|  |  |
| --- | --- |
| Old | new |
| Cardiovascular diseases, including heart disease, are among the leading causes of morbidity and mortality worldwide. Early detection and prevention of heart disease are crucial to reduc- ing the burden on healthcare systems and improving patient outcomes [1]. Machine learning techniques have shown great potential in the analysis of complex datasets and the prediction of various medical conditions, including heart disease [2]. By using machine learning models, researchers and healthcare professionals can identify patterns and relationships in the data that may not be apparent through traditional statistical methods. In this study, we aim to explore the effectiveness of four popular machine learning techniques – Linear Regression, Logistic Regression, Neural Networks, and Support Vector Machines (SVM) – in predicting heart disease using a dataset consisting of 303 instances and 75 attributes. | Cardiovascular diseases, including heart disease, are among the leading causes of morbidity and mortality worldwide. Early detection and prevention of heart disease are crucial to reduc- ing the burden on healthcare systems and improving patient outcomes [1]. Machine learning techniques have shown great potential in the analysis of complex datasets and the prediction of various medical conditions, including heart disease [2]. By using machine learning models, researchers and healthcare professionals can identify patterns and relationships in the data that may not be apparent through traditional statistical methods |
| The application of machine learning in the medical do- main has been growing rapidly in recent years, driven by advancements in computational power and the availability of vast amounts of data. Machine learning models have been successful in diagnosing and predicting a wide range of med- ical conditions, enabling more accurate and efficient patient care. Heart disease prediction is an essential area of research, as timely diagnosis can significantly improve treatment and management strategies, ultimately saving lives. The motivation behind this study is to compare the performance of various ma- chine learning techniques in predicting heart disease, providing insights into their strengths and limitations and informing the choice of appropriate models for future research and clinical applications. | The application of machine learning in the medical do- main has been growing rapidly in recent years, driven by advancements in computational power and the availability of vast amounts of data [2]. Machine learning models have been successful in diagnosing and predicting a wide range of medical conditions, enabling more accurate and efficient patient care [3]. The motivation behind this study is to compare the performance of various machine learning techniques in predicting heart disease, providing insights into their strengths and limitations and informing the choice of appropriate models for future research and clinical applications |
| This paper is organized into the following sections:  Section 2 provides an overview of the dataset used in the study, discussing the data source, attributes, distribution, and associated challenges.  Section 3 describes the data preprocessing steps, including handling missing values, feature selection, feature scaling and normalization, and train-test split.  Section 4 introduces the four machine learning models used in the study, explaining their underlying concepts and mechanisms.  Section 5 covers the model selection and hyperparameter tuning process, discussing the use of cross-validation techniques, grid search, random search, and model selection criteria.  Section 6 presents the evaluation and comparison of the models, detailing the evaluation metrics used and providing a comprehensive analysis of each model's performance in predicting heart disease. Feature importance analysis is also discussed in this section.  Section 7 offers a discussion of the key findings, strengths and limitations of each model, and their practical implications in the context of heart disease prediction.  Section 8 concludes the paper by summarizing the results, providing recommendations for future research directions, and highlighting the potential impact of this study on heart disease prediction and management.  Finally, Section 9 lists the references used throughout the paper. | This paper is organized into the following sections. Section II provides an overview of the dataset used in the study, explaining attributes, their distribution and explaining chal- lenges concerning the data. In Section III we explain the preprocessing steps including e.g. handling missing values and normalization. In Section IV we introduce the four machine learning models used in the study shortly, explaining their underlying concepts and mechanisms. We then continue to explain the model selection process and hyperparameter tuning in Section V, where we discuss cross-validation as well as grid search. Section VI presents the evaluation and comparison of the models, detailing the evaluation metrics used and providing a comprehensive analysis of each model’s performance in predicting heart disease. Feature importance analysis is also discussed in this section. In Section VII we discuss the key findings, as well as the strenghts and limitations of the models considering their practical implications in the context of heart disease prediction. We finally conclude by summarizing results and providing a recommendation for future research |
| The heart disease dataset used in this study was collected from four different institutions, and it comprises a combination of continuous, discrete, and categorical attributes. The dataset contains 303 instances and 75 attributes, which are useful for performing classification tasks related to heart disease diagnosis  The authors of the databases have requested that any publications resulting from the use of the data acknowledge the principal investigators responsible for the data collection at each institution. The principal investigators and their respective institutions are as follows:  \begin{itemize}  \item Hungarian Institute of Cardiology, Budapest: Dr. Andras Janosi  \item University Hospital, Zurich, Switzerland: Dr. William Steinbrunn  \item University Hospital, Basel, Switzerland: Dr. Matthias Pfisterer  \item V.A. Medical Center, Long Beach, and Cleveland Clinic Foundation: Dr. Robert Detrano  \end{itemize}  In the dataset, the attributes consist of various factors related to heart disease, such as age, sex, chest pain type, blood pressure, cholesterol level, and electrocardiographic results, among others. A detailed description of these attributes can be found in the previously provided table (Table~\ref{tab:variables}). The dataset also contains missing values, which need to be appropriately handled during the preprocessing stage before applying machine learning algorithms. | The heart disease dataset used in this study was collected from V.A. Medical Center, Long Beach, and Cleveland Clinic Foundation: Dr. Robert Detrano [4]. The dataset contains 303 instances and 75 attributes, which are useful for performing classification tasks related to heart disease diagnosis. The dataset can be extended by data from the following institutions: • Hungarian Institute of Cardiology, Budapest: Dr. Andras Janosi • University Hospital, Zurich, Switzerland: Dr. William Steinbrunn • University Hospital, Basel, Switzerland: Dr. Matthias Pfisterer • V.A. Medical Center, Long Beach, and Cleveland Clinic Foundation: Dr. Robert Detrano Combining all datasets into one leads to a larger number of missing values. Therefore all machine-learning algorithms have been performed on the Cleveland dataset [4]. Although the dataset contains of 75 attributes, only a subset of these attributes have been used in former publications. This subset contains 14 attributes, which are contained in a preprocessed file that we build the machine learning of this research paper on. In the dataset, the attributes consist of various factors related to heart disease, such as age, sex, chest pain type, blood pressure, cholesterol level, and electrocardiographic results, among others. A detailed description of these attributes can be found in the previously provided table (Table I). The dataset also contains missing values, which need to be appropriately handled during the preprocessing stage before applying ma- chine learning algorithms. |
| C. Associated Tasks and Challenges The primary objective of this study is to perform classi- fication tasks using the heart disease dataset. The goal is to create a model capable of predicting the presence of heart disease based on the attributes provided. However, working with real-world datasets, especially in the medical domain, presents several challenges that need to be addressed to ensure the effectiveness and reliability of the models developed. Some of the key tasks and challenges associated with this project are: 1) Handling missing data: The dataset contains missing values, which can impact the performance of the ma- chine learning models. Appropriate techniques, such as imputation or removal of instances with missing values, need to be employed to ensure the integrity of the dataset used for model training and evaluation. 2) Feature selection: With 75 attributes in the dataset, it is essential to determine the most relevant features that contribute to accurate heart disease prediction. Tech- niques such as correlation analysis, mutual information, or feature importance ranking can be employed to select the most informative features and reduce the complexity of the models. 3) Data preprocessing: The dataset contains categorical, integer, and continuous variables. These diverse data types require appropriate preprocessing, such as nor- malization, scaling, or encoding, to ensure compatibility with the machine learning algorithms used for classifi- cation tasks. 4) Model selection and evaluation: Four machine learning models—linear regression, logistic regression, neural networks, and support vector machines—will be used in this study. Each model’s performance needs to be carefully evaluated and compared using relevant metrics such as accuracy, precision, recall, F1 score, and area under the ROC curve. 5) Model interpretability: As the models are intended for medical applications, it is crucial to ensure that they are interpretable and can provide insights into the relationships between the input features and the pre- dicted outcomes. Techniques such as feature importance analysis, partial dependence plots, or local interpretable model-agnostic explanations (LIME) can be employed to improve model interpretability. 6) Validation and generalization: The models developed should be validated using appropriate techniques, such as cross-validation or holdout validation, to ensure that they generalize well to new, unseen data. This is par- ticularly important in medical applications, where the consequences of incorrect predictions can be severe. Addressing these tasks and challenges is crucial for devel- oping accurate and reliable models for heart disease prediction using the given dataset |  |
| Data Preprocssing |  |
| The dataset used in this study has been collected from multiple sources, including the Cleveland Clinic Foundation, the Hungarian Institute of Cardiology, the University Hospital in Zurich, and the University Hospital in Basel. The data preprocessing is an essential step before training any machine learning model, as it helps to clean, format, and organize the data, making it suitable for analysis. In this section, we present the preprocessing steps taken, which include handling missing values, feature selection, feature scaling and normalization, and train-test split. We will discuss each step in detail to provide a comprehensive understanding of the data preprocessing stage. |  |
|  |  |
|  |  |
| The dataset contains 13 features, including continuous, discrete, and binary variables. We have divided these features into two groups: continuous and discrete factors. Continuous factors include age, cholesterol levels, oldpeak, maximum heart rate, and resting blood pressure. These variables represent critical aspects of a patient's medical history and are directly related to heart disease risk. Discrete factors consist of chest pain type, exercise-induced angina, fasting blood sugar, resting electrocardiographic results, sex, slope of the peak exercise ST segment, number of major vessels, and thalassemia. These variables provide additional insights into a patient's condition and can help improve the model's ability to predict heart disease.  Feature selection is an essential step in the preprocessing pipeline, as it helps identify the most relevant variables for the task at hand. By selecting the appropriate features, we can reduce the complexity of the model and improve its interpretability, while also reducing the risk of overfitting. In this study, we have chosen to include all available features in the dataset, as they represent a comprehensive set of variables related to heart disease risk. This approach ensures that the model has access to the most relevant information for making accurate predictions. | The dataset contains 13 features, including continuous, discrete, and binary variables. We have divided these features into two groups: continuous and discrete factors. Continuous factors include age, cholesterol levels, oldpeak, maximum heart rate, and resting blood pressure. These variables represent critical aspects of a patient's medical history and are directly related to heart disease risk. Discrete factors consist of chest pain type, exercise-induced angina, fasting blood sugar, resting electrocardiographic results, sex, slope of the peak exercise ST segment, number of major vessels, and thalassemia. These variables provide additional insights into a patient's condition and can help improve the model's ability to predict heart disease.  According to \cite{Dua2023} all contributions worked with the preprocessed dataset, based upon the 13 features that have already been selected. To compare our results, we decided to use the same features. After a first comparison, we perform feature selection algorithms to improve our results. |
| Feature scaling and normalization are essential steps in data preprocessing, as they ensure that all features have the same scale and do not dominate each other during the model training. In this study, we used different techniques for continuous and discrete variables to achieve the best results.  For continuous variables, we first replaced missing values with the median of the corresponding feature. Then, we scaled the data by subtracting the mean and dividing by the standard deviation. This process, known as standardization, ensures that the continuous features have a mean of zero and a standard deviation of one. Standardization is particularly important for models that are sensitive to the scale of the input features, such as linear regression and support vector machines. By standardizing the continuous variables, we ensure that they contribute equally to the model, leading to better performance and more accurate predictions.  For discrete variables, we performed one-hot encoding, which transformed each categorical variable into a set of binary variables. This process is necessary because many machine learning algorithms, including neural networks, cannot handle categorical variables directly. One-hot encoding creates a new binary feature for each category, with a value of 1 if the original feature matches the category and 0 otherwise. This transformation allows the model to capture the relationships between categories without making any assumptions about their ordinal nature. One-hot encoding also ensures that the discrete variables are on a similar scale as the continuous variables, which can improve the model's performance and convergence during training.  After performing feature scaling and normalization, we combined the continuous and discrete variables to create a fully preprocessed dataset. This dataset was then used as input for the various machine learning models explored in this study. | For continuous variables, we first replaced missing values with the median of the corresponding feature. Then, we scaled the data by subtracting the mean and dividing by the standard deviation. This process, known as standardization, ensures that the continuous features have a mean of zero and a standard deviation of one. Standardization is particularly important for models that are sensitive to the scale of the input features. By standardizing the continuous variables, we ensure that they contribute equally to the model, leading to better performance and more accurate predictions.  For discrete variables, we performed one-hot encoding, which transformed each categorical variable into a set of binary variables. This process is necessary because many machine learning algorithms, including neural networks, cannot handle categorical variables directly. One-hot encoding creates a new binary feature for each category, with a value of 1 if the original feature matches the category and 0 otherwise. One-hot encoding also ensures that the discrete variables are on a similar scale as the continuous variables, which can improve the model's performance and convergence during training. |
|  |  |
| An essential step in any machine learning pipeline is splitting the dataset into training, validation, and test sets. This process ensures that the model can be trained and evaluated on different subsets of the data, reducing the risk of overfitting and providing an unbiased estimate of its performance. In this study, we performed an 80-20 train-test split, where 80\% of the data was used for training and 20\% for testing. This split provides a good balance between the amount of data available for training and the size of the test set for evaluating the model's performance.  To ensure that the train-test split was representative of the overall data distribution, we used stratified sampling based on the target variable (heart disease diagnosis). This approach ensures that the proportion of positive and negative cases in the training and test sets is similar to the overall dataset, which can help improve the model's ability to generalize to new data.  In addition to the train-test split, we also considered using a separate validation set for hyperparameter tuning and model selection. However, due to the relatively small size of the dataset, we opted to use cross-validation during model training and hyperparameter optimization. Cross-validation involves dividing the training set into k-folds, where the model is trained on k-1 folds and evaluated on the remaining fold. This process is repeated k times, with each fold serving as the validation set once. The average performance across all folds is used as an estimate of the model's performance, allowing for a more robust evaluation while maximizing the use of the available data for training.  In summary, the data preprocessing stage involved handling missing values, selecting relevant features, scaling and normalizing the input variables, and splitting the dataset into training and test sets. These steps were critical for ensuring the quality and reliability of the subsequent machine learning models and their predictions. By carefully preprocessing the data, we were able to build models that can accurately predict heart disease risk and provide valuable insights for medical professionals and patients alike. | An essential step in any machine learning pipeline is splitting the dataset into training, validation, and test sets. This process ensures that the model can be trained and evaluated on different subsets of the data, reducing the risk of overfitting and providing an unbiased estimate of its performance. In this study, we performed an 80-20 train-test split, where 80\% of the data was used for training and 20\% for testing. This split provides a good balance between the amount of data available for training and the size of the test set for evaluating the model's performance. |
|  |  |
| Model Descriptions |  |
| In this section, we provide an short overview of the four machine learning models used for predicting heart disease in our study: Linear Regression, Logistic Regression, Neural Networks, and Support Vector Machines (SVM). Each model has its unique characteristics, assumptions, and strengths, making them suitable for different types of problems. We will briefly explain the underlying concepts and mechanisms of each model to provide a foundation for understanding their performance in the context of heart disease prediction. | In this section, we provide an short overview of the four machine learning models used for predicting heart disease in our study: Linear Regression, Logistic Regression, Neural Networks, and Support Vector Machines (SVM). We will briefly explain the underlying concepts and mechanisms of each model to provide a foundation for understanding their performance in the context of heart disease prediction. |
|  |  |
| Model selection and hyperparameter tuning |  |
| The process of model selection and hyperparameter tuning is critical for building an effective machine learning model. This section outlines the techniques used for cross-validation, hyperparameter optimization, model selection criteria, and hyperparameter tuning for each model | The process of model selection and hyperparameter tuning is critical for building an effective machine learning model. This section outlines the techniques used for cross-validation, hyperparameter optimization, model selection criteria, and hyperparameter tuning for each model |
|  |  |
| Grid search  In order to prevent overfitting and ensure a robust evaluation of the model’s performance, cross-validation techniques such as k-fold cross-validation and train-test split are employed. These techniques help to assess the model’s generalizability on unseen data. K-fold cross-validation involves partitioning the dataset into k equally sized subsets or ”folds”. The model is then trained k times, using each fold as the validation set once and the remaining k-1 folds as the training set. The average performance across all k iterations is considered as the model’s performance estimate. This technique helps to mitigate the risk of overfitting and provides a better estimate of the model’s performance on unseen data. Alternatively, the train-test split technique simply divides the dataset into two distinct sets – one for training and one for testing. This method is computationally less expensive compared to k-fold cross-validation. However, it might not provide as accurate an estimate of the model’s performance on unseen data. To implement these cross-validation techniques, we will use the Scikit-learn library, which provides various utility functions and tools to streamline the process. | Cross-validation techniques such as k-fold cross-validation and train-test split are used to assess the model's generalizability on unseen data. K-fold crosses-validation involves partitioning the dataset into k equally sized subsets or "folds". The model is then trained k times, using each fold as the validation set once and the remaining k-1 folds as the training set. This technique helps to mitigate the risk of overfitting and provides a better estimate of the model's performance. Train-test split splits the dataset into two distinct sets for training and testing. To implement these techniques, the Scikit-learn library provides various utility functions and tools to streamline the process. |
| Hyperparameter optimization is essential for improving the performance of machine learning models. Two popular tech- niques are grid search and random search. We will leverage gradient descent and use libraries to perform the optimization. Grid search involves exhaustively searching through a pre- defined set of hyperparameter values, evaluating the model performance for each combination. Although this approach is systematic, it can be computationally expensive, especially when dealing with a large number of hyperparameters. Random search, on the other hand, involves randomly sampling hyperparameter values from a specified distribution. This approach can be more efficient than grid search, as it does not require evaluating every possible combination of hyperparameters. However, it may not find the optimal hyperparameter values with the same certainty as grid search. | Hyperparameter optimization is essential for improving the performance of machine learning models. Two popular tech- niques are grid search and random search. We will leverage gradient descent and use libraries to perform the optimization. Grid search involves exhaustively searching through a pre- defined set of hyperparameter values, evaluating the model performance for each combination. Although this approach is systematic, it can be computationally expensive, especially when dealing with a large number of hyperparameters. |
| Model selection criteria  To select the best model, we need to establish a set of criteria that take into consideration the following hyperparameters:  Regularization parameter lambda: This parameter controls the amount of regularization applied to the model, preventing overfitting by adding a penalty term to the cost function.  Number of hidden layer units: This determines the complexity of the neural network model and can impact its ability to learn complex patterns in the data.  Number of hidden layers: Adding more hidden layers to a neural network can help it learn more complex representations of the data, but it can also lead to overfitting.  Sigma, C, k, etc.: These are additional hyperparameters specific to certain models, such as support vector machines and k-nearest neighbors classifiers.  We will employ a systematic approach to explore different combinations of these hyperparameters rather than randomly choosing values. | Model selection criteria include regularization parameter lambda, hidden layer units, number of hidden layers, Sigma, C, k, etc., and additional hyperparameters specific to certain models. We will employ a systematic approach to explore different combinations of these hyperparameters rather than randomly choosing values. |
| In this subsection, we will focus on tuning the hyperparameters for each model, which is a crucial step in developing an effective machine learning model. First, we will visualize the cost function trajectory over iterations graphically. This allows us to monitor the model's progress during training and identify potential issues, such as slow convergence or oscillations in the cost function value. Next, we will train each model using both regularized and nonregularized cost functions. Regularization is a technique used to prevent overfitting by adding a penalty term to the cost function, which discourages the model from assigning too much importance to individual features. Regularization can be particularly useful when dealing with high-dimensional datasets or when the number of features is larger than the number of samples. To find the optimal hyperparameter values for each model, we will employ techniques such as grid search, random search, or a combination of both. We will leverage gradient descent optimization and use libraries to perform the search efficiently. During the hyperparameter tuning process, we will use cross-validation techniques such as k-folds or train-test splits to ensure that the model generalizes well to unseen data and to prevent overfitting. By validating the model on different subsets of the data, we can obtain a more reliable estimate of its performance. Once we have identified the best performing model for each algorithm, we will fine-tune the hyperparameters further to achieve the highest possible performance. This may involve exploring a smaller range of hyperparameter values around the optimal values found in the initial search, or using more advanced optimization techniques such as Bayesian optimization or genetic algorithms. Throughout the hyperparameter tuning process, we will keep track of the model's performance on various evaluation metrics, such as accuracy, precision, recall, F1-score, and area under the ROC curve (AUC-ROC). This will help us identify the model that achieves the best balance between complexity and generalization performance, and ultimately, the most suitable model for the given dataset. | This subsection focuses on tuning the hyperparameters for each model, which is a crucial step in developing an effective machine learning model. To find the optimal hyperparameter values, we will employ techniques such as grid search, random search, or a combination of both. During the hyperparameter tuning process, we will use cross-validation techniques such as k-folds or train-test splits to ensure that the model generalizes well to unseen data and to prevent overfitting. We will also monitor the model's performance on various evaluation metrics, such as accuracy, precision, recall, F1-score, and area under the ROC curve (AUC-ROC). This will help us identify the model that achieves the best balance between complexity and generalization performance, and ultimately, the most suitable model for the given dataset. |
|  |  |
| MODEL EVALUATION AND COMPARISON |  |
| After completing the model selection and hyperparameter tuning process, we will evaluate and compare the performance of the different models. This section will provide a detailed overview of the evaluation metrics used, as well as a compre- hensive comparison of the models’ performance. |  |
| Accuracy measures the proportion of correct predictions made by the model. It is a commonly used metric for classification problems but may be misleading in cases where the class distribution is imbalanced. In such situations, the model may achieve high accuracy by simply predicting the majority class. However, accuracy still provides a base- line performance measure that can be easily understood and compared across models. | Accuracy is a measure of the proportion of correct predictions made by a model. It is used for classification problems, but may be misleading if the class distribution is imbalanced. However, accuracy still provides a base-line performance measure that can be understood and compared across models. |
| Precision measures the proportion of true positive predictions among all positive predictions, while recall measures the proportion of true positive predictions among all actual positive instances. The F1-score is the harmonic mean of precision and recall and provides a single metric that balances the trade-off between precision and recall. |  |
| The AUC-ROC is a measure of the model's ability to discriminate between positive and negative instances. It is computed by plotting the true positive rate (sensitivity or recall) against the false positive rate (1-specificity) at various threshold settings and calculating the area under the resulting curve. A higher AUC-ROC value indicates better model performance, with a value of 1 representing perfect discrimination and a value of 0.5 representing random guessing. The AUC-ROC is particularly useful for comparing models, as it is not affected by the threshold setting and provides a single metric that summarizes the model's discrimination ability across all possible thresholds. |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
| Feature importance analysis |  |
| Although this topic has been partially addressed in previous sections, we may choose to further investigate the importance of individual features in the final model. This can be done using techniques such as Lasso regularization, which assigns a penalty to the model’s coefficients and effectively reduces the importance of less relevant features. Alternatively, we can use common sense and domain knowl- edge to argue for the importance of specific features in the model. This can help provide insights into the underlying patterns in the data and inform future feature engineering efforts. From the results of the optimized hyperparameters, we can see that the Support Vector Machine (SVM) model has the highest accuracy and F1 Score. The SVM model also has the highest ROC AUC Score, indicating better discrimination between classes. The Lasso model has the second-highest performance across most of the metrics. The Neural Network model has the highest precision but lower recall, which may be a result of overfitting on the training data. Overall, the SVM model seems to be the best performing model for this dataset, given the optimized hyperparameters. However, depending on the specific needs of the problem, other models could be more appropriate based on different evaluation criteria |  |
|  |  |
|  |  |
|  |  |

\section{Help}

why heart disease recognition matters?

utility for society! (some reason..) \newline

motivation for picking this dataset

\subsection{Heart disease}

https://www.mayoclinic.org/diseases-conditions/heart-disease/symptoms-causes/syc-20353118

\subsection{Heart disease risk factors}

explain heart disease and contributors that lead to heart disease

\section{Data analysis}

explain data source (who got it, what hospitals...)

"Data description and preprocessing (if necessary normalization, feature selection,

transformation, etc.). Motivation for choosing the particular problem.

Data visualization (histograms, box plots, other plots)."

\subsection{Data source presentation}

hospitals, ...

\subsection{Data description}

mean, number features, important features, std (should be doable with .describe operator in python)

\subsection{Data visualization}

box plot, histogram \newline

maybe pose some hypothesis

\subsection{Preprocessing}

Normalization (min max normalization and mean-std normalization)

\section{Model application}

Increase model complexity throughout this chapter

structure of subsections: explain model mathematically, show implementation (pseudocode), some illustration of model (SVM, neural networks,...), important selected parameters in table, show result, some metric like accuracy

\begin{itemize}

\item Short description of the implemented ML models.

\item Model training (data splitting – train, validate, test, k-fold Cross validation)

\item Visualize graphically the cost function trajectory over iterations.

\item Training with regularized and nonregularized

cost function.

\item Model hyper-parameter selection - regularization parameter lambda, number of NN hidden

layer units, number of hidden layers (if necessary), sigma, C, k, etc.. Systematic approach

instead of just one or several randomly chosen values.

\item For a classification problem, you need to present the confusion Matrix (accuracy, precision,

recall, F1 score, etc.).

\end{itemize}

\subsection{Linear Regression}

just with correlation analysis

\subsection{Polynomial Regression}

basic variant, optimized hyperparameters

\subsection{SVM}

basic variant, optimized hyperparameters

\subsection{Neural Networks}

basic variant, optimized hyperparameters

\section{Improvements}

\subsection{Feature Selection}

\subsection{SVM}

\section{Evaluation}

evaluate models, use matrix she mentioned (confusion matrix)

Performance comparison between the models.

\begin{itemize}

\item Results in graphical or table formats.

\item Conclusions.

\item Problem complexity.

\end{itemize}

\section{Comparison to other work}

"Compare your solution with the works of other authors (published references) , try to propose

a better solution, e.g. improve the performance of the ML model in solving the problem you

work with."