Red Wine Quality

Table of Contents

Introduction

In this machine learning assignment, we will investigate a dataset comprising data on several red wines and their quality ratings. We hope to use this data to create a model that can properly predict the quality of a red wine based on its chemical features. Understanding the aspects that contribute to the quality of a red wine allows us to assist winemakers in producing higher quality products and potentially benefit the wine industry as a whole. This project will involve pre-processing and analysing the data, training and evaluating various machine learning models, and ultimately choosing the best model for the task at hand.
The data set includes a number of features that describe the characteristics of the wine, such as its acidity, sugar content, and alcohol content. The idea is to use this data to anticipate the wine's quality, which is represented by a whole number on a scale from 0-10, but the data set only includes qualities from 3-8, hence there are only currently 6 quality cases.
Classification is a machine learning task in which a class label is predicted for a given input. In the instance of the red wine data set, the class label represents the wine's quality. Because quality is a discrete value rather than a continuous number, classification should be used to tackle this problem rather than regression.
Regression, on the other hand, is a machine learning task that involves predicting a continuous value for a given input. If the wine's quality was represented by a continuous value, such as a decimal number, rather than a discrete value, regression would be a better choice. Rather than attempting to predict a continuous value for the quality, we can anticipate which of the ten possible class labels (i.e. quality ratings) the wine belongs to by utilising classification to forecast the quality. This helps us to capture the distinct nature of the quality ratings more effectively and create more accurate forecasts.

Data And Exploratory Data Analysis

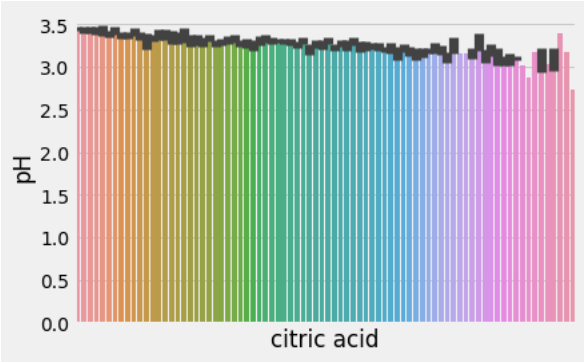
The initial stage in this process was to check the data for any missing or null values. We employed suitable functions or methods in the software or tool we were using for analysis and discovered that there were no null or missing values in the data, therefore no action was required to handle them. We then looked for outliers in the data. Outliers are data points that differ greatly from the rest of the data, and they can have a considerable impact on an analysis's results (David & Tukey, 1977). To detect outliers in the data, we employed a variety of techniques, including visualisation and statistical metrics. Again, we were relieved to see that there were no outliers in the data, thus no action was required to deal with them. Finally, we looked to see whether any of the data needed to be modified or scaled. Data might be skewed or have a non-uniform scale, which can have an impact on the outcomes of a study. However, in this situation, we discovered that the data was already useful and did not need to be modified or scaled (David & Tukey, 1977). Overall, we were successful in cleaning and pre-processing the data for this red wine dataset, since we encountered no missing values or outliers, and the data did not require any further modification or scaling. This allowed us to proceed with our analysis with confidence, knowing that the data was of high quality and ready for use.
We then performed exploratory data analysis to understand and summarise the red wine dataset's properties. This dataset contains a variety of features or variables that could be useful in determining wine quality. The summary statistics for each variable in the dataset were included in the table, including the number of observations (count), mean, standard deviation, minimum, 25th percentile (Q1), 50th percentile (median), 75th percentile (Q3), and maximum value.

	count	mean	...	75%	max
fixed acidity	1599.0	8.319637	...	9.200000	15.90000
volatile acidity	1599.0	0.527821	...	0.640000	1.58000
citric acid	1599.0	0.270976	...	0.420000	1.00000
residual sugar	1599.0	2.538806	...	2.600000	15.50000
chlorides	1599.0	0.087467	...	0.090000	0.61100
free sulfur dioxide	1599.0	15.874922	...	21.000000	72.00000
total sulfur dioxide	1599.0	46.467792	...	62.000000	289.00000
density	1599.0	0.996747	...	0.997835	1.00369
pH	1599.0	3.311113	...	3.400000	4.01000
sulphates	1599.0	0.658149	...	0.730000	2.00000
alcohol	1599.0	10.422983	...	11.100000	14.90000
quality	1599.0	5.636023	...	6.000000	8.00000

alcohol	0.476166
sulphates	0.251397
citric acid	0.226373
fixed acidity	0.124052
residual sugar	0.013732
free sulfur dioxide	-0.050656
pH	-0.057731
chlorides	-0.128907
density	-0.174919
total sulfur dioxide	-0.185100
volatile acidity	-0.390558

(All data analysis techniques can be found in data-testing.py)

We could see from these summary statistics that the mean value for each variable was generally near to the median, indicating that the data was not excessively skewed in any direction (Aggarwal, 2016). The standard deviation was smaller for variables such as fixed acidity, citric acid, and residual sugar, but larger for variables such as volatile acidity and chlorides. This suggested that the latter variables might be more variable. We could also observe that most variables had a large range of values, with minimum and maximum values that were pretty far apart. For example, the least fixed acidity value was 4.6 and the maximum was 15.9, whereas the minimum volatile acidity value was 0.12 and the maximum was 1.58. This revealed that there could be considerable variances in the chemical makeup and sensory features of the wines in the sample.
Then, using a red wine dataset, we looked at the relationship between various parameters and wine quality. The correlation values for each factor were shown in the table above. Alcohol had the strongest positive link with wine quality, according to the table, with a value of 0.47. This suggested that as the alcohol percentage of the wine increased, so did the wine's quality. With a value of 0.25, sulphates also had a favourable link with wine quality. Citric acid had a positive association as well; however, it was weaker than alcohol or sulphates. Several factors, on the other hand, demonstrated a negative link with wine quality. With a score of -0.39, volatile acidity exhibited the highest negative link. This suggested that as the amount of volatile acidity in the wine increased the quality of the wine decreased. Wine quality was also negatively correlated with chlorides, density, and total sulphur dioxide. PH seemed to have little effect on the quality of the wine, with a score of -0.058; this is interesting because, as previously mentioned, citric acid has a fairly high impact on the quality of the wine. In wine making one of the main roles of citric acid is to lower the pH. This is backed up by our data as, when more citric acid is added, the pH reduces.



The problem with this is that, as mentioned earlier, pH seems to have very little effect on the overall quality of the wine. This suggests that citric acid is not only altering the pH, but also affecting the overall taste of the wine; with the later having more of an impact on the overall quality of the wine. Overall, the summary statistics assisted us in understanding the major aspects of the red wine dataset and identifying trends or patterns that may be useful in forecasting wine quality using classification machine learning. Alcohol and sulphates appeared to be the most important components in predicting wine quality, according to the table, with both having a positive corelation with quality; volatile acidity was also an important component, having a negative corelation with quality. Residual sugar had a less significant impact, with it seemingly having little to no impact on the quality of the wine. These findings could aid in the development of a machine

learning model to predict wine quality based on the most essential criteria, aiding in creating better tasting red wines.

Machine Learning Models We Explored:

XGBOOST

Methodology: Xgboost was another model we looked at. Some of the advantages of xgboost include its scalability, capacity to handle imbalanced datasets, and automatic feature interaction learning, which can be useful in the context of wine data where there may be a huge number of samples and features and quality values that are not uniformly distributed (Chen & Guestrin, 2016). Xgboost has also demonstrated exceptional performance on a range of tasks, including wine quality prediction.
However, there are some possible drawbacks to xgboost, including as its complexity, which can make it more difficult to build and tweak than other algorithms. Furthermore, xgboost is widely regarded as a "black box" model, which means that it is difficult to determine the specific elements influencing its predictions (Chen & Guestrin, 2016). This can be a disadvantage in situations when interpretability is critical. The performance of Xgboost can also be affected by hyperparameter tuning, which can take a significant amount of time and effort. To prepare the data for xgboost, we used a LabelEncoder from scikit-learn to encode the target variable "quality" and split the data into features and the target variable. We then scaled the data using StandardScaler and split it into a training set and a test set using train test split. We used the fit function to train the xgboost model on the scaled training set and the predict function to evaluate its performance on the scaled test set. We utilised scikit-classification report and accuracy score functions to assess the model's precision, recall, f1-score, and overall accuracy.
To summarise, we chose xgboost because of its capacity to handle huge and imbalanced datasets and learn feature interactions automatically. The use of a train/test split for model evaluation reduced overfitting and offered a more accurate assessment of the model's generalisation capabilities. Overall, the xgboost model performed decently in predicting the quality of red wines based on chemical attributes. However, in situations where interpretability or convenience of implementation are more essential concerns, xgboost may not be the ideal choice.

Results: The wine's quality is a discrete value ranging from 0 to 5. We achieved the following evaluation metrics after training and testing the model: Precision is the percentage of true predictions made by the model. For example, a precision of 0.71 for class 2 suggests that the model correctly predicted 71% of the time. Remember, this is the percentage of actual instances of a class correctly predicted by the model. The harmonic mean of precision and recall is represented by the F1-score. It measures the balance of precision and memory and is often regarded as a superior overall statistic than either precision or recall alone. in this case was 0.75 for class 2 meaning the model has a relatively high level of precision and recall, but for other classes, for instance, class 5 it was 0.29 meaning it has a low level of precision. Support is the number of instances of each class in the test set. The model's overall accuracy was 68%. The macro average was the unweighted average of each class's precision, recall, and f1-score. The weighted average considered the level of support for each class, giving more weight to classes with more instances. We discovered that the model's performance for specific classes (such as class 0 and class 5) was not very excellent, as evidenced by low precision, recall, and f1-score. This could be due to several variables, including a small number of occurrences in the test set or a lack of data to learn patterns for these classes. We may want to explore tweaking the model or gathering more data to enhance the results for these classes.

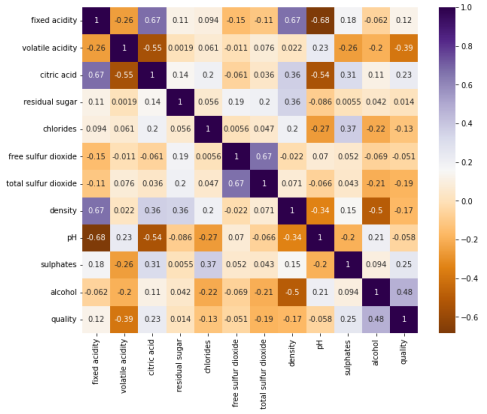
Support Vector Classifier (SVC)

Methodology: To begin, we imported the data into a Pandas DataFrame and divided it into features (X) and the target variable (y). The data is then split into a training set and a test set using the train_test_split function from scikit-learn. Before training the model, the data was pre-processed with StandardScaler to scale the features. Scaling the data can increase model performance because SVC models are sensitive to the scale of the input features. Following that, the SVC model was set up with a "rbf" kernel and a regularisation parameter (C) of 3. The fit function was then used to train the model on the scaled training set (Géron, 2023). The predict function was used to examine the model's performance on the test set, and the classification report and accuracy score functions from scikit-learn are used to measure the model's precision, recall, f1-score, and overall accuracy.
SVC has the advantage of being able to represent complex, non-linear relationships between features and the target variable due to the usage of kernels. SVC is also a suitable choice when there are a lot of features because it is more resistant to the "curse of dimensionality" than other models. However, SVC can be sensitive to the kernel and regularisation parameters used, and determining the best values can take some trial and error. Furthermore, SVC models can be computationally expensive to train and forecast, especially for large datasets (Géron, 2023).

Results: The algorithm properly classified the quality of 63% of the wines in the test set, which is the first thing we can notice. Looking at the individual quality classes, the model fared best for class 5 wines, with precision, recall, and f1-scores of 0.66, 0.76, and 0.71, respectively. This indicates that the model is reasonably effective at detecting class 5 wines. Similarly, for class 6 wines, the model performed reasonably well, with precision, recall, and f1-scores of 0.62, 0.66, and 0.64, respectively. The model, on the other hand, struggled to categorise wines in classes 3 and 4, with precision, recall, and f1-scores of 0.00 for each class. This shows that the model may struggle to differentiate between these groups and could benefit from further training data or further adjustment of its hyperparameters. Overall, these findings indicate that the SVC model is reasonably good in classifying red wine characteristics, although there is still space for improvement, particularly in the lower quality groups.

Decision tree

Decision Trees are one of the most widely used methods for representing classifiers. Which are considered human-readable. Decision trees are a type of supervised machine learning that can be used for both classification and regression problems. It has a hierarchical, tree structure, where internal nodes represent dataset features, branches represent decision rules, and each leaf node represents the output of the tree.
The advantages of using a decision tree over other algorithms are that it’s easy to interpret the decision, the Boolean logic, and visual representations of it are easier to understand. And the hierarchical structure of a decision tree makes it simple to determine which attributes are most important. Decision trees are more flexible due to the number of characteristics they have, which makes it easier to deal with various data types and to automatically handle values with missing values. Decision trees necessarily require less data preparation during the pre-processing stage. And the training process of decision trees is much faster than some other models. Even though decision trees are simple to understand, they have some limitations that prevent them from being used on large datasets. One of the main drawbacks of the decision trees is overfitting, complex decision trees are prone to overfitting and do not scale well to new data, which can lead to poor decisions and inaccuracy results.
Methodology: Import all the necessary packages and load the win equality dataset into data frame, Split the dataset into training and testing data and then build the model (decision tree) and fit the training data. Must calculate the accuracy and calculate the number of nodes in the tree. Then try to increase Accuracy of the model by hyper parameter tuning, create a Parameter grid and checked for best parameters and score. Finally test for Decision Tree and calculate the accuracy.



Result: From the heatmap we can see that citric acid" and "fixed acidity" have a correlation coefficient of 0.67. Therefore, both labels are 67% relevant. And the correlation coefficient between "fixed acidity" and "ph" is -0.68. Therefore, both have a 68% inverse relevance. After splitting and training the data on Decision tree, it tends to be 61% accurate. but after implementing the Hyperparameter tuning the accuracy get decreased by 4% to 57%. According to the results from the decision tree, "alcohol" feature tend to be very beneficial for prediction. Moreover the "citric acid" feature is being the least useful for prediction.

Random Forest

Random forests are a type of machine learning model that can be used to predict a categorical or continuous target variable based on a set of input features. They are particularly well-suited for classification tasks, where the goal is to predict the class label of an observation based on its features. Random forest is an ensemble learning algorithm. This means it uses several classifiers together and combines them for a final prediction. The random forest classifier is trained using a training set of data. The classifier is initialized with certain hyperparameters, such as the number of decision trees in the forest and the class weight (n_estimators and class_weight in our code). The class weight determines the importance of each class in the training set, with "balanced" meaning that the classifier will attempt to balance the weight of each class. Finally, the code uses the trained random forest classifier to make predictions on the testing set (using the predict method in code). The predictions are then evaluated using the classification_report and accuracy_score functions from scikit-learn, which provide the overall accuracy of the predictions as well as a summary of the model's performance in terms of precision, recall, f1-score and support.

- Precision: Precision is the proportion of true positive predictions made by the model to all positive predictions.
- Recall: Recall is the proportion of true positive predictions made by the model to all actual positive cases.
- F1 score: The F1 score is the harmonic mean of precision and recall. It is a balance between the two and is generally used as a metric for evaluating the performance of a classifier.
- Support: Support is the number of samples of the true response that lie in that class.

In the context of the red wine quality dataset, a random forest model could be used to predict the quality rating of a red wine based on its various attributes, such as pH, alcohol content, and density. This could be useful for wine producers, who may want to predict the quality of their wines before they are bottled and sold. Looking at the classification report, we can see that the model performs relatively well for the classes 5 and 6, with F1 scores of 0.75 and 0.72, respectively. However, the model performs poorly for the other classes, with F1 scores ranging from 0.10 to 0.51. This suggests that the model may not be very effective at predicting the quality of wines that fall outside of the 5 and 6 range. It also has very low precision and recall scores for the class 3, and very low support for the class 8. This is likely caused by there are very few samples in these classes, which makes it difficult for the model to learn to predict them accurately.

Neural Networks

Another observed model we approached was Neural Networks. This is primarily due to its usefulness with classification problems by training a Multi-Layer Perceptron (MLP) with a backpropogation algorithm, applying thresholding using a continuously differentiable function to ultimately create a network that can represent the wine rating classifications by patterns observed in the training data, which is far more inclusive and efficient than tuning perceptrons singularly. An advantage of neural networks is shown through the fine-comb features that a MLP possesses, able to tweak the parameters and utilise functions such as early stopping, dropout and regularisation. Contextualised with the red wine dataset, the format of the y column (the wine quality score) was already in a numerical form, meaning the column's data did not have to be encoded to match a numerically analysable form.
The challenge that is raised when handling neural networks is to train them as such that it best fits the training data, and ultimately minimising a measure of error. This delves into the importance of backpropogation algorithms using a gradient descent approach to find close to an optimum accuracy neural networking model, working backwards through a network performing calculations using inputs to update the weights using the value calculated for the delta rule to reduce the output layer's error by handling the hidden layer's said weights. An important role in tackling backpropogation is to find a sensible convergence criterion using the stopping criteria to converge the Euclidean norm of the gradient vector to some sufficiently small threshold of the gradient (*Kramer and Sangiovanni-Vincentelli,1989*)
Backpropogation, whilst having a simple and robust implementation, working to some degree for all types of classification problems when implemented correctly, is not without its faults or limitations. These would include the primary issue of local minima, which demonstrates that getting stuck in local minima prevents the model from utilising the optimal accuracy found by utilising the global minimum. However, with this issue aside, the neural networks model was chosen in theory due to its effective handling of multi-class classification problems, which would serve us well with the range of possible scores the red wine model presents.

Results: As observed in our creation of a MLP handled using the lbfgs solver (which is more appropriate for smaller datasets), the precision varied heavily depending on the quantity of data observed for each type of class. For wines rated 5, the model had a precision rating of 0.62, which means that it could correctly predict class 5 wines at an accuracy of 62%. Observing the F1-score, which entails the harmonic mean of precision and recall, often giving a more statistically accurate representation of performance than handling precision or recall, class 5 wines had an accuracy of 0.65. The more shocking observation to note is the polarity between classes predicted in the model, with class 3,4 and 8 having a F1-score of 0. This meant that those classes had no predicted samples at all, with the model only referring to classes 5,6 and 7 in its predictions. This heavily influenced the total accuracy of the model which lies at 0.5925, or 59% when rounded. This is due to the model 'playing favourites' in terms of the classes it chose to predict. This problem could be due to a multitude of factors, such as whether the optimum tuning for the MLP was reached, whether the quantity of data was not enough for the model to accurately predict for data that has so many class possibilities, or whether the data had so many variables that it is extremely hard to pinpoint precisely what patterns of the data correspond to what typical rating value a red wine receives. Analysing the Support, which is the number each class appears in the test set, gives us a different insight, as the classes with the highest accuracies had over 150 instances, whilst those with no predictions whatsoever only had less than 20 instances, with one class only having a single instance whatsoever. This leads me to believe that the problem may lie within the rating of the wines themselves, as it demonstrates that wines with ratings such as 3 or 8 make much lower appearances than wines with ratings of 5, therefore potentially skewing the data as such that it has not a lot of instances of certain classes to be able to correctly analyse patterns within them.

Other Models

We also constructed a logistic regression and a k-nearest neighbour (kNN) machine learning model for our dataset in our project. However, after testing, we discovered that the accuracy of these models was less than 60%. We decided against completely analysing these models since we wanted to guarantee that the models we utilised were accurate. We opted to focus on models with an accuracy of around 60% and more in order to adequately analyse which model would be the greatest fit for our purposes. While the logistic regression and kNN models had various advantages, their lesser accuracy made them unsuitable for our needs. Another approach taken was the construction of a Naïve Bayes system using Gaussian Naïve Bayes due to the continuous data used in the dataset. This led to, on average, an accuracy of around 55%, therefore was deemed unsuitable for complete analysis due to the low accuracy rate. An interesting insight into this model, however, was observing the f1-scores, with an f1 score for wines rated 3 and 8 being complete 0. This gave us some further knowledge on the effect of the quantity of variables and the quantity of observable classes on the accuracy of our models, as previously testing Naïve Bayes on datasets with less variables and classes led to a significantly higher accuracy rate.

Evaluation

In general, the XGBoost model has the best overall performance on this dataset, followed by the SVC model. The other models have relatively lower accuracies and f1-scores. The precision and recall values for each class vary widely among the models, indicating that some models may be better at predicting certain classes than others. This is likely down to the lack of data in certain classes. We can see that the numbers in classes 3, 4 and 8 are very low (1, 18 and 4, respectively), which contributes to the poor performance of the models on these classes. This is likely because when a wine is submitted for tasting, it's quite likely to not taste awful, which would put it in the lower categories, as the person making it would not submit the wine; it is also likely that it's not going to be the best wine ever, which would put it in the higher categories. Looking at the data, a bell curve forms, with its peak being around categories 5 and 6.

The XGBoost model had the highest overall accuracy, with a value of 0.68. This means that out of all the samples in the dataset, the model was able to correctly predict the quality of 68% of them. The XGBoost model also had the highest f1-score for the class with labels 2 and 3, with a value of 0.75 and 0.69 respectively. This means that the model was able to achieve a good balance between precision and recall for these classes. However, the XGBoost model had very low precision and recall values for the class with labels 0, 1 and 5. This indicates that the model was not very effective at predicting this class, again, likely due to the lack of data in these classes (this is a common trend among all of the models).

The random forest model had a slightly lower overall accuracy, with a value of 0.69, which is still more than acceptable. The random forest model had a relatively high precision and recall; for example, the class with label 5 had a precision and recall value of 0.71 and 0.78, respectively. This means that the model was able to achieve a good balance between precision and recall. the model was able to achieve a good balance between precision and recall indicating the model was quite good at identifying a high proportion of the positive instances in the dataset.

The SVC model had a relatively high overall accuracy, with a value of 0.63. While this score is lower than that of the XGBoost model and random forest, it still means that out of all the samples in the dataset, the model was able to correctly predict the quality of 63% of them, which is higher than the rest of the models and still acceptable.

The decision tree model had a lower overall accuracy, with a value of 0.6. given the other models have a much higher accuracy this model is slightly less applicable than the previous ones. With an f1-score of 0.59 for category 6 it also massively underperforms other models in this category.

The neural network model had a slightly lower overall accuracy, with a value of 0.59. This means that out of all the samples in the dataset, the model was able to correctly predict the quality of 59% of them. With its highest f1-score being 0.65 it also consistently

underperforms against other models. Its poor performance could be down to a couple of reasons. One of these reasons could be insufficient data. A neural network model requires a large amount of training data to learn effectively. If the dataset is too small or lacks diversity, the model may not be able to learn the underlying patterns in the data, resulting in poor performance. Another reason for the poor performance could be overfitting. This occurs when a model is too complex for the given dataset and begins to memorize the training data rather than generalize to new data. This can result in poor performance on unseen data.

Conclusion

Lesson Learned

One of the primary lessons learnt throughout this project was the importance of the dataset being handled, and how the dataset is structured to provide the most accurate models possible. Throughout our models, the highest general accuracy score found was 68%, which raised several questions as to what features are limiting the capabilities of our models within the dataset. These demonstrate the severity of having a dataset that contains a large sample of each class to be analysed, as our dataset included many points of data with a rating of 5 or 6 and very few with boundary scores such as 3 and 8. This leads our models to not only make less predictions for those scores due to not having enough patterns within the scores to analyse, but to also wrongly categorise the lesser observable classes as the classes with higher quantities of observable data. This also highlights the complexity of having many variables within the data, which leads each class to have more struggles with finding cohesive patterns to map to a specific class. Due to us correctly having the data formatted and structured with no null data points and no need for reformatting, this highlights the severity of the class quantity on the impact of machine learning models. Another primary lesson learnt emphasises that not all models may be suitable for specific datasets, even if all models are built for handling classification problems. This was shown as even though all models used theoretically are built to solve classification problems, and the data was the same used every time, the model’s response to the data varied heavily throughout the process, showing the importance of testing through various means of AI modelling when handling a problem, and that it's not ‘one model fits all’ for a type of problem being solved. These primary lessons learnt demonstrate both the importance of the data and the actual modelling of the data are both unique problems, furthermore the severity of testing not only upon a range of models, but within the parameters of a model to attempt to find the most appropriate model with the closest accuracy is a complex problem necessary to tackle any classification or regression task.

Future Work:

In the future, we intend to address various difficulties we came across during our project to increase the performance of our machine learning models. First, we'll turn our dataset's classes into a two-category problem. Because previous research has indicated that this can result in better results, for example, we will designate classes 3, 4, and 5 as 0 and classes 6, 7, and 8 as 1 for our wine data set. This will allow us to deploy machine learning algorithms more readily and create more accurate forecasts. Following that, we will modify the hyperparameters of each model to discover the greatest combination of settings that will result in the best performance. To find the ideal hyperparameters for each model, we'll employ approaches like grid search, random search, and Bayesian optimization. Finally, we will use oversampling or under sampling approaches to address the issue of class imbalance in our dataset. In order to balance the distribution of classes in our dataset, we will either generate new samples of the underrepresented class or remove samples from the overrepresented class. We believe that tackling these challenges will allow us to considerably increase the performance of our machine learning models and obtain more accurate forecasts.

	precision	recall	f1-score	support
3	0.00	0.00	0.00	1
4	0.00	0.00	0.00	18
5	0.66	0.76	0.71	156
6	0.62	0.66	0.64	174
7	0.53	0.38	0.44	47
8	0.00	0.00	0.00	4
accuracy			0.63	400
macro avg	0.30	0.30	0.30	400
weighted avg	0.59	0.63	0.61	400

0.63

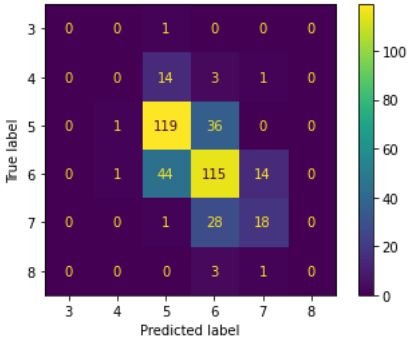
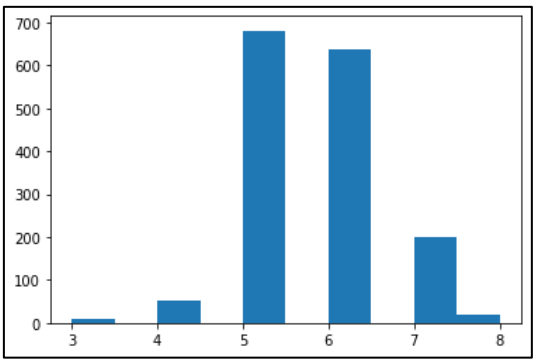


Fig 1. Classification report for SVC model Fig 2. Confusion Matrix, SVC Model



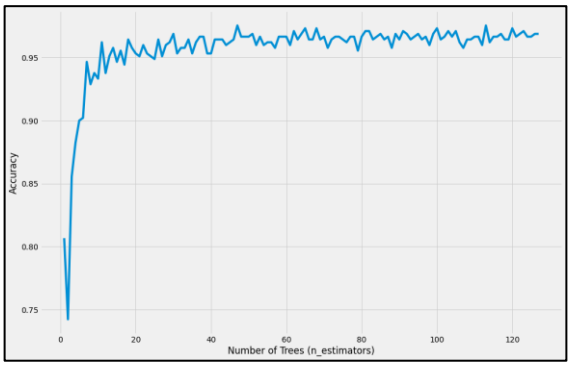


Fig 3. Histogram for Decision tree model Fig 4. Plot, Random Forest Model

References

David, F.N. and Tukey, J.W. (1977) "Exploratory Data Analysis," *Biometrics*, 33(4), p. 768. Available at: https://doi.org/10.2307/2529486.

Aggarwal, C.C. (2016) "Supervised outlier detection," *Outlier Analysis*, pp. 219–248. Available at: https://doi.org/10.1007/978-3-319-47578-3_7.

Chen, T. and Guestrin, C. (2016) "XGBoost," *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* [Preprint]. Available at: https://doi.org/10.1145/2939672.2939785.

XGBoost parametersÁ (no date) *XGBoost Parameters - xgboost 1.7.2 documentation*. Available at: https://xgboost.readthedocs.io/en/stable/parameter.html (Accessed: December 21, 2022).

Géron Aurélien (2023) *Hands-on machine learning with scikit-learn, keras and tensorflow: Concepts, tools, and techniques to build Intelligent Systems*. Sebastapol, CA: O'Reilly.

Kramer, A.H. and Sangiovanni-Vincentelli, A. (1989) *Optimization techniques for Neural Networks*. Available at: https://www2.eecs.berkeley.edu/Pubs/TechRpts/1989/1146.html

udita3996. "Eda, Logistic Regression & Decision Tree." *Kaggle*, Kaggle, 13 July 2020, www.kaggle.com/code/udita3996/eda-logistic-regression-decision-tree/data.

"What Is a Decision Tree." *IBM*, www.ibm.com/uk-en/topics/decision-trees.