Final Project Report

Project Title: Machine Learning Approaches for Heart Disease Classification

Raghda Kailany, Cameron Tinney, Fabliha Bushra, and **Md Omar Faruque**

1. Data:

1.1 What Kind of Data?

The dataset is related to a heart disease dataset. It contains information about patients' health and symptoms.

1.2. Dataset Information:

The dataset contains several attributes (features) and a target attribute:

● Number of Instances: The number of rows or instances is 920 in the dataset.

● Number of Attributes: The dataset has multiple attributes, including numerical and categorical ones.

● Type of Attributes: The attributes include features related to patients' health, such as age, blood pressure, cholesterol levels, exercise-induced angina, etc.

1.3. Data Processing:

The data is processed in several steps:

● EDA: We implemented various EDA techniques such as data summarization (i.e., mean, median, mode, variance, standard deviation, and range.), visualization ((i.e., heatmaps, scatter plots, boxplots), and statistical analysis i.e., correlation, Cross-tabulation etc.

● Handling Missing Values: Missing values in numeric columns are imputed using the mean strategy for the multiclass classifier. For binary classification, all the missing rows were deleted which gave a total of 299 samples out of the original around 920 samples. The Synthetic Minority Over-sampling Technique (SMOTE) is used for oversampling the minority class in the training set

● Outlier Detection and Treatment: Outliers in numerical features are detected using the Interquartile Range (IQR) method and replaced with the mean value. However, outlier removal had a negligible impact on the accuracy of matric.

● Categorical Feature Encoding: Categorical features are label-encoded.

● Normalization: Different normalization techniques have been tested i.e. StandardScaler, MinMaxScaler, RobustScaler.

● Multiclass Classification Model: We have built **12 multiclass classification models** and fine-tuned them using a random search algorithm to detect heart disease (where 0 indicates no presence of heart disease and 1, 2, 3, and 4 indicate different severity levels).

● Binary Classification Model: We also built **15 binary classifiers** that predict the presence or absence of heart disease (1 or 0) and compared their performance based on Accuracy, Precision, Recall, F1 score, AUC. We also fine-tuned the hyperparameters using grid search algorithm.

1.4. Attributes Usage:

The code drops specific attributes from the dataset for various reasons:

● High Missing Values: Columns with a high percentage of missing values (e.g., 'slope', 'ca', 'thal') are dropped after conducting correlation analysis with the target 'num' feature.

● Outliers: Outliers in numerical features are treated, but the features themselves are retained.

2. Data Mining Task:

● The primary data mining task in this analysis is **Classification**. We can treat it as a multiclass classification/binary classification problem.

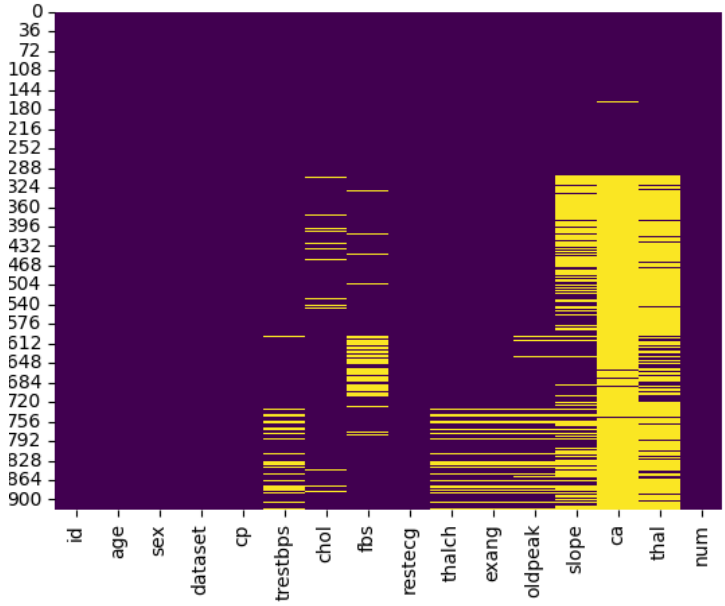
● For **multiclass classification problem**, the task is to classify patients into one of the five classes based on the severity of heart disease, which can be inferred from the "num" column.

● For **the binary classification problem**, the ML model predicts the presence or absence of heart disease (1 or 0) which can be inferred from the binary transformation of the 'num' column. The dataset is health-related data, and the data mining task is to predict the presence (or if present then up to what extent) or absence of certain health conditions based on the patients' attributes after preprocessing, feature selection, and outlier treatment. Specific attributes are dropped based on missing values and correlation with the target variable. The code prepares the dataset for machine learning by filling in missing values, handling outliers, and encoding features.

3. Results:

3.1 EDA and Data Preprocessing: This section will discuss the EDA and preprocessing techniques used in our project

3.1.1 Visualization of null values



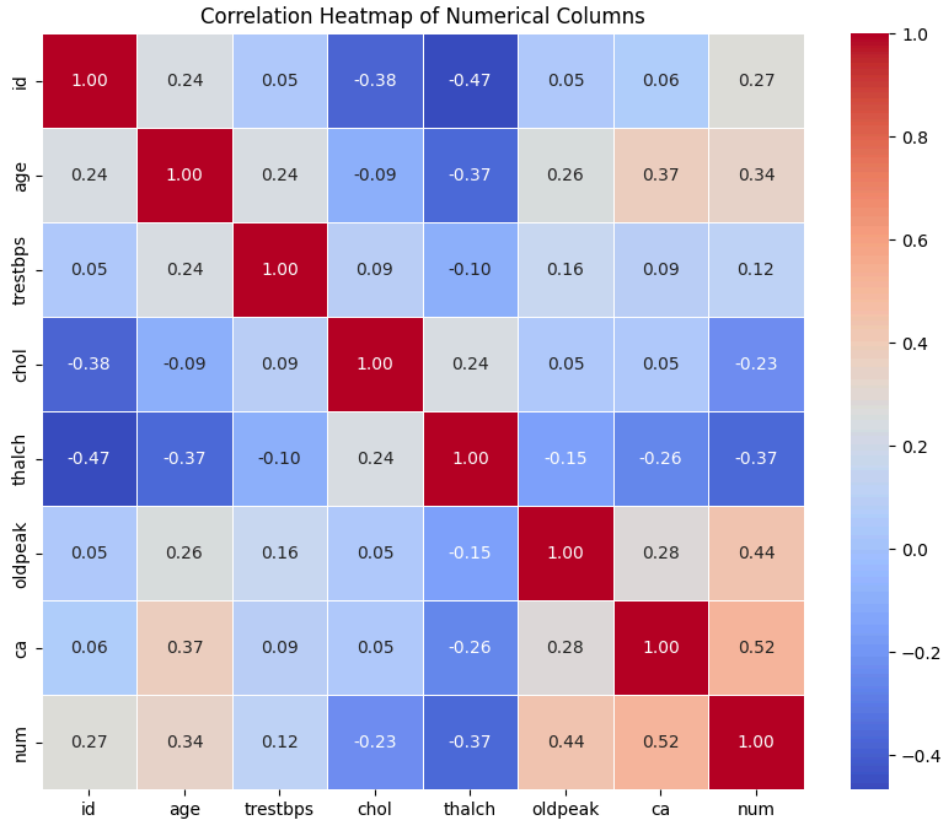
The above graph indicates several features contain missing values depicted using yellow color. Below is the percentage of missing data for numerical and categorical columns in the dataset

Variable	Missing%
Numerical Columns - Missing Value Percentages	
id	0.00%
age	0.00%
trestbps	6.41%
chol	3.26%
thalch	5.98%
oldpeak	6.74%
ca	66.41%
num	0.00%
Categorical Columns - Missing Value Percentages	
sex	0.00%
dataset	0.00%
cp	0.00%
fbs	9.78%
restecg	0.22%
exang	5.98%
slope	33.59%
thal	52.83%

Findings: The columns 'slope', 'ca', and 'thal' exhibit high missing values, exceeding the 30% threshold. Imputing missing values in these columns might introduce biases or inaccuracies, especially if the missingness is not entirely random. The challenge lies in accurately imputing values that reflect the true nature of the missing data.

3.1.2 Correlation Analysis:

Cor graph:

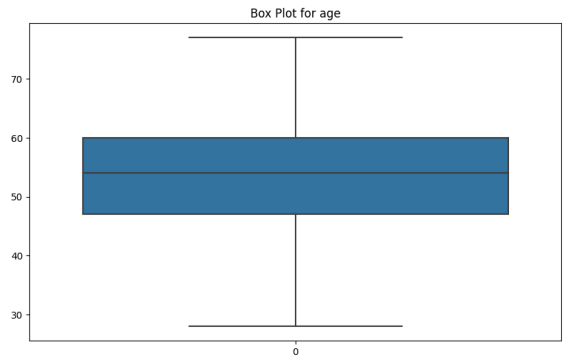


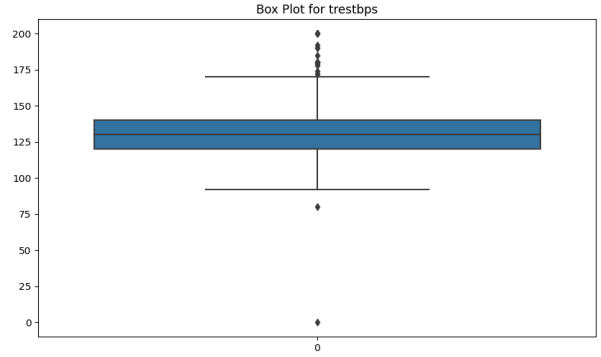
The heatmap indicates several notable correlations in the data:

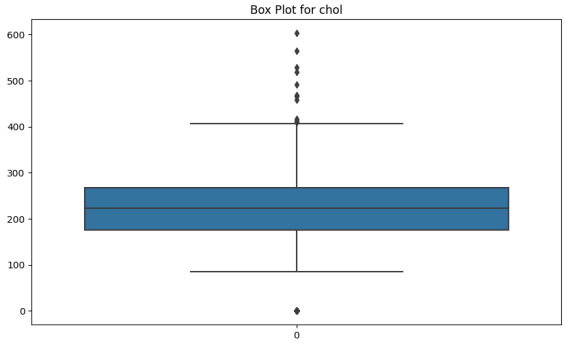
- "ca" and "num" appear to have a strong positive correlation since the color is a darker red. This suggests a high correlation, meaning as the number of major vessels ("ca") increases, the diagnosis of heart disease ("num") also tends to increase.
- "thalach" and "oldpeak" show a darker blue, indicating a strong negative correlation. This suggests that as the maximum heart rate achieved ("thalach") increases, the ST depression ("oldpeak") tends to decrease, or vice versa.
- Other variables, such as "chol" (cholesterol level) and "trestbps" (resting blood pressure), show much lighter shades, indicating a weaker correlation with the diagnosis of heart disease ("num").

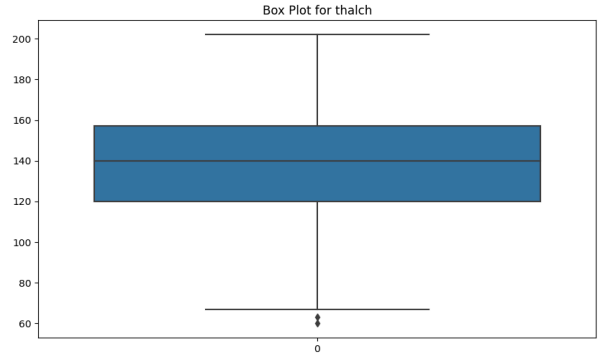
3.1.2 Outliers Analysis:

The box plots provided for each variable ("age," "trestbps," "chol," "thalach," "oldpeak," "ca," and "num") show the distribution of values in the dataset.







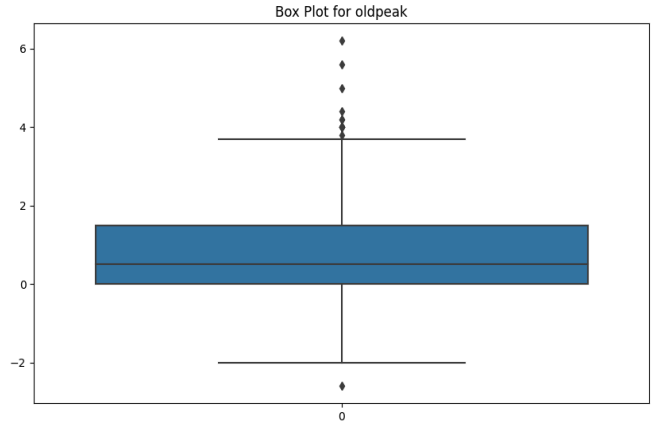


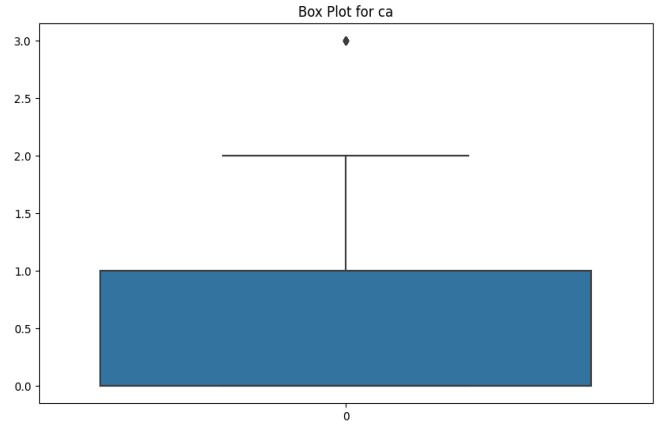
Age: The distribution of ages is fairly symmetrical, with the median age around 55. The range of ages is mostly between the late 40s to the mid-60s, with no outliers, which suggests a relatively consistent age distribution without extreme values.

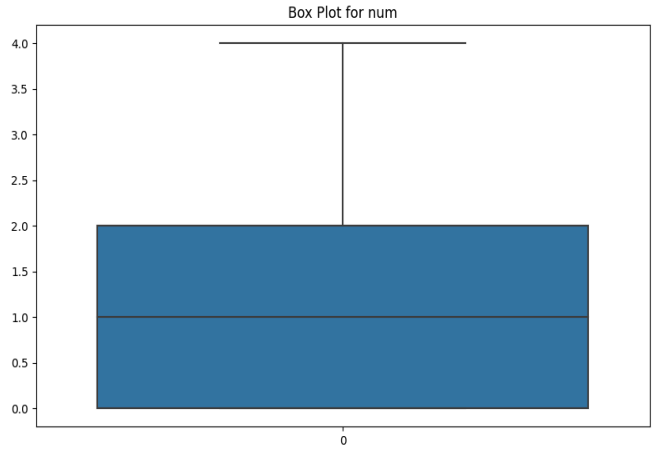
Resting Blood Pressure (trestbps): This distribution is slightly right-skewed, with a few outliers on the higher end, indicating that most people have a resting blood pressure around the median value, with some having significantly higher values.

Cholesterol (chol): Cholesterol levels are also right-skewed with several high outliers. The median is around 250 mg/dl, suggesting that half of the individuals have cholesterol levels below this value and half above. The presence of outliers indicates that some individuals have extremely high cholesterol levels compared to the rest.

Maximum Heart Rate Achieved (thalach): The heart rate distribution is left-skewed, with most individuals having higher maximum heart rates and a few outliers with unusually low values.







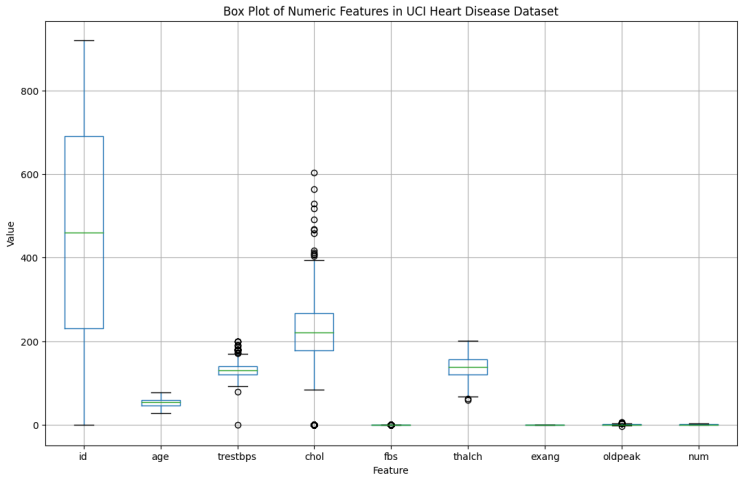
ST Depression Induced by Exercise Relative to Rest (oldpeak): This variable shows a right-skewed distribution, with a median close to 1, and some outliers with higher ST depression values, which could be of clinical significance.

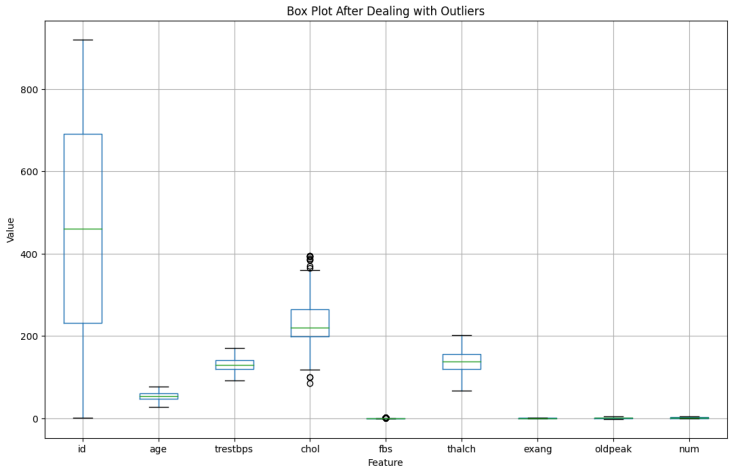
Number of Major Vessels Colored by Fluoroscopy (ca): Most individuals have 0 detectable major vessels, with a few having 1 or more. There are outliers with a high number of detectable vessels, which could indicate more severe cardiovascular issues.

Diagnosis of Heart Disease (num): The 'num' variable appears to have a distribution with a median of 0, suggesting that most individuals in the dataset do not have a diagnosis of heart disease, with a few individuals having higher values, indicating different levels or presence of heart disease.

Each box plot provides a five-number summary: the minimum, first quartile, median, third quartile, and maximum. Outliers are plotted as individual points. These summaries provide a quick visual insight into the central tendency, spread, and symmetry of the data distribution for each variable.

3.1.3 OutliersTreatment:





The graphs show two box plots for the same variables from the UCI Heart Disease Dataset, before and after dealing with outliers.

Before Dealing with Outliers:

- There are visible outliers in several variables, particularly in "chol" (cholesterol) and "trestbps" (resting blood pressure).
- The "id" variable's box plot is consistent, suggesting it may be an index with equal spacing between values.
- "age" has a tight interquartile range (IQR), centered around the 50s, with no outliers, suggesting that most individuals in the dataset are of middle age.
- "chol" has a large number of outliers, which indicates that there are several individuals with unusually high cholesterol levels.
- "thalach" (maximum heart rate achieved) shows a tight distribution with a few low outliers, while "oldpeak" (ST depression induced by exercise relative to rest) shows a skewed distribution with high outliers.

After Dealing with Outliers:

- The "chol" box plot has fewer outliers, indicating that the extreme values have been addressed.
- The "trestbps" also shows fewer outliers, suggesting a cleaner dataset which may provide more accurate insights.
- The scales of the plots for "thalach," "exang" (exercise-induced angina), "oldpeak," and "num" (diagnosis of heart disease) remain unchanged, indicating that no significant outliers were present or that they have been appropriately managed.

By handling outliers, the data becomes more normalized and potentially more useful for analysis, as outliers can sometimes skew the results of statistical models. It's important to note that the decision to remove or adjust outliers should be made carefully, considering the context and potential impact on the analysis.

3.2 Classification Section: This section depicts all the results for the binary and multiclass classifiers used in our project

3.2.1 Binary classifier models

Table 1: Performance matrices of various binary classifiers used in our project

Rank	Classifier	Accuracy	Precision	Recall	F1 Score	AUC
1	SVC	0.863	0.902	0.792	0.840	0.906
2	CalibratedClassifierCV	0.853	0.894	0.778	0.828	0.907
2	PassiveAggressiveClassifier	0.853	0.911	0.756	0.824	0.904
4	NuSVC	0.850	0.928	0.734	0.816	0.907
5	LogisticRegression	0.849	0.873	0.792	0.828	0.907
6	LinearSVC	0.846	0.884	0.770	0.820	0.905
7	LinearDiscriminantAnalysis	0.843	0.895	0.748	0.813	0.906
7	RidgeClassifierCV	0.843	0.889	0.756	0.814	0.905
7	RidgeClassifier	0.843	0.889	0.756	0.814	0.906
10	ExtraTreesClassifier	0.836	0.856	0.777	0.813	0.911
11	KNeighborsClassifier	0.833	0.849	0.777	0.810	0.890
12	RandomForestClassifier	0.829	0.861	0.762	0.805	0.904
13	XGBClassifier	0.803	0.828	0.749	0.777	0.891
14	BaggingClassifier	0.799	0.826	0.727	0.770	0.883
15	Perceptron	0.749	0.738	0.713	0.724	0.838

The table above lists the performance of various classifiers based on different metrics.

An interpretation of what these metrics mean is given below:

Rank: The classifiers are ranked based on one or more of the performance metrics, likely a combination or a specific important one like Accuracy or F1 Score. Here test accuracy is used for ranking.

Classifier: This column lists different types of statistical or machine learning classifiers that have been tested. For example, "SVC" stands for Support Vector Classifier, "NuSVC" is a similar type but allows for a nu parameter, and "LogisticRegression" is a classifier for binary outcomes.

Accuracy: This metric shows the overall correctness of the model, calculated as the number of correct predictions divided by the total number of predictions.

Precision: Precision is the ratio of true positives to the sum of true and false positives. It is a measure of the accuracy of the positive predictions.

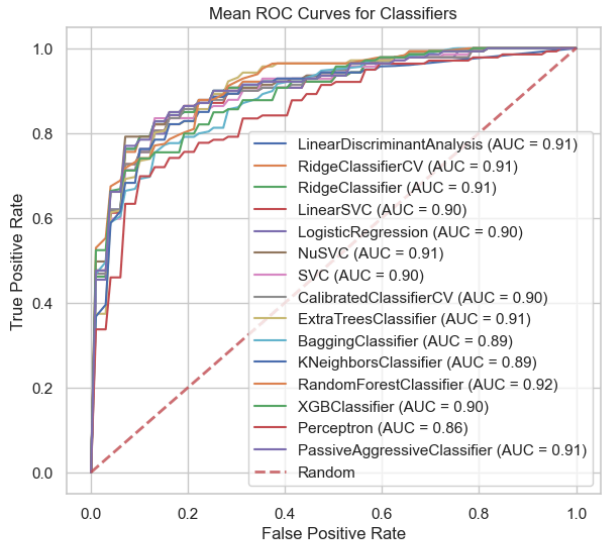
Recall: Also known as sensitivity, recall is the ratio of true positives to the sum of true positives and false negatives. It shows the ability of the classifier to find all the positive samples.

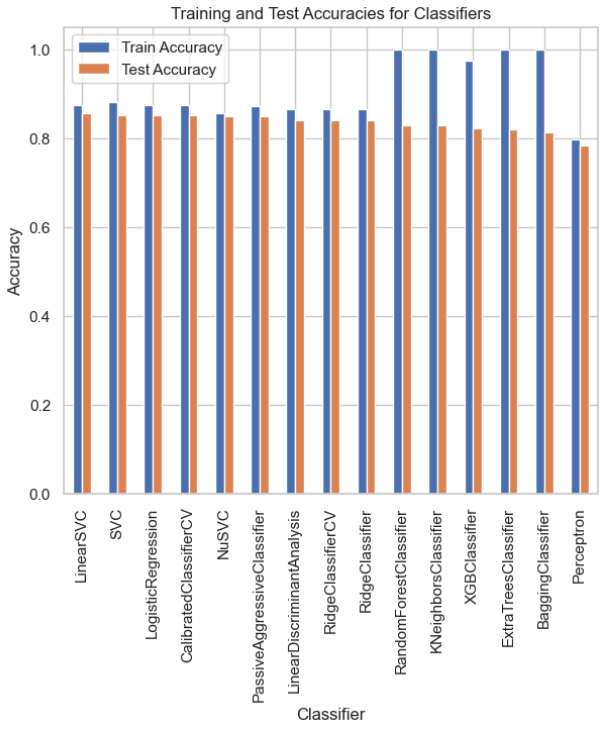
F1 Score: The F1 Score is the harmonic mean of precision and recall, providing a single score that balances the two metrics. It is especially useful when the class distribution is imbalanced.

AUC: The Area Under the Receiver Operating Characteristic Curve (ROC AUC) measures the ability of a classifier to distinguish between classes. A value of 0.5 suggests no discriminative ability, while a value of 1 suggests perfect discrimination.

Findings: From the table, the "**SVC**" classifier ranks first with the highest 5-fold cross-validation accuracy of around 0. 86 and a fairly high F1 score, suggesting it is the best-performing classifier among those tested according to the ranking criterion used. Each classifier has been evaluated on the same metrics, allowing for a comparison of their performance. T**o the best of our knowledge, our best model outperforms all the models submitted in Kaggle for this problem.** Even our 5th best model is better than the currently available solutions in Kaggle. We have achieved this by running an extensive grid search with a wide range of hyperparameters

Graph 1: ROC and Train/Test accuracy figures for the binary classifier.





The graphs provided above offer a visual comparison of different classifiers' performance using ROC curves and accuracy measurements.

Mean ROC Curves for Classifiers (Left Graph):

- The ROC curve graph displays the trade-off between sensitivity (True Positive Rate) and specificity (1 - False Positive Rate). Each line represents a classifier's performance across all possible threshold levels.
- The AUC values range from 0.86 to 0.92, with higher values indicating better overall performance. AUC values closer to 1 suggest that the classifier is better at distinguishing between the positive and negative classes.
- The Random Forest Classifier shows the highest AUC (0.92), suggesting that it has the best performance in terms of distinguishing between the classes for the given problem.
- A classifier's line closer to the top-left corner indicates higher sensitivity and specificity, meaning it is more capable of correctly classifying positive and negative cases.
- The dashed line represents the performance of a random classifier, with an AUC of 0.5. Any classifier performing close to this line is no better than random guessing.

Training and Test Accuracies for Classifiers (Right Graph):

- The bar chart compares the training and test accuracies of each classifier. Ideally, these should be high and close to each other to indicate a model that learns well and generalizes well.
- A large gap between training and test accuracy might indicate overfitting, where the model learns the training data too well, including noise and outliers, which do not generalize to new data.
- The classifiers in the graph mostly show a small gap between training and test accuracies, suggesting that they generalize well without significant overfitting.
- It's also important to note that while high accuracy is desirable, it is not the only measure of a good classifier, especially if the dataset is imbalanced. This is where the ROC and AUC metrics provide additional insights.

Findings: From these interpretations, one could conclude that the Random Forest Classifier is among the best performers for the problem at hand, with high AUC and closely matched training and test accuracies. However, for a definitive conclusion, one would also need to consider other factors such as the precision, recall, F1 score, and the context of the problem, such as class balance and the costs of different types of errors.

3.2.2 Multiclass classifier models

Table 2: Performance matrices of various multiclass classifiers used in our project

	Classifier	Accuracy	Precision	Recall	F1 Score	Best Hyperparameters
0	RandomForest	0.625	0.624611	0.625	0.6216	{'n_estimators': 400, 'min_samples_split': 2, 'min_samples_leaf': 1, 'max_features': 'log2', 'max_depth': None}
8	BaggingClassifier	0.592391	0.588054	0.592391	0.589234	{'n_estimators': 100}
7	ExtraTreesClassifier	0.576087	0.560362	0.576087	0.562428	{'n_estimators': 200, 'max_depth': None}
3	RidgeClassifierCV	0.559783	0.576095	0.559783	0.549315	{'alphas': (0.1, 1.0, 10.0)}
4	RidgeClassifier	0.559783	0.576095	0.559783	0.549315	{'alpha': 0.1}
11	PassiveAggressiveClassifier	0.559783	0.689811	0.559783	0.558325	{'C': 0.1}
9	KNeighborsClassifier	0.548913	0.569048	0.548913	0.555346	{'weights': 'distance', 'n_neighbors': 3}
6	CalibratedClassifierCV	0.527174	0.59473	0.527174	0.540753	{'method': 'isotonic'}
2	LinearDiscriminantAnalysis	0.51087	0.543398	0.51087	0.518364	{'solver': 'svd'}
5	LogisticRegression	0.51087	0.55464	0.51087	0.51473	{'solver': 'lbfgs', 'multi_class': 'ovr', 'C': 1.0}
1	DecisionTree	0.494565	0.478106	0.494565	0.483266	{'splitter': 'random', 'min_samples_split': 5, 'min_samples_leaf': 1, 'max_depth': None, 'criterion': 'gini'}
10	Perceptron	0.434783	0.473731	0.434783	0.374088	{'alpha': 0.0001}

Classifier: This column names the machine learning algorithms that have been used for classification.

Accuracy: This is the ratio of correctly predicted observations to the total observations. Higher accuracy means the model made more correct predictions. In the table, accuracy ranges from about 43% to 62.5%, with the Random Forest Classifier having the highest accuracy.

Precision: Precision is about how precise/accurate the model is out of those predicted positives, and how many of them are actually positive. Precision is a good measure to determine when the cost of a False Positive is high. The ExtraTreesClassifier has the highest precision in the table.

Recall: Recall calculates how many of the Actual Positives our model captures by labeling it as Positive (True Positive). High recall means most of the positive examples are correctly recognized (low false negatives). Random Forest and Bagging Classifier have the highest recall.

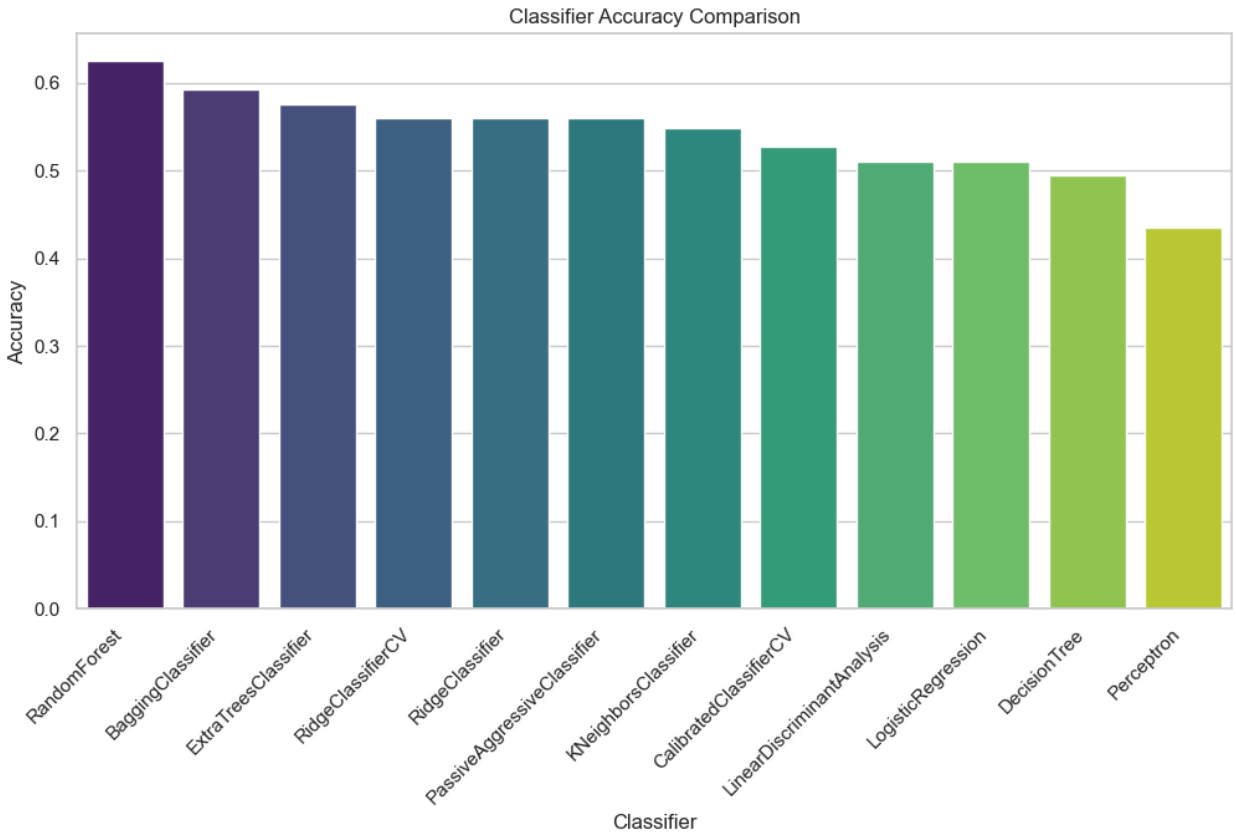
F1 Score: The F1 Score is the weighted average of Precision and Recall. Therefore, this score takes both false positives and false negatives into account. It is a good way to show that a classifier has a good value for both recall and precision. In this case, the Random Forest and ExtraTreesClassifier have the highest F1 scores.

Hyperparameters: This column shows the settings for each classifier. Hyperparameters are the configuration settings used to structure the machine learning model. These are set before the model runs and can affect the performance of the model. We used a random search to find the best hyperparameter for each model and listed them in Table 2.

Findings: From these results, the Random Forest Classifier seems to be the best-performing model with the highest accuracy and recall. However, the ExtraTreesClassifier has slightly better precision but lower accuracy and recall. The Bagging Classifier also shows good performance. The Perceptron, on the other hand, appears to be the weakest model with the lowest values across all metrics. It's important to note that the choice of model might depend not only on these metrics but also on the specific context of the problem, the computational resources, the interpretability of the model, and the type of data being used. Furthermore, if the dataset is imbalanced (i.e., more instances of one class than another), accuracy alone can be misleading, and one might have to look more closely at precision, recall, and the F1 score.

Compared to our binary classifiers the multiclass classifiers' 5-fold cross-validation accuracies are relatively low. We tried PCA, and various normalization techniques but could not improve the accuracies to attain a fair accuracy score of around 80%. Reasons for these low accuracy numbers could be a high percentage of missing data and class imbalances. We also could not apply feature engineering as it requires domain-specific knowledge which we did not possess.

Graph 2: Comparison of test accuracies of various multiclass classifiers



The bar chart compares the accuracy of various classifiers. Accuracy is a measure of how often the classifier makes the correct prediction, regardless of class.

Findings

Random Forest: This classifier has the highest accuracy among those listed, which is reflected by the tallest bar in the chart. It indicates that for this specific dataset, the Random Forest algorithm outperforms the others in making correct predictions

Bagging Classifier and Extra Trees Classifier: These classifiers have slightly lower accuracy than the Random Forest but still perform relatively well compared to the others.

Ridge Classifier CV through to Decision Tree: These classifiers show a gradual decrease in accuracy. The bars represent a spectrum of performance, with the Ridge Classifier CV being the best among this group and the Decision Tree being the worst.

Perceptron: This classifier has the lowest accuracy of all the classifiers shown. This could indicate that the Perceptron model is too simple to capture the complexities of the dataset or that it may need to be configured with the optimal parameters.

The different colors of the bars do not seem to signify anything other than making the chart easier to read; they help distinguish between the various classifiers.
From this visualization, it's evident that ensemble methods like Random Forest, Bagging, and Extra Trees tend to perform better on this dataset for accuracy. However, it's important to note that accuracy isn't the only metric to consider when evaluating classifiers. Depending on the application, other metrics like precision, recall, and the F1 score might be more relevant, especially if the dataset is imbalanced.