Predicting vinho verde’s wine variants quality, deploying several machine learning classification techniques, and predicting wine’s variants

Luís Henriques, up202204386
Paulo Portela, up202200871

Abstract

This work aims to identify a data analytics problem that can be addressed by the available data in the Dataset selected: “Wines”, which is the merge of two datasets related to red and white variants of the "Vinho Verde" wine. An analysis of 6496 wines was made. To do this, the project team proceeded according to the CRISP-DM methodology to:

- Predict if each wine sample is a red or white wine;
- Mainly to predict the quality of each wine sample, which can be low, medium, or high.

The analysis and development of the project was based on several predictive attributes, like fixed acidity, density, pH, and alcohol. Techniques like PCA and z-score outlier detection were applied to prepare the final dataset

As our final output, it's expected to have several classification techniques - that shows how well the quality and type of the wine were predicted for the low ,medium and high quality. For those, the respective accuracy, precision, recall, F1-score, AUC and execution time were computed. An accuracy of 99.39% and an AUC of 98.80% were obtained for the prediction of the wine variants (red and white). To classify the wine’s quality, in general, simple models weren’t able to capture the nuances of variation in the data, especially of low quality wines with so few records in comparison to the other classes. On the other hand, good results were obtained in the major families of the ensemble methods. Alongside the ensemble algorithms, a developed neural network presents high values for both balanced and unbalanced data. In the top four models obtained, two were with RF, one with MLP neural network, and the other using the KNN algorithm. The best model got an accuracy of 77.9%, precision of 82.4%, recall of 77.9%, f1-score of 79.2%, and an AUC of 76.7%. The running time of this model was 7 minutes and 36 seconds.

Notebook with the python code should be run in Google Colaboratory.

1. Introduction

Data is in everything we see, touch and do. Data is produced by us, society, and we are also the targets of the data we created. We are targets of that data because it is the basis of who we are and of what we build, and that can help other entities understand what are, and could be, our future needs. If we go to a retail store and we constantly buy the same set of items, if other people have the same behavior but buy something else, then it could be interesting for us to buy that other item, and also for the stores because they increase liquidity by reducing their stock. Another example could be to use previous data about a factory's shop floor machines failures to understand and prevent those events from occurring by predicting in which conditions does that happen.

But, what is data? Data could be numbers, addresses, the color of several clothes, people's names, stock prices. Data consists of bits that create numbers, words, and so on, so that when we add information about what they mean, we can manipulate it in order to extract knowledge from them [1].

This report focuses on the basics of data mining and machine learning, meaning that in this report the main goal is to apply the theoretical concepts learned in the course IACEC, Introduction to Machine Learning and Data Mining in portuguese, in practice, applying several machine learning techniques in order to extract knowledge from a dataset. In order words, we want to do data mining to a dataset in order to possibly predict a certain characteristic of an object or entity. Data Mining is nothing more than, from a prepared dataset, learning about a certain set of objects, for example patterns in customers transactions in a retail store, to predict future objects.

2. Methodology

The dataset retrieved was, firstly, separated in two datasets, one for red and the other for white wines, where the first one had 1 599 instances, and the second dataset had

4 989 instances. For the sake of this report, objects and instances mean the same, the rows of the dataset.

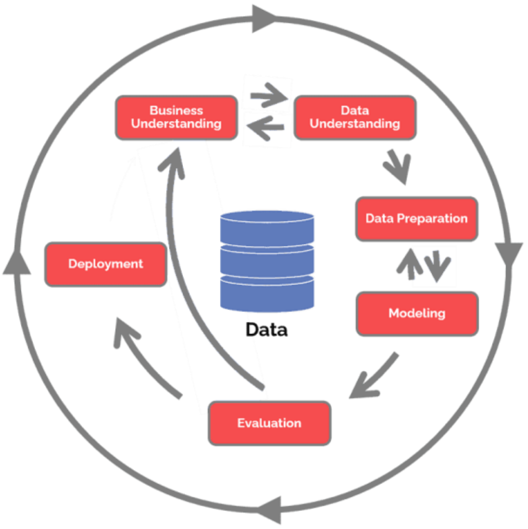

Figure1 - Crisp-DM Methodology

Regarding the methodology applied in this project, the one chosen was the methodology most used in data mining projects and applications, that is, the CRISP-DM Methodology, Cross-Industry Standard Process for Data Mining. There are other methodologies that could be applied here, like the KDD process. CRISP-DM is a six-step methodology that, although it has its steps well defined, the fact is that this methodology could be applied throughout months, even years, and that makes it a possible perpetual process [1]. Then, the steps in the CRISP-DM methodology will be mentioned, explained, and also contextualized in the work carried out throughout the development of this project (Figure1). These steps will be different sections, where some standard sections in a paper, like Results and Conclusions, will be done within specific steps of those sections.

3. Business understanding

In the first step of the CRISP-DM methodology, the focus is in understanding, from the client's or business point of view, what the project should achieve, as well as the problems and limitations in that domain, as well [1].

Because the datasets were retrieved from a dataset website, the project team had to put its shoes in the place of the client, and think about what could be the problem that businesses could be interested in being resolved. In this case, the dataset was wine ratings of two variants of the portuguese wine called “Vinho Verde”.

So, the problem could come from a company that produced wines in the north of Portugal, for example in Porto. This company could be in the process of establishing itself as a market leader in Portugal, however it couldn’t achieve that position because they couldn’t find a way to understand what was considered a bad, average, and good wine, considering its market’s opinions.

In order to do that, the client wanted to know which wines had more chances of having a low, medium, or high rating from their customers, depending on different characteristics. Not only the goal of the client is to know what are the wine’s ratings depending on the characteristics, but also they wanted to know beforehand what would be the quality of a certain wine, given some initial conditions (characteristics). From a data analytics point of view, the goal was to develop the team’s skills by developing several models, using several classification techniques, to predict the wine’s ratings, and then understand what predictive variables impacted more the results.

Because the team selected to do that didn’t have almost any experience with data mining, the client simply demanded some sort of model(s) that could predict the wine’s rating better than a coin flip, or that the model(s) could predict the rating right at least 70% of the time. From a data analytics point of view, the translated goal was to achieve, at least, an accuracy of 70% in the chosen model(s). Although that’s the translation from the business success criteria, the project team will try to do better than that given the limitations of the accuracy as a predictive measure.

To do this job, the situation was assessed in order to understand what resources the team had at its disposal, terminology, contingencies, requirements, and so on.

Regarding resources, each team member had a personal computer, and several were the options to create and get the results, which are the usage of programming languages used in data science, Python and R language, and a software option, RapidMiner. The tool chosen was Python 3.7. IDE’s used were Pycharm, and notebooks were Jupyter and Google Colaboratory. The dataset was downloaded from the Kaggle website. In terms of requirements, the deadline of the project is the 5th of december, 2022, and because this is a supervised learning problem with categorical labels, then the target label should have, at least, 3 classes.

In terms of business terminology:

- Fixed Acidity: most acids involved with wine or fixed or nonvolatile;
- Volatile Acidity: the amount of acetic acid in wine;
- Citric Acid: found in small quantities, citric acid can add 'freshness' and flavor to wines;
- Residual Sugar: the amount of sugar remaining after fermentation stops;
- Chlorides: the amount of salt in the wine;
- Free Sulfur Dioxide: the free form of SO2 exists in equilibrium between molecular SO2 (as a dissolved gas) and bisulfite ion;
- Total sulfur dioxide: amount of free and bound forms of S02;
- Density: the density of water is close to that of water depending on the percent alcohol and sugar content;
- pH: describes how acidic or basic a wine is on a scale from 0 (very acidic) to 14 (very basic);

- Sulphates: a wine additive which can contribute to sulfur dioxide gas (S02) levels;
- Alcohol: the percent alcohol content of the wine;
- Quality: output variable (based on sensory data, score between 0 and 10).

In term of data analytics terminology (techniques will be explained further in the report):

- NB: Naive Bayes;
- KNN: K-Nearest Neighbor;
- LR: Logistic Regression;
- DT: Decision Tree;
- ANN: Artificial Neural Network;
- MLP: Multi-Layer Perceptron neural network;
- SVM: Support Vector Machines;
- LinearSVC: Linear Support Vector Machines;
- RF: Random Forest;
- adaBoost: Adaptive Boosting;
- gBoost: Gradient Boosting (for classification);
- SMOTE: a state of the art technique to balance a dataset that performs oversampling via the generation of artificial examples;
- Random Oversampling: introduces replicas of randomly selected examples from the minority classes of the dataset;
- accuracy: number of predicted labels comparing with the total number of predictions;
- precision: number of predicted labels of a certain class comparing with the total number of predictions for that class;
- recall: number of labels predicted right for a certain class comparing with the total number of the true labels for that class;
- f1-score: predictive measure that takes into account precision and recall to access a classifier;
- AUC: Area Under the Curve (ROC curve), one of the most complete predictive measures because it tells how much a model is capable of distinguishing between classes, in other words, represents the degree or measure of separability for a model;
- Some descriptive statistics: M stands for mean, STD for standard deviation, and ME for median.

4. Data understanding

In this step, the researchers have a first look at the raw data, compute some statistics and encode that data in a visual fashion. Not only that, but it is also necessary to search for missing values, outliers, and other data quality related issues like completeness, noise, irrelevancy or inconsistency in the data. The dataset was retrieved from the Kaggle website [4].

The datasets contained 4898 and 1599 instances for the red and white wines datasets.

In the red wines dataset, the predictive variables (except the wine type), and their respective statistics, are fixed acidity (M=8.320, STD=1.741, ME=7.900), volatile acidity (M=0.528, STD=0.179, ME=0.520), citric acid (M=0.271, STD=0.195, ME=0.260), residual sugar (M=2.539, STD=1.410, ME=2.200), chlorides (M=0.087, STD=0.047, ME=0.079), free sulfur dioxide (M=15.875, STD=10.460, ME=14.000), total sulfur dioxide (M=46.468, STD=32.900, ME=38), density (M=0.997, STD=0.002, ME=0.997), pH (M=3.311, STD=0.154, ME=3.310), sulphates (M=0.658, STD=0.170, ME=0.620) and alcohol (M=10.423, STD=1.066, ME=10.200).

On the other hand, in the white wines dataset, the predictive variables (except the wine type), and their respective statistics, are fixed acidity (M=6.855, STD=0.844, ME=6.800), volatile acidity (M=0.278, STD=0.100, ME=0.260), citric acid (M=0.334, STD=0.121, ME=0.320), residual sugar (M=6.391, STD=5.072, ME=5.200), chlorides (M=0.046, STD=0.022, ME=0.043), free sulfur dioxide (M=35.308, STD=17.007, ME=34.000), total sulfur dioxide (M=138.361, STD=42.498, ME=134), density (M=0.994, STD=0.003, ME=0.994), pH (M=3.188, STD=0.151, ME=3.380), sulphates (M=0.490, STD=0.114, ME=0.470) and alcohol (M=10.514, STD=1.231, ME=10.400).

From these basic statistics there can be visible some differences, for example in the free sulfur acid variable. In general, the characteristics between both wine variants tend to be very different, however data visualizations are necessary in order to see other differences and verify the ones present above.

Besides the wine type that is a nominal variable, all other predictive attributes are quantitative. The target attribute, the quality, is presented as an ordinal variable, ranging theoretically from 0 to 10, although none of the instances achieves those extreme grades (Figure 2).

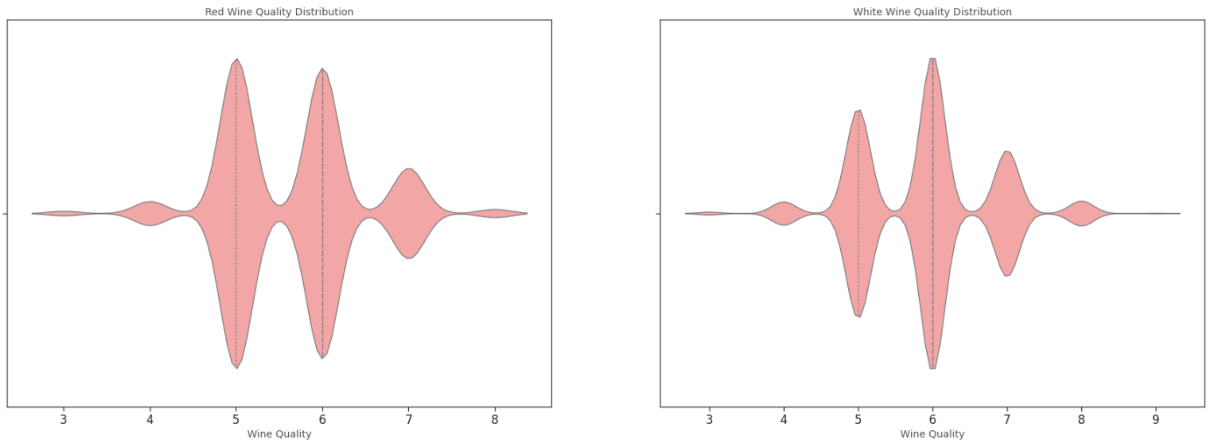

Figure 2 - Distribution of the quality, not discretized, for both wines

In this step, the existence of missing values was also checked because some methods are sensitive to those occurrences in the dataset. In that regard, the dataset had no missing values and, because all variables are quantitative and one binary, the conclusion is that the dataset is complete. Regarding the rest of the data quality dimensions, the dataset seemed to be accurate, consistent, trustworthy and interpretable. This dataset doesn't seem to be timely because one reference made in the Kaggle website, for this dataset, traces back to 2009. In many cases, not having timely data is a big problem, however in this case it isn't that big of a deal as the project team simply wants to explore the various techniques used, and struggles, in a data mining project.

Although it was said that the dataset was consistent, the fact is that, as in any other dataset, there were some outliers (Figure 3). The definition of an outlier is something very subjective, because depending on the evaluator, one instance could be an outlier or not. Also in terms of techniques to detect outliers, there are some subjective aspects. For example, clustering techniques like DBSCAN could be used to detect outliers, however that technique has two hyper-parameters, minimum number of reachable instances for one to be considered a core instance, and the radius for that assessment.

Another issue that was considered was the possibility of existing duplicate data in the dataset. The existence of duplicate data makes the dataset more redundant, as it does not give new information about the wines.

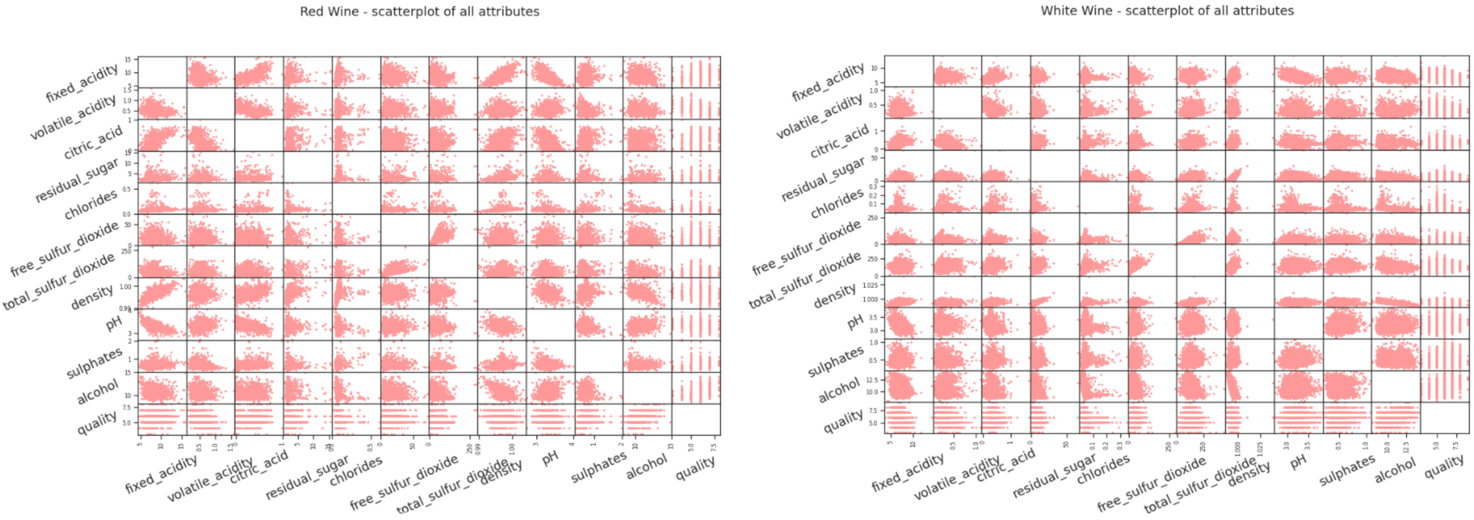

Figure 3 - Quality, not discretized, for both wines

Besides all this, several visualizations were encoded, not only to detect some of the problems mentioned above, but also to have a grasp of what data was being processed/dealt with.

5. Data preparation

In the data preparation step there are several approaches for setting up the final dataset to be used. So, after the data understanding step, where we can look for issues related with the data quality, it is in this step where dealing with those issues becomes tangible. Keep in mind that the order of these steps is very important, so when running the python script the order shown shouldn't be neglected.

Because the original datasets didn't have a column for the wine type, and because that could be an interesting predictive attribute for the models, the first step was to create a new variable, in each dataset, to identify what instances were related to red or white wines.

Now that the instances were labeled with the type of wine, the two datasets were combined into one so that the columns remained the same, and only the rows were concatenated together. This is a data preparation approach known as Data Integration.

After the data integration step, Data Cleaning approaches were applied to deal with duplicated data, as well as outliers. One good example on how important this order is here. Notice that if we first clean the integrated dataset and only then the type of wines are labeled, possible different instances could be removed because they have all other attributes different, but the actual type of wine is different.

For this project, the z-score was used to detect outliers, more specifically, an instance was considered an outlier if the absolute value of the z value, for any quantitative predictive variable, were to be greater than 3. This was a fairly straightforward choice because, in the data preparation step, the data was normalized, making it easy from that point on to assess. Keep in mind that for a z-score greater than 3, the probability of that instance occurring is very unlikely, and so is a standard bound choice when using this method to detect outliers.

The next step in preparing the dataset was transforming the data, more specifically discretizing the target attribute, the quality. To do that, the average of the quality variable was used to first determine what could be considered as a medium quality. Because the average quality for all wines is between 5 and 6, medium quality was considered between 4 and 6. Then, low quality is between 0 and 3, and high quality is higher than 6. We discretize target attributes because it is proven that discretizing variables increases the chances of discovering new knowledge and insights, besides the loss of some detail. To finish, the quality column not discretized was removed from the dataset.

To finish the data preparation step, there is one last approach, Data Reduction. To reduce the number of attributes in the dataset, and to avoid the curse of dimensionality, there are two main ways of proceeding: feature selection and feature aggregation. In feature selection there are filter methods, wrapper methods and embedded methods. In this project, it was decided to pursue a feature aggregation approach as the number of attributes is not that big, and also because feature aggregation was a new concept learned in lectures, and so it was decided that it would be interesting to see its practical impact on results.

The technique chosen to do feature aggregation was PCA, Principal Components Analysis. This technique creates several linear combinations of attributes. Those combinations are called principal components, and they are not correlated between each other. By combining the original attributes, we not only could promote better results in the data mining step, but also that reduces the dimensions of the dataset, and also the complexity of the models. Now, one disadvantage of PCA is that these new predictive attributes are not interpretable, and neither are the results.

Besides applying PCA, the truth is that, depending on the methods requirements, there was a need to apply a filter technique to do feature selection, more specifically finding correlations between predictive attributes and, if they were correlated, remove one of them. PCA already creates non-correlated attributes, however depending on the results feature aggregation could not be the best approach.

To finish, regarding data construction, techniques to balance a dataset were applied only to the training test because to test or validate a model, new only existing instances should be used.

6. Modeling

In the modeling phase of the CRISP-DM methodology the machine learner selects several techniques, in general supervised or unsupervised, in order to describe or predict something based on the dataset prepared in the last step. Keeping that in mind, certain techniques require a specific preparation of the dataset, and because of that we might need to go back to the data preparation step, and then come back to this one [1]. Other things that are done in this step is hyper-parameter tuning and comparing results based on the different hyper-parameters and features selected or aggregated.

The dataset was imbalanced, meaning that there existed a majority class and two minority classes, in this case. Because of that, the project team divided the script for the modeling part in three separated sections, one with the models with imbalanced data, a second for a balanced dataset using the SMOTE technique, and a third section for the random oversampling technique.

Because the final dataset had less than 5 000 instances, it was decided not to use undersampling techniques, since this further reduces the size of the dataset.

The classification techniques used to predict the wine's quality were Naive Bayes (NB), K-Nearest Neighbor (KNN), Logistic Regression (LR), Decision Tree (DT), Artificial Neural Networks (more specifically a Multi-Layer Perceptron Neural Network, MLP), Support Vector Machines (SVM), Linear Support Vector (LinearSVC), Random Forest (RF), Adaptive Boosting (adaBoost) and Gradient Boosting for Classification (gBoost for Classification). A Logistic Regression (LR), to predict the type of wine, was also applied, but the main focus of the project was predicting the wine's quality.

Very briefly explanation of each of those techniques:

- NB assumes that all predictive variables are independent in order to calculate conditional probabilities of a class being associated with a certain instance quicker and more easily; the conditional probabilities come from the Bayes Theorem;
- KNN uses normalized data to calculate the distance between the instance, for which we want to predict the label, and all the other labels, using majority voting in the k closest instances to predict the class;
- LR is a linear regression (univariate or multivariate) where its output is used as input in a logistic distribution to calculate its probability, and depending on a threshold determine the class;
- DT does feature selection by assessing which predictive variables decrease the impurity of the instances that will be separated into two nodes, in a tree-like shape, starting from a root node and ending in leaf nodes, with instances separated by class;
- An ANN is a computerized version of our brain because it learns by making mistakes, more specifically by changing the weights of every connection between perceptrons, minimizing the number of wrong labels predicted; usually, in order to decrease overfit, machine learners add a validation set in order to see if that happens;
- SVM uses margins to separate the instances by their classes and, when the problem is not separated linearly, the algorithm for this technique creates higher dimensions to linearize that separation;
- Linear SVC similar to SVC with parameter kernel='linear', but implemented in terms of liblinear rather than libsvm, so it has more flexibility in the choice of penalties and loss functions and should scale better to large numbers of samples;
- RF decreases the variance of DT by creating several of those models in a parallel manner, and then uses majority voting to determine the class prediction for each instance; in this technique the number of variables available to be inserted in each DT variates, differing from the DT original method;
- AdaBoost and gBoost for classification come from a group of techniques called Boosting techniques.

Different techniques have distinct assumptions. For example, the LR has a representation bias of looking for linear separations between the target label's classes, contrasting with a DT, a SVM or a MLP neural network. However, the ones that should be considered carefully are if the data should be normalized, correlated attributes and near-zero predictors.

Data should be normalized for the following techniques: MLP neural network, SVM, KNN, and LR. Techniques sensitive to highly correlated attributes are the following: MLP neural network and LR. Near-zero predictors should be removed when using the following techniques: MLP neural network, KNN, LR, and NB.

One especially important issue to tackle is the experimental setup. For all classification techniques, besides MLP neural networks, the setup was the following. Using a normalized dataset, and after testing the assumptions of each technique, when applicable, the data was separated in training and test set with a ratio of 80/20. Those two sets were used for all models. To assess if PCA should be applied, and to do hyper-parameter tuning, was used a GridSearch with a 10-fold cross validation in the training set, in order to assess which were the best hyper-parameters and if PCA improved the results or not. Depending on the section of the model, the training set was, or wasn’t, balanced. After this, the final model was developed, and tested using the test set.

Regarding the MLP neural network, the experimental setup was a bit different. The project team decided to implement a more strict experimental setup. In this technique, the dataset was divided into training, validation, and test sets. The training set was used to train the model, the validation set was used to validate and help improve the model by verifying if the model was doing overfit in each epoch. After validating several times, the final model was created, saved, and loaded, to achieve the final results using the test set.

Because the project team created several models for imbalanced and balanced data, the hyper-parameters changed. The hyper-parameters chosen appear in the output of the python script, besides the one of the MLP neural network, that were set up manually. Because there are differences, here will be mentioned the hyper-parameters applying the SMOTE technique. KNN used the manhattan distance and a k equal to 23. For the DT, the criterion was entropy, the max depth of the tree was 6, and the minimum instances in each leaf node was 117. In the MLP neural network, the learning rate was 0.001, number of epochs was 11, the number of layers was four (two hidden layers), the number of nodes was 12, 100, 40, and 3, the activation function was ReLu, and the momentum term was ADAM, Adaptive Moment Optimization. In the SVM was used a C of 0.08, gamma equal to 0.1, and the gaussian kernel (rbf). In RF the hyper-parameters were the gini criterion, max depth of 6, minimum instances in leaf nodes of 117, and 2000 trees. In the gBoost technique the learning rate was 0.01, maximum depth of 10, and 400 estimators. To finish, adaBoost used 500 estimators, a learning rate of 0.2, and the SAMME.R algorithm.

The “Results” section continues the modeling step of the CRISP-DM methodology.

Results

In general, the models were capable of predicting the wine's quality, however the results were not that impressive.

Imbalanced Data

The best results, overall, come from the imbalanced data because the models focus on predicting the majority class, medium quality. On the other hand, the models, supported by the experimental setup explained above, predict more times medium quality wines than the number of medium quality wines in the test set. Because of these two factors, the results for imbalanced data are higher. The results were converted to 3 decimal places and sorted in ascending accuracy order (Table 1).

Table 1 - Score Metrics, performance and time execution for several classification algorithms applied - Imbalanced Data

Model	Accuracy	Precision	Recall	F1 Score	AUC	Execution Time
MLP Neural Network	*0.827*	*0.891*	*0.827*	*0.812*	*0.681*	*00:12:25*
KNN	*0.814*	*0.790*	*0.814*	*0.784*	*0.623*	*00:00:04*
Gradient Boosting	*0.812*	*0.799*	*0.820*	*0.792*	*0.636*	*00:09:27*
Random Forest	*0.805*	*0.778*	*0.803*	*0.751*	*0.568*	*00:02:44*
Decision Tree	*0.804*	*0.774*	*0.804*	*0.769*	*0.600*	*00:00:01*
Linear SVC	*0.803*	*0.774*	*0.802*	*0.751*	*0.570*	*00:00:05*
Logistic Regression	*0.802*	*0.771*	*0.820*	*0.770*	*0.606*	*00:00:03*
Naive Bayes	*0.796*	*0.781*	*0.796*	*0.785*	*0.651*	*00:00:01*
Support Vector Machine	*0.787*	*0.829*	*0.787*	*0.696*	*0.505*	*00:00:16*
AdaBoost	*0.755*	*0.770*	*0.755*	*0.761*	*0.671*	*00:02:36*

Being the table ordered by the accuracy of the test set, we can see that the 3 best models come from the MLP neural network, KNN and gBoost techniques, respectively. Regarding the f1-score, the order changes to MLP neural network, gBoost, and NB. From an AUC perspective, the best 3 models come from the MLP neural network, AdaBoost, and NB techniques, respectively. Looking at the fastest algorithms, the techniques associated are NB, DT, and LR. Next will be described the three best models for accuracy.

Regarding the MLP neural network model computed, it has the highest predictive measures of all, except precision (the best is SVM), predicts 720 out of the 768 test labels for the majority class, yet, for the low rating the results were not that great (0 out of 4 predicted correctly), as well as in the high rating (89 out of 206 predicted correctly). In terms of predictive measures, for the low rating the results were 0 for the precision, recall, and f1-score. For the high rating a precision of 65%, a recall of 43% and a f1-score of 52%. The AUC for the MLP neural network was 68.1%, the best among all techniques.

Applying the KNN technique, for the low rating, 0 out of 4 labels were correctly predicted, in the medium rating 735 out of 768 were correctly predicted, and in the high rating 61 out of 206 were correctly predicted. In terms of predictive measures, the low rating has 0 for the precision, recall and f1-score, whereas for the high rating it has a precision of 65%, recall of 30%, and f1-score of 41%. Looking at the AUC, the results were 49,18%, 83,65%, and 84,36% for the low, medium, and high ratings, respectively.

In the gBoost model, for the low rating it wasn't able to predict any of the 4 test labels, in the high rating predicted 62 out of 206. In terms of predictive measures for the minority classes, it got 0 for precision, recall and f1-score, whereas for the high rating it got a 65% for precision, 30% for recall and 41% for the f1-score. Looking at the AUC for each rating, the results were 51.16%, 83.07% and 83.74% for low, medium and high ratings, respectively.

Any of the techniques used to create models (except for KNN that doesn't create any model) were able to predict any of the 4 labels for the low quality wines which is expected given the fact that the training set used was imbalanced.

For the execution time there is a clear difference between Adaboost, gBoost and even RF and the other remaining methods. These previously identified have times of approximately, 9 minutes and 27 seconds, 2 minutes and 33 seconds and 2 minutes and 44 seconds, respectively, while all the remaining others take only a few seconds to be executed.

SMOTE

Regarding the results for balanced data, and starting with the results for the SMOTE technique. In this case, the models don't give too much attention to the majority class, and thus when assessing the results for the test set, no longer the vast majority of the predictions fall in the medium class. The results were converted to 3 decimal places and sorted in ascending accuracy order (Table 2).

Table 2 - Score Metrics, performance and time execution for several classification algorithms applied - Balanced Data through SMOTE technique

Model	Accuracy	Precision	Recall	F1 Score	AUC	Execution Time
Gradient Boosting	*0.796*	*0.799*	*0.796*	*0.797*	*0.691*	*00:16:59*
MLP Neural Network	*0.766*	*0.812*	*0.766*	*0.781*	*0.743*	*00:14:23*
Random Forest	*0.711*	*0.823*	*0.711*	*0.742*	*0.750*	*00:08:39*
Support Vector Machine	*0.700*	*0.822*	*0.700*	*0.732*	*0.750*	*00:01:35*
AdaBoost	*0.652*	*0.812*	*0.652*	*0.690*	*0.722*	*00:07:13*
KNN	*0.639*	*0.817*	*0.639*	*0.885*	*0.723*	*00:00:07*
Decision Tree	*0.636*	*0.791*	*0.640*	*0.690*	*0.701*	*00:00:03*
Naive Bayes	*0.596*	*0.800*	*0.596*	*0.656*	*0.686*	*00:00:02*
Logistic regression	*0.575*	*0.801*	*0.575*	*0.643*	*0.678*	*00:00:04*
Linear SVC	*0.509*	*0.815*	*0.609*	*0.577*	*0.674*	*00:00:11*

When it comes to accuracy, the 3 best models come from the gBoost, MLP neural network, and RF techniques, respectively. Regarding the f1-score, the best techniques were KNN, gBoost and MLP neural network, respectively. Looking at the AUC, the best techniques are the RF, SVM and MLP neural network, respectively. Looking at the fastest algorithms, the techniques associated are NB, DT, and LR. Next will be described the three best models for accuracy.

Regarding the gBoost model computed, although it has the highest accuracy, that fact occurs because it predicts 674 out of the 768 test labels for the majority class, for the low rating (0 out of 4 predicted correctly) and, for the high rating (104 out of 206 predicted correctly). In terms of predictive measures, for the low rating the results were 0 for the precision, recall, and f1-score. For the high rating a precision of 57%, a recall of 50% and a f1-score of 53%. For the AUC, the results were 42.02% and 84.24% for the low and high ratings, respectively. The medium rating got an AUC of 81.54%.

In the MLP neural network, for the low rating it wasn't able to predict any of the 4 test labels, however it decreased the predictions for the majority class (604 out of 768 predicted correctly) in order to increase the correctly predicted labels for the high rating, with 145 out of 206. In terms of predictive measures for the minority classes, it got 0 for precision, recall and f1-score, whereas for the high rating it got a 48% for precision, 70% for recall and 57% for the f1-score. Looking at the AUC, it got a result of 74.34%.

Looking at the RF model trained, the number of correctly predicted labels for the majority class decreases, giving space to a higher number of correctly predicted test labels for the high rating. For the low rating, 0 out of 4 labels were correctly predicted, in the medium rating 528 out of 768 were correctly predicted, and in the high rating 167 out of 206 were correctly predicted. In terms of predictive measures, the low rating has 0 for the precision, recall and f1-score, whereas for the high rating it has a precision of 45%, recall of 81%, and f1-score of 58%. Looking at the AUC, the results were 40.43%, 79.67%, and 83,74% for the low, medium, and high ratings, respectively. In general, it got an AUC of 75.05%.

Some models were able to predict some of the 4 labels for the low quality wines. With one prediction correct, the techniques were KNN, LR, and SVM. With two predictions correct, the techniques were DT, and LinearSVC.

Random Oversampling

For the results of the random oversampling technique, similar decay occurs, and with the same explanation as for the SMOTE technique. As in the previous ones, the results were converted to 3 decimal places and sorted in ascending accuracy order (Table 3).

Table 3 - Score Metrics, performance and time execution for several classification algorithms applied - Balanced Data through Oversampling technique

Model	Accuracy	Precision	Recall	F1 Score	AUC	Execution Time
Gradient Boosting	0.814	0.796	0.814	0.801	0.673	00:13:48
Random Forest	0.779	0.824	0.779	0.792	0.767	00:07:36
MLP Neural Network	0.756	0.788	0.756	0.767	0.705	00:22:13
KNN	0.730	0.824	0.730	0.753	0.759	00:00:05
Support Vector Machine	0.724	0.627	0.724	0.751	0.645	00:01:19
AdaBoost	0.720	0.791	0.718	0.739	0.712	00:06:42

Decision Tree	0.687	0.776	0.687	0.720	0.678	00:00:02
Logistic regression	0.561	0.801	0.561	0.633	0.674	00:00:04
Naive Bayes	0.528	0.802	0.528	0.594	0.671	00:00:01
Linear SVC	0.510	0.804	0.510	0.578	0.664	00:00:11

Being the table ordered by the accuracy of the test set, we can see that the three best models come from the gBoost, RF techniques, and MLP neural network, respectively. Regarding the f1-score, it is visible that the order maintains the same as for the accuracy.

However, looking at the AUC, that no longer is the case. From an AUC perspective, the best three models are the RF, KNN and adaBoost, respectively. Comparing these three models, in terms of execution time, KNN is way the fastest, followed by adaBoost and RF.

Regarding the gBoost model computed, although it has the highest accuracy and f1-score, that fact occurs because it predicts 707 out of the 768 test labels for the majority class, but falls flat in predicting the low rating (0 out of 4 predicted correctly) and in the high rating (68 out of 206 predicted correctly). In terms of predictive measures, for the low rating the results were 0 for the precision, recall, and f1-score. For the high rating a precision of 60%, a recall of 43% and a f1-score of 50%. For the AUC, the results were 54.54% and 83.69% for the low and high ratings, respectively. The medium rating got an AUC of 83.12%.

In the RF model, the higher AUC comes from the fact that it was able to better predict the high rating. For the low rating it wasn’t able to predict any of the 4 test labels, however it decreased the predictions for the majority class in order to increase the correctly predicted labels for the high rating, with 156 out of 206. In terms of predictive measures for the minority classes, it got 0 for precision, recall and f1-score, whereas for the high rating it got a 49% for precision, 76% for recall and 60% for the f1-score. Looking at the AUC for each rating, the results were 59.75%, 83.28% and 85.12% for low, medium and high ratings, respectively.

Looking at the MLP neural network, the number of correctly predicted labels for the majority class decreases, giving space to a higher number of correctly predicted test labels for the high rating. For the low rating, 0 out of 4 labels were correctly predicted, in the medium rating 610 out of 768 were correctly predicted, and in the high rating 129 out of 206 were correctly predicted. In terms of predictive measures, the low rating has 0 for the precision, recall and f1-score, whereas for the high rating it has a precision of 45%, recall of 63%, and f1-score of 52%. Looking at the AUC, the result was 70.50%.

Some models were able to predict half of the 4 labels for the low quality wines, those being LR and LinearSVC.

Wine variants

Besides training models to predict wine ratings, the team also trained a LR model to predict if the wine was red or white wine. In other words, a binary classification problem. Because this was not the main goal of the project, the resources allocated to this task were minimal.

In terms of results, the LR got an accuracy, precision, recall, and f1-score of 99.4%, as it was able to predict 218 (out of 223) of the red class of wine type, and 754 (out of 755) of the white wine variant; the AUC result was 98.8%. This was done with an imbalanced dataset.

7. Evaluation

In the evaluation phase the main focus is to abstract ourselves from the data analytics interpretation perspective, and focus on the value the work done in the modeling step adds to the client, to the business. What this means, then, is to understand how such a solution or model(s) will affect the business, and also if the client's requirements are fulfilled with the work developed [1]. The conclusions and discussion will be done in this 5th step of the CRISP-DM methodology.

The usage of an imbalanced dataset is not recommended because the models become more prone to predict any label as part of the majority class. In fact, the results obtained reflect that. The AUC of the models trained with an imbalanced dataset are the lowest, showing a lack of capacity to predict labels from the less populated classes. One consequence of that is the inflation of the accuracy, a predictive measure very used and known, although its limitations and misleading conclusions are justified by the work done in this project.

Regarding the assessment of the models developed, twenty of them were able to surpass the data analytics requirement of delivering a model with an accuracy of, at least, 70%. However, because the project team knew about the limitations of only looking at that predictive measure, the team manipulated the models, specially the MLP neural network, in a way to have good results in terms of accuracy, but also in terms of AUC and f1-score (and consequently precision and recall). For example, in the MLP neural network model for the balanced dataset using random oversampling, the accuracy was decreased on purpose in order to increase the AUC and the other predictive measures.

So, because of the problem with imbalanced datasets, the models approved, based on the business requirements, are:

- gBoost, using SMOTE or Random Oversampling;
- MLP neural network, using SMOTE or Random Oversampling;
- RF, using SMOTE or Random Oversampling;
- SVM, using Random Oversampling;
- KNN, using Random Oversampling;
- adaBoost, using Random Oversampling;

From these models, the top four models in terms of AUC, and recommended by the project team to the client, are:

1. RF, using Random Oversampling;
2. KNN, using Random Oversampling;
3. RF, using SMOTE;
4. MLP neural network, using SMOTE;

In terms of the work done in this project, and because it is the first work in a data mining project, a process review is especially important.

Looking back and ahead, there were some improvements that could be interesting to do. Starting with more detailed data and a greater amount of data could have allowed the application of other methodologies/techniques in the development of the project. Simultaneously, a study and evaluation of the application of more models for the type of wine could have been carried out, even if this was not the main objective. Perhaps it could be possible to train the regression model to predict the quality as a number or even try a quality classification in the all range of quality.

For the positive points, it is important to highlight the different models applied, and essentially the perception, study and analysis of what is behind each one, which ends up justifying the different values presented, even if they vary when comparing unbalanced and balanced data, also as expected.

To reach these conclusions the methodology chosen was essential and helped in terms of process organization and management. Having a methodology allowed to set and split tasks and at the same time follow a path with a final goal - essentially predict the quality of the wine.

In the end, the research was a good way to explore the immense capabilities and tunes of Python language and Machine Learning Techniques, allowing to put into practice and improve the knowledge acquired throughout the IACEC curricular unit.

8. References & Bibliography

[1] João Mendes Moreira, André C. P. L. F. de Carvalho, Tomáš Horváth, “A General Introduction to Data Analytics”, John Wiley & Sons, 2019.

[2] Destin Gong, “Top 6 Machine Learning Algorithms for Classification”, 2022.

Available in: https://towardsdatascience.com/top-machine-learning-algorithms-for-classification-2197870ff501

[3] Rohan Joseph, “Grid Search for model tuning“, 2018.

Available in: https://towardsdatascience.com/grid-search-for-model-tuning-3319b259367e

[4] MARCELO MARQUES, “Wines Type and Quality Classification Exercises”, 2018. Available in: https://www.kaggle.com/code/mgmarques/wines-type-and-quality-classification-exercises