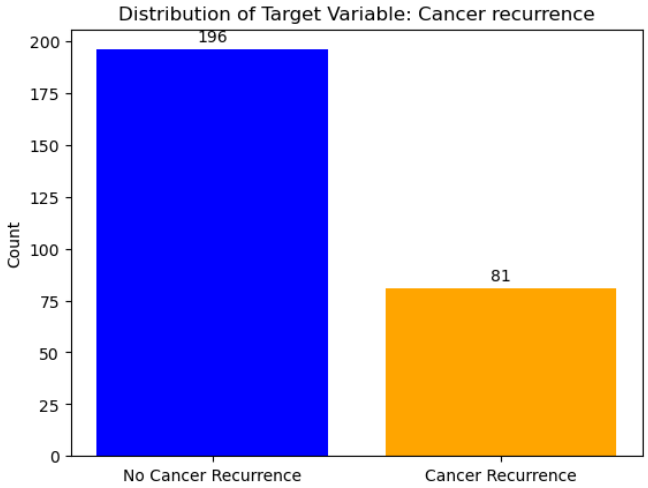

Figure 55: Distribution of target variable: cancer recurrence

C.1.2 Age

As seen on the histograms in figure 56, it is clear that the age distribution follows a Gaussian function for both the group of no recurrence (with a mean of 3.70 and a standard deviation of 1) and the group with cancer recurrence (with a mean of 3.5 and a standard deviation of 1.03). In general, the risk of developing cancer increases with age. [50]

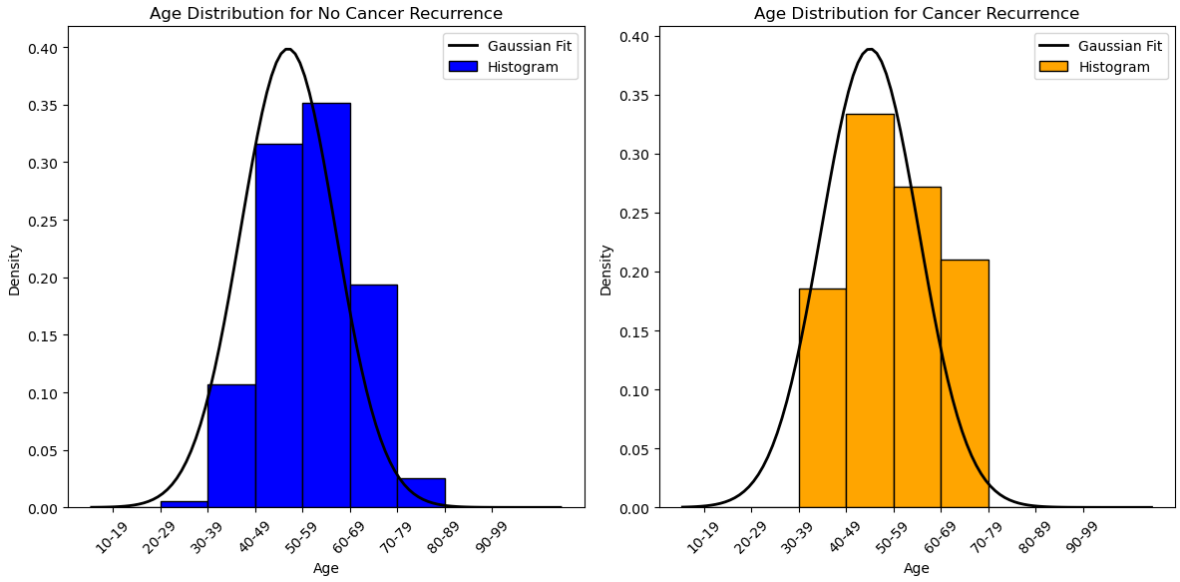

Figure 56: Histograms of age distribution for no cancer recurrence (left) and cancer recurrence (right)

C.1.3 Menopause

As seen on the pie charts in figure 57, there are more people with a cancer recurrence before experiencing their menopause than after. This seems to be in contrast with literature, which states that postmenopausal individuals have higher levels of the hormone oestrogen circulating in their body and therefore, have around twice the risk of developing breast cancer. [51] According to research, postmenopausal women have a lower risk of developing breast cancer compared to premenopausal women of the same age and childbearing pattern. Therefore, women that experience menopause at 55 years rather than 45 years have approximately 30% higher risk of developing breast cancer. [52] In addition, according to the National Institute on Aging, most women begin the menopausal transition between the ages 45-55 years. [53]

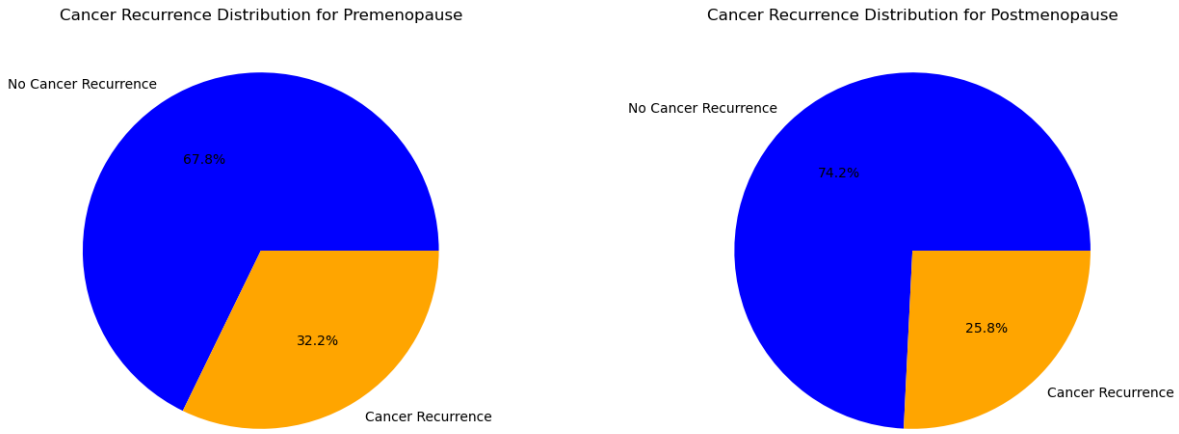

Figure 57: Pie charts with distribution of cancer recurrence for premenopause (left) and postmenopause (right)

C.1.4 Tumor size

According to the pie charts visible in figure 58, more patients with a larger tumor size experienced cancer recurrence in a later stage. This seems quite intuitive as a larger tumor indicates a more progressive growth of cancerous cells in the breast and possibly a higher risk of resident tumor tissue after surgical removal of the breast tumor.

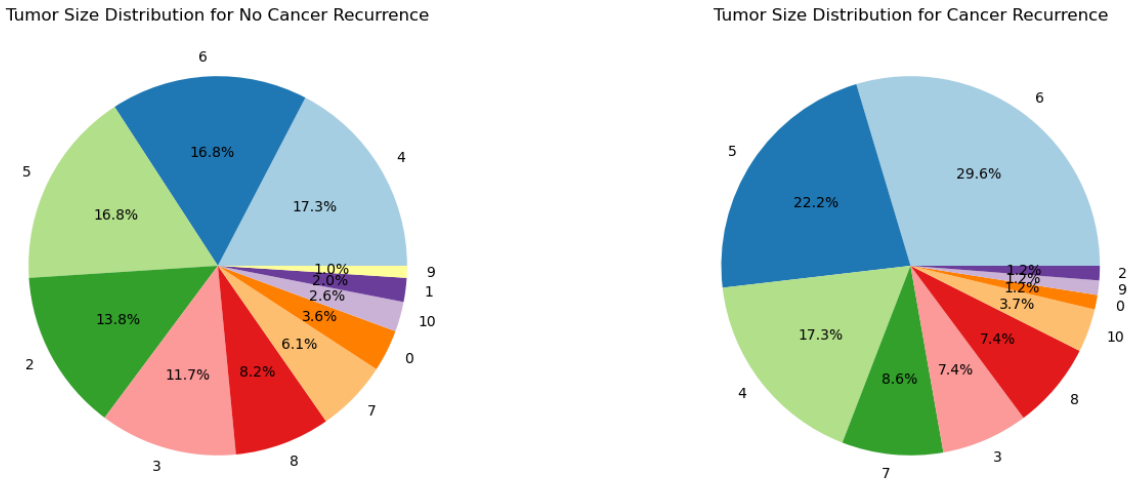

Figure 58: Pie charts with distribution of tumor size (in mm) for no cancer recurrence (left) and cancer recurrence (right)

C.1.5 Radiotherapy

Radiotherapy can be combined with surgery in case the breast cancer is more advanced. This treatment could be given as a first step to shrink a tumor before the main treatment, indicated as neoadjuvant radiotherapy. Therefore, the use of ionizing radiation is an attempt to downstage the tumor and it favors the tumor resectability. [54, 55] As illustrated in the two pie charts in figure 59, it is clear that when the patient needed radiotherapy in addition to the removal of the tumor, the chances of recurrence of the cancer is larger. This is related to the fact that additional radiotherapy is needed when the tumor stage is more advanced.

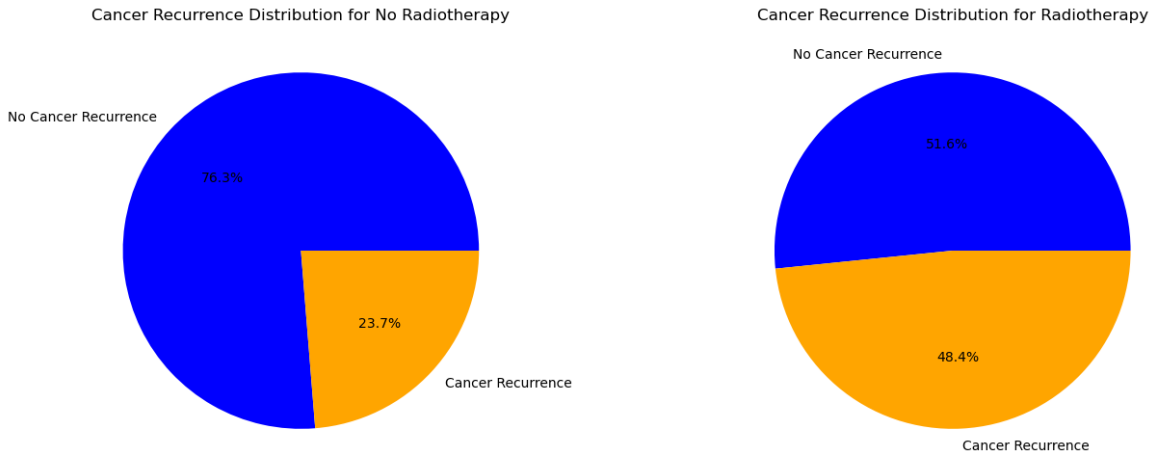

Figure 59: Pie charts with distribution of cancer recurrence for no radiotherapy (left) and radiotherapy (right)

C.1.6 Left or right breast

Several studies have shown that unilaterial breast cancer is slightly more frequent in the left breast than in the right. [**?**] This laterality - meaning an increased frequency of left-side breast cancers compared to right-sided - is not yet well understood, although it is still observed. [57] This trend is also visible in the dataset used in this work, illustrated in figure 60, where 31% of the patients developing a cancer in the left breast suffered from cancer recurrence compared to 27.3% of the patients with a cancer in the right breast.

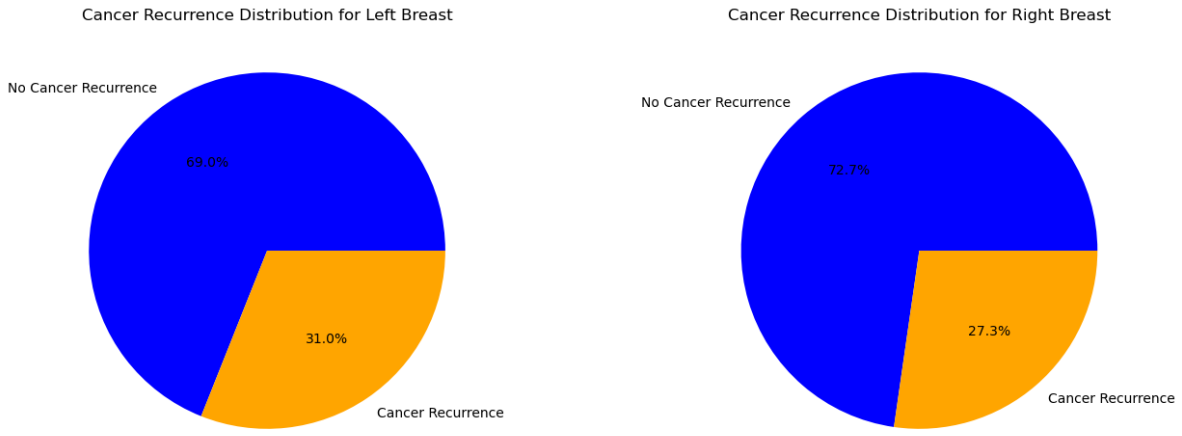

Figure 60: Pie charts with distribution of cancer recurrence for left breast (left) and right breast (right)

C.1.7 Cancer LN capsule

Tumor cells can pass through and spread outside of the lymph node capsule, indicated as extracapsular extension (ECE). This is an indicator of poor prognosis as ECE is correlated with lymphovascular invasion and macrometastases in the sentinel lymph node. [58] This negative impact of cancer going through the LN capsule can be observed in figure 61, where a clear increase of cancer recurrence is visible in patients where the tumor cells have propagated outside of the LN capsule.

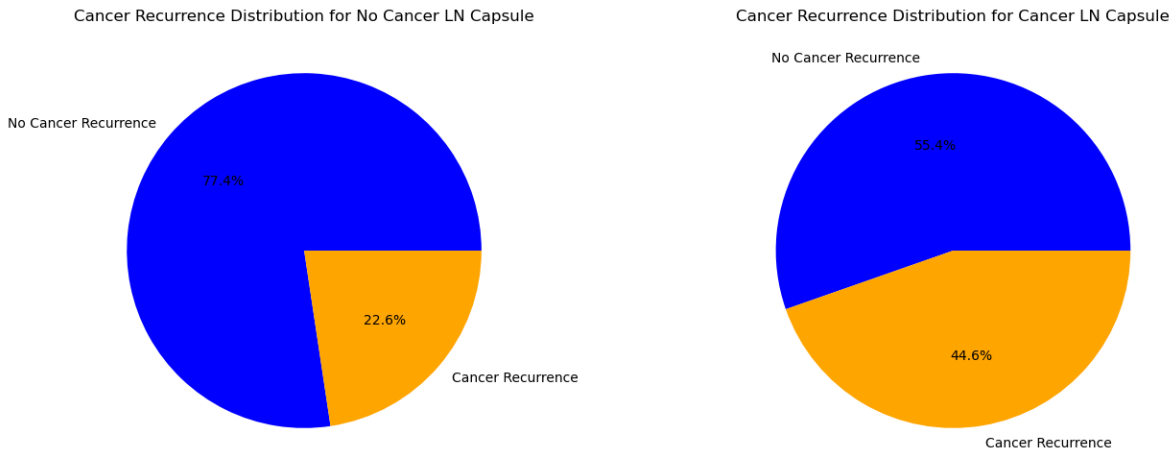

Figure 61: Pie charts with distribution of cancer recurrence for no cancer out of LN capsule (left) and cancer out of LN capsule (right)

C.1.8 Tumor grade

The higher the grade of the tumor, the higher the probability of recurrence of the tumor. This is visible on figure 62, in which the probabilities of developing a new cancer increases with increasing severity of malignancy. Literature supports this observation, in which patients with grade 3 tumors significantly developed more frequent distant metastases compared to patients with a tumor grade of 1. [59]

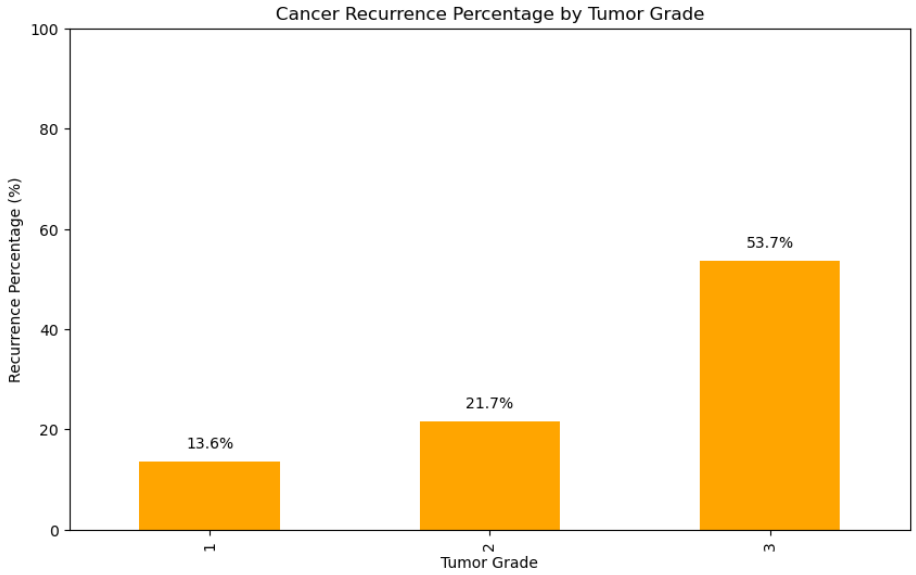

Figure 62: Probability of cancer recurrence based on tumor grade

C.1.9 Amount of LNs infiltrated with cancer

When no lymph nodes were infiltrated with cancerous cells, the probability is higher of not having a recurrence of breast cancer. This is visible on figure 63. This is because these so-called in-situ cancers are less invasive and therefore can be more easily removed upon surgery, without keeping residual tumor cells in the breast.

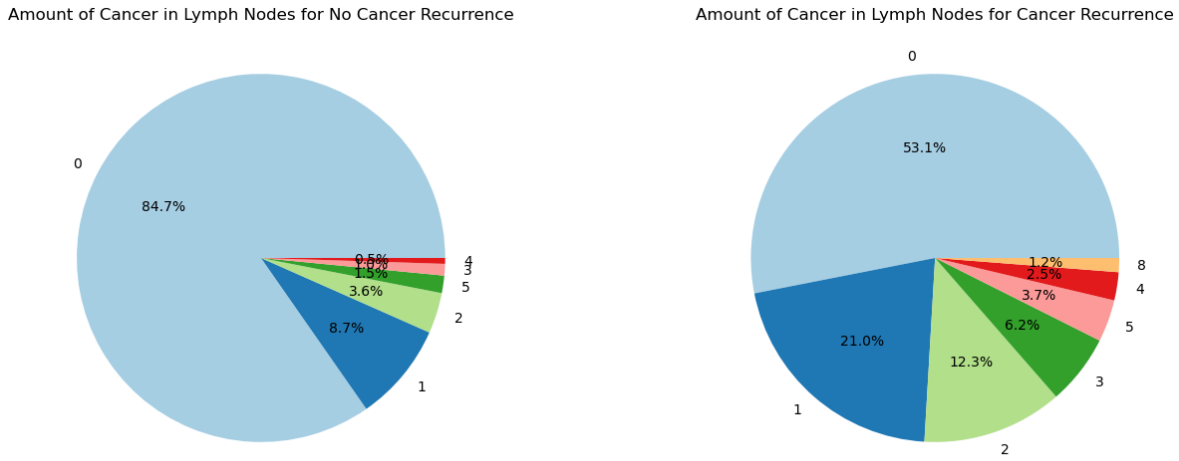

Figure 63: Pie charts with distribution of amount of lymph nodes infiltrated by cancerous tissue for no cancer recurrence (left) and cancer recurrence (right)

C.1.10 Location in affected breast

Research shows that breast cancer is most likely to occur in the upper and outer quadrant. However, there was no direct association between quadrant tumor density and tumor occurrence. [60] In figure 64 four pie charts are visualized separated based on the left or right breast and cancer recurrence. It can be seen that for the left breast, the chances are a lot higher for developing a tumor in the outer (left) quadrants. However, this trend is not clearly visible for the right breast.

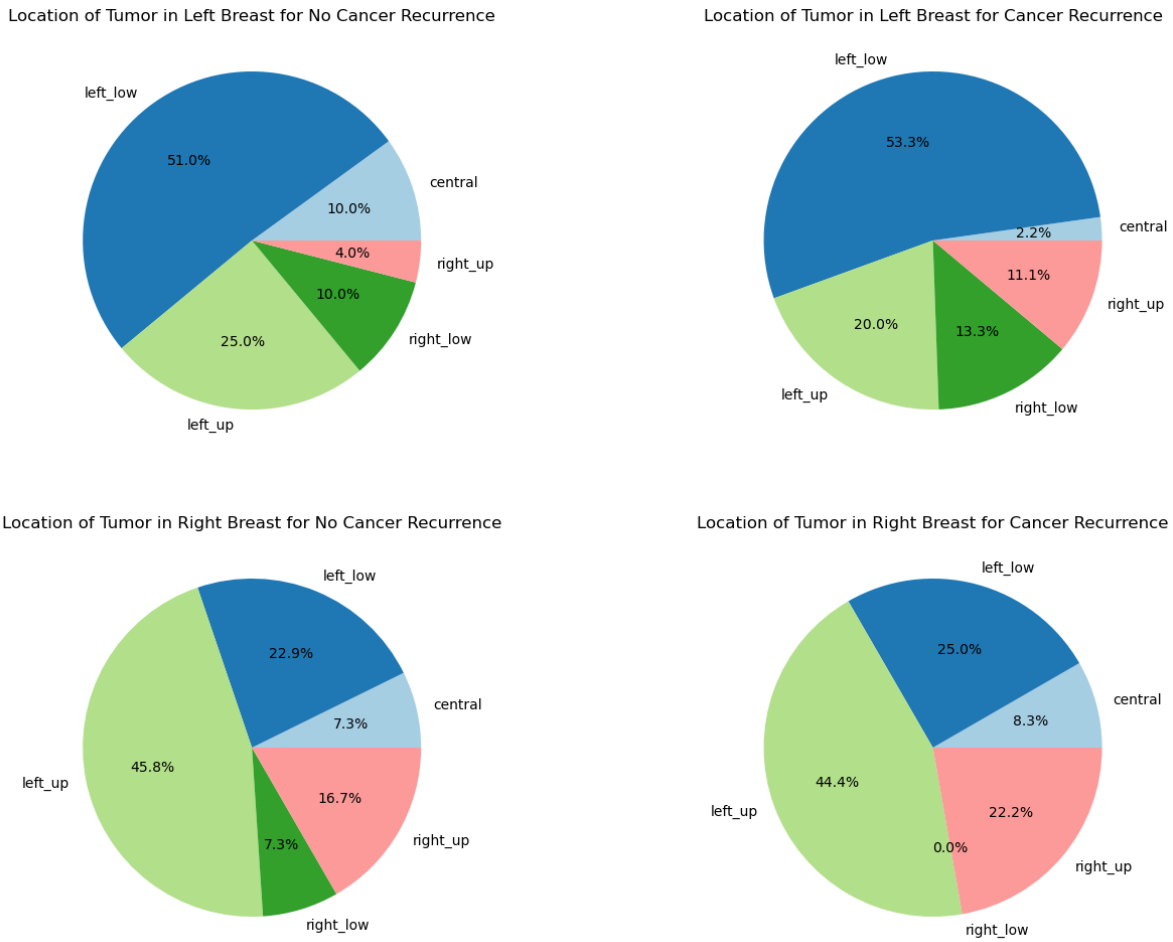

Figure 64: Pie charts with distribution of tumor location for in upper row the left breast and lower row the right breast, and left column no cancer recurrence and right cancer recurrence

C.2 Correlation

In figure 65, the correlation matrix of the features with the output variable is visible.

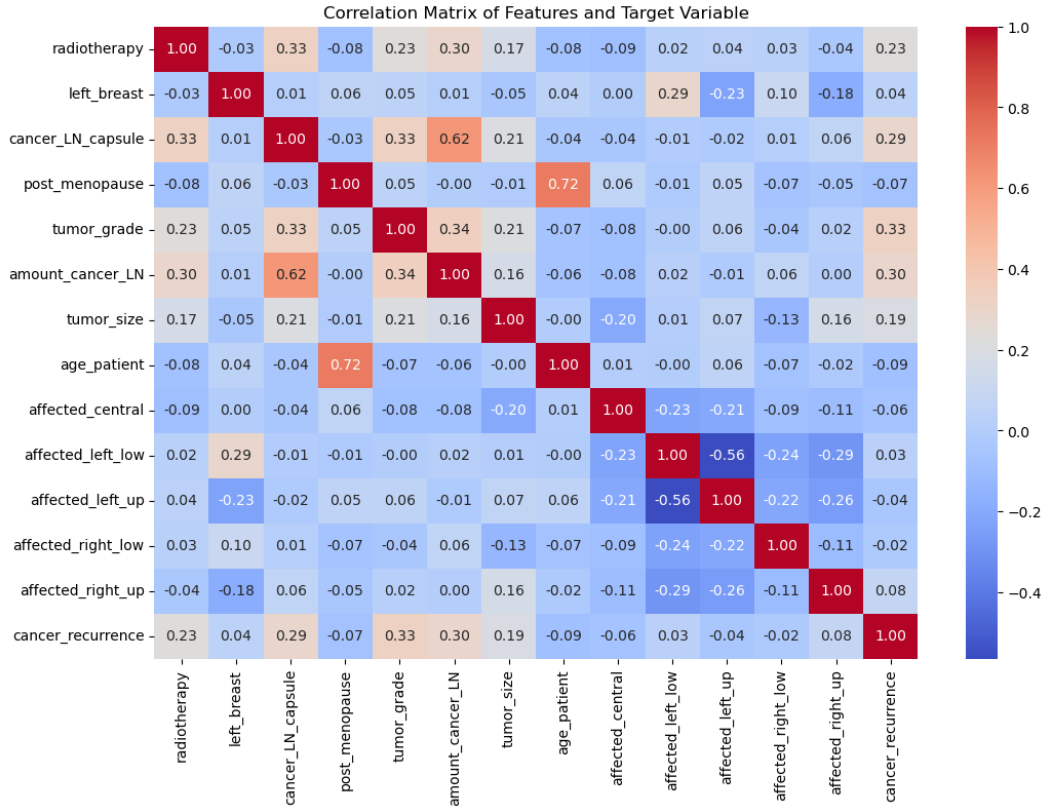

Figure 65: Correlation matrix of the outcome variables

In figure 66, it is clearly shown that the features having the strongest correlation with the outcome variable 'cancer recurrence' are 'tumor_grade', 'amount_cancer_LN', 'cancer_LN_capsule', 'radiotherapy' and 'tumor_size'. Therefore, it could be recommended to discard the features with a low correlation with the target variable in further analysis in order to reduce model complexibility and interpretability.

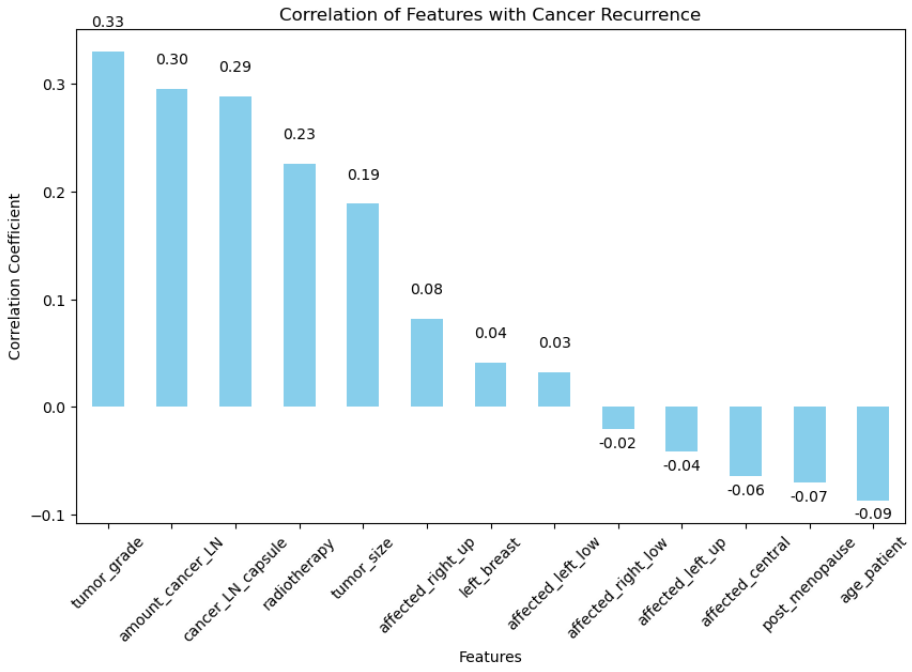

Figure 66: Correlation of features with outcome variable

D Additional online courses

In preparation for my master thesis, I followed the courses on "Neural Networks and Deep Learning" and on "Convolutional Neural Networks" on Coursera. These online courses consisted of videos recorded by the instructor Andrew Ng, as well as several notebooks that needed to be completed to translate the theoretical knowledge into practice.
For the first course, five notebooks were completed with the following subjects:

- Python Basics with Numpy
- Planar Data Classification with One Hidden Layer
- Logistic Regression with a Neural Network Mindset
- Building Your Deep Neural Network Step by Step
- Deep Neural Network Application

In the second course, the following notebooks were completed:

- TensorFlow Introduction
- Convolution Model Step by Step (v1)
- Convolution Model Application
- Residual Networks
- Image Segmentation with U-Net (v2)
- Face Recognition
- Autonomous Driving Application: Car Detection
- Transfer Learning with MobileNet (v1)
- Art Generation with Neural Style Transfer

After completing these courses, I obtained two certificates, which are shown in Figures 67 and 68. These classes have contributed to a broader and more profound knowledge of deep learning in general, as well as convolutional neural networks—knowledge that is essential for tackling the task of image classification and segmentation.

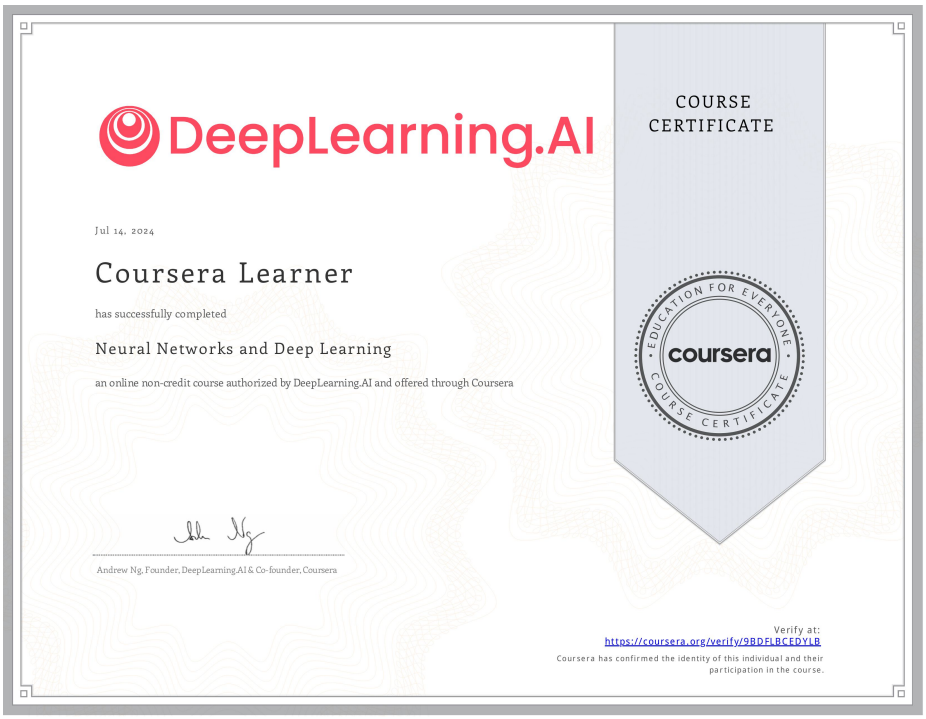

Figure 67: Certificate of Coursera course on Neural Networks and Deep Learning

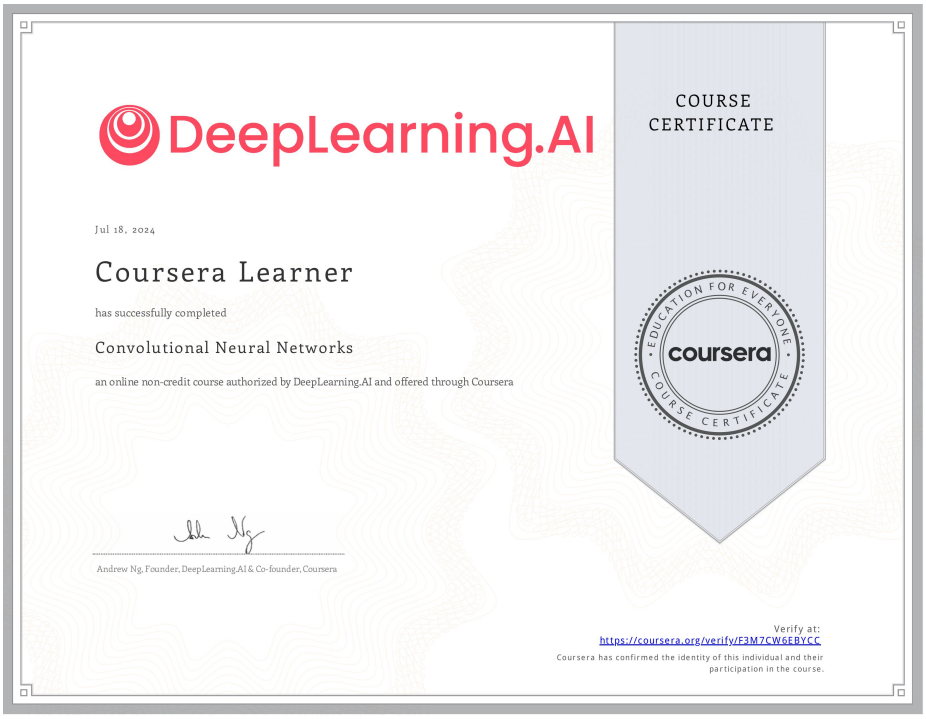

Figure 68: Certificate of Coursera course on Convolutional Neural Networks

References

[1] Malhotra, A., Ayappa, I., Ayas, N., Collop, N., Kirsch, D., Mcardle, N., Mehra, R., Pack, A. I., Punjabi, N., White, D. P., & Gottlieb, D. J. (2021). Metrics of sleep apnea severity: beyond the apnea-hypopnea index. SLEEP, 44(7). https://doi.org/10.1093/sleep/zsab030

[2] Slowik, J. M., Sankari, A., & Collen, J. F. (2024, March 21). Obstructive sleep apnea. StatPearls - NCBI Bookshelf. https://www.ncbi.nlm.nih.gov/books/NBK459252/

[3] Asghari, A., & Mohammadi, F. (2013, August 1). Is Apnea-Hypopnea Index a proper measure for Obstructive Sleep Apnea severity? https://pmc.ncbi.nlm.nih.gov/articles/PMC3917481/

[4] Sleep Cycle Centers. (2023, October 25). How common is sleep apnea? https://sleepcyclecenters.com/blog/how-common-is-sleep-apnea/

[5] George, C. F. P. (2004). Sleep middle dot 5: Driving and automobile crashes in patients with obstructive sleep apnoea/hypopnoea syndrome. Thorax, 59(9), 804–807. https://doi.org/10.1136/thx.2003.007187

[6] Tufik, S., Santos-Silva, R., Taddei, J. A., & Bittencourt, L. R. A. (2010). Obstructive sleep apnea syndrome in the Sao Paulo Epidemiologic Sleep Study. Sleep Medicine, 11(5), 441–446. https://doi.org/10.1016/j.sleep.2009.10.005

[7] Senaratna, C. V., Perret, J. L., Lodge, C. J., Lowe, A. J., Campbell, B. E., Matheson, M. C., Hamilton, G. S., & Dharmage, S. C. (2016). Prevalence of obstructive sleep apnea in the general population: A systematic review. Sleep Medicine Reviews, 34, 70–81. https://doi.org/10.1016/j.smrv.2016.07.002

[8] Bixler, E. O., Vgontzas, A. N., Have, T. T., Tyson, K., & Kales, A. (1998). Effects of age on sleep apnea in men. American Journal of Respiratory and Critical Care Medicine, 157(1), 144–148. https://doi.org/10.1164/ajrccm.157.1.9706079

[9] Ghuman, M., & St Anna, L. (2011, May 1). Clinical indicators of Obstructive sleep apnea. AAFP. https://www.aafp.org/pubs/afp/issues/2011/0501/od1.html

[10] Hoffstein, V., & Szalai, J. P. (1993). Predictive value of clinical features in diagnosing obstructive sleep apnea. Sleep, 16(2), 118–122. https://pubmed.ncbi.nlm.nih.gov/8446830/

[11] Kapur, V., Blough, D. K., Sandblom, R. E., Hert, R., De Maine, J. B., Sullivan, S. D., & Psaty, B. M. (1999). The medical cost of undiagnosed sleep apnea. SLEEP, 22(6), 749–755. https://doi.org/10.1093/sleep/22.6.749

[12] Zou, J., Guan, J., Yi, H., Meng, L., Xiong, Y., Tang, X., Su, K., & Yin, S. (2013). An Effective model for screening Obstructive Sleep Apnea: A Large-Scale Diagnostic Study. PLoS ONE, 8(12), e80704. https://doi.org/10.1371/journal.pone.0080704

[13] Rowley, J. A., Aboussouan, L. S., & Badr, M. S. (2000). The use of clinical prediction formulas in the evaluation of obstructive sleep apnea. SLEEP, 23(7), 929–938. https://doi.org/10.1093/sleep/23.7.929

[14] MLOps: Continuous delivery and automation pipelines in machine learning. (2024, August 28). Google Cloud. https://cloud.google.com/architecture/mlops-continuous-delivery-and-automation-pipelines-in-machine-learning

[15] Website, N. (2024, September 17). Calculate your body mass index (BMI) for adults - NHS. nhs.uk. https://www.nhs.uk/health-assessment-tools/calculate-your-body-mass-index/calculate-bmi-for-adults

[16] Bobbitt, Z. (2021, May 12). When to Use Mean vs. Median (With Examples). Statology. https://www.statology.org/when-to-use-mean-vs-median/

[17] Simplilearn. (2023, August 16). Introduction to data imputation. Simplilearn.com. https://www.simplilearn.com/data-imputation-article

[18] GeeksforGeeks. (2024, May 23). Scatter Plot Matrix. GeeksforGeeks. https://www.geeksforgeeks.org/scatter-plot-matrix/

[19] Schober, P., Boer, C., & Schwarte, L. A. (2018). Correlation Coefficients: appropriate use and interpretation. Anesthesia & Analgesia, 126(5), 1763–1768. https://doi.org/10.1213/ane.0000000000002864

[20] Chaudhary, K. (2024, October 10). How to deal with Imbalanced data in classification? Medium. https://medium.com/game-of-bits/how-to-deal-with-imbalanced-data-in-classification-bd03cfc66066

[21] Trotta, F., & Ackerson, D. (2024, March 7). How to handle imbalanced data for Machine Learning in Python. Semaphore. https://semaphoreci.com/blog/imbalanced-data-machine-learning-python

[22] Roepke, B., & Roepke, B. (2021, November 25). Don't Get Caught in the Trap of Imbalanced Data When Building a Model. https://dataknowsall.com/blog/imbalanced.html

[23] Over-sampling methods — Version 0.12.4. (n.d.). https://imbalanced-learn.org/stable/references/over_sampling.html

[24] Song, Y., Westerhuis, J. A., Aben, N., Michaut, M., Wessels, L. F. A., & Smilde, A. K. (2017). Principal component analysis of binary genomics data. Briefings in Bioinformatics, 20(1), 317–329. https://doi.org/10.1093/bib/bbx119

[25] Importance of feature scaling. (n.d.). Scikit-learn. https://scikit-learn.org/1.5/auto_examples/preprocessing/plot_scaling_importance.html

[26] Team, B. (2018, September 18). How to read PCA biplots and scree plots - BioTuring Team - Medium. Medium. https://bioturing.medium.com/how-to-read-pca-biplots-and-scree-plots-186246aae063

[27] Erdem, K. (2022, July 21). t-SNE clearly explained - Towards Data Science. Medium. https://towardsdatascience.com/t-sne-clearly-explained-d84c537f53a

[28] BI, P. (2024, November 22). Understanding feature scaling in Machine Learning - Punyakeerthi BL - Medium. Medium. https://medium.com/@punya8147_26846/understanding-feature-scaling-in-machine-learning-fe2ea8933b66

[29] Shaibu, S. (2024, October 15). Normalization vs. Standardization: How to Know the Difference. https://www.datacamp.com/tutorial/normalization-vs-standardization

[30] Jiang, W. (2021). Applications of deep learning in stock market prediction: Recent progress. Expert Systems With Applications, 184, 115537. https://doi.org/10.1016/j.eswa.2021.115537

[31] Importance of feature scaling. (n.d.). Scikit-learn. https://scikit-learn.org/1.5/auto_examples/preprocessing/plot_scaling_importance.html

[32] Pinjosovsky, S. B., PhD. (2023, July 6). Normalize data before or after split of training and testing data? Medium. https://medium.com/@spinjosovsky/normalize-data-before-or-after-split-of-training-and-testing-data-7b8005f81e26

[33] 3.1. Cross-validation: evaluating estimator performance. (n.d.). Scikit-learn. https://scikit-learn.org/1.5/modules/cross_validation.html#cross-validation

[34] Ploomber. (2022, April 11). Model selection done right: A gentle introduction to nested cross-validation. https://ploomber.io/blog/nested-cv/

[35] Wong, T., & Yeh, P. (2019). Reliable Accuracy Estimates from k-Fold Cross Validation. IEEE Transactions on Knowledge and Data Engineering, 32(8), 1586–1594. https://doi.org/10.1109/tkde.2019.2912815

[36] Cawley, G. C., & Talbot, N. L. C. (2010). On over-fitting in model selection and subsequent selection bias in performance evaluation. https://www.jmlr.org/papers/v11/cawley10a.html

[37] Brownlee, J. (2021, November 19). Nested Cross-Validation for Machine Learning with Python. MachineLearningMastery.com. https://machinelearningmastery.com/nested-cross-validation-for-machine-learning-with-python/

[38] Nested versus non-nested cross-validation. (n.d.). Scikit-learn. https://scikit-learn.org/stable/auto_examples/model_selection/plot_nested_cross_validation_iris.html

[39] Qualtrics. (n.d.). https://www.qualtrics.com/support/stats-iq/analyses/regression-guides/interpreting-residual-plots-improve-regression/

[40] Tian, Y., Shi, Y., Chen, X., & Chen, W. (2011). AUC Maximizing Support Vector Machines with Feature Selection. Procedia Computer Science, 4, 1691–1698. https://doi.org/10.1016/j.procs.2011.04.183

[41] Brazier, Y. (2024, September 4). How much should I weigh for my height and age? https://www.medicalnewstoday.com/articles/323446#body-mass-index-bmi

[42] Lundberg, S., & Lee, S. (2017, May 22). A unified approach to interpreting model predictions. arXiv.org. https://arxiv.org/abs/1705.07874

[43] Lini, R. (2024, November 25). Ablation Study: What is it, in Machine Learning? - Raji Lini - Medium. Medium. https://medium.com/@rajilini/ablation-study-what-is-it-in-machine-learning-0a1d362b366d

[44] How to build an End-to-End machine learning project? (2024, October 28). ProjectPro. https://www.projectpro.io/article/end-to-end-machine-learning-project/1047

[45] World Health Organization: WHO & World Health Organization: WHO. (2024, March 13). Breast cancer. https://www.who.int/news-room/fact-sheets/detail/breast-cancer

[46] Gupta, S., Kumar, D., & Sharma, A. (2011). Data mining classification techniques applied for breast cancer diagnosis and prognosis. Indian Journal of Computer Science and Engineering (IJCSE), 2(2), 188-195.

[47] Cesar, M. R., German, L., Patricia, A. P., Eugenia, A., Clementina, O. E., Jose, C., Alberto, P. M., Enrique, M. F., & Margarita, R. (2020). Method based on data mining techniques for breast cancer recurrence analysis. In Lecture notes in computer science (pp. 584–596). https://doi.org/10.1007/978-3-030-53956-6_54

[48] Zwitter, M. & Soklic, M. (1988). Breast Cancer [Dataset]. UCI Machine Learning Repository. https://doi.org/10.24432/C51P4M

[49] GeeksforGeeks. (2024, March 21). One hot encoding in machine learning. GeeksforGeeks. https://www.geeksforgeeks.org/ml-one-hot-encoding/

[50] White, M. C., Holman, D. M., Boehm, J. E., Peipins, L. A., Grossman, M., & Henley, S. J. (2014). Age and Cancer Risk. American Journal of Preventive Medicine, 46(3), S7–S15. https://doi.org/10.1016/j.amepre.2013.10.029

[51] HRT, Menopause and breast cancer — Breast Cancer UK. (2024, July 19). Breast Cancer UK. https://www.breastcanceruk.org.uk/resources/hrt-menopause-and-breast-cancer/

[52] Surakasula, A., Nagarjunapu, G., & Raghavaiah, K. (2014). A comparative study of pre- and post-menopausal breast cancer: Risk factors, presentation, characteristics and management. Journal of Research in Pharmacy Practice, 3(1), 12. https://doi.org/10.4103/2279-042x.132704

[53] What is menopause? (2024, October 16). National Institute on Aging. https://www.nia.nih.gov/health/menopause/what-menopause

[54] NCI Dictionary of Cancer Terms. (n.d.). Cancer.gov. https://www.cancer.gov/publications/dictionaries/cancer-terms/def/neoadjuvant-therapy

[55] Sousa, C., Cruz, M., Neto, A., Pereira, K., Peixoto, M., Bastos, J., Henriques, M., Roda, D., Marques, R., Miranda, C., Melo, G., Sousa, G., Figueiredo, P., & Alves, P. (2020). Neoadjuvant radiotherapy in the approach of locally advanced breast cancer. ESMO Open, 5(2), e000640. https://doi.org/10.1136/esmoopen-2019-000640

[56] Tulinius, H., Sigvaldason, H., & Ólafsdóttir, G. (1990). Left and right sided breast cancer. Pathology - Research and Practice, 186(1), 92–94. https://doi.org/10.1016/s0344-0338(11)81015-0

[57] Saad, S. A., Shenawi, H. A., Almarabheh, A., Shenawi, N. A., Mohamed, A. I., & Yaghan, R. (2022). Is laterality in breast Cancer still worth studying? Local experience in Bahrain. BMC Cancer, 22(1). https://doi.org/10.1186/s12885-022-10063-y

[58] Gooch, J., King, T. A., Eaton, A., Dengel, L., Stempel, M., Corben, A. D., & Morrow, M. (2014). The Extent of Extracapsular Extension May Influence the Need for Axillary Lymph Node Dissection in Patients with T1–T2 Breast Cancer. Annals of Surgical Oncology, 21(9), 2897–2903. https://doi.org/10.1245/s10434-014-3752-0

[59] Gayar, O. H., Patel, S., Schultz, D., Mahan, M., Rasool, N., & Elshaikh, M. A. (2013). The impact of tumor grade on survival end points and patterns of recurrence of 949 patients with Early-Stage Endometrioid carcinoma: a Single institution study. International Journal of Gynecological Cancer, 24(1), 97–101. https://doi.org/10.1097/igc.0000000000000018

[60] Chan, S., Chen, J., Li, S., Chang, R., Yeh, D., Chang, R., Yeh, L., Kwong, J., & Su, M. (2017). Evaluation of the association between quantitative mammographic density and breast cancer occurred in different quadrants. BMC Cancer, 17(1). https://doi.org/10.1186/s12885-017-3270-0