Application of Deep Learning tools in data analysis for the "Data Science and Engineering Club UAB"

Ferran Rodríguez Mir

Resumen – Ante un mundo cada vez más complejo, con mayores interacciones entre usuarios y sistemas informáticos de datos, se está generando una gran necesidad por parte de varios sectores empresariales de profesionales especializados en el Análisis y la Ciencia de Datos. Con el afán de responder a esta creciente necesidad nace el Data Science and Engineering Club en la Universidad Autónoma de Barcelona, una plataforma web que permite exponer a un público específico cómo los estudiantes son capaces de resolver distintos problemas que abarcan grandes cantidades de datos mediante el uso de herramientas de Machine Learning y Deep Learning. Así mismo, el objetivo principal de este proyecto es el de estudiar y proponer una solución a varios problemas utilizando distintas técnicas de Aprendizaje Computacional, mostrando el conocimiento adquirido, así como las técnicas utilizadas y resultados obtenidos, en la plataforma web del Club.

Palabras clave – Aprendizaje Computacional, Ciencia de Datos, Análisis de Datos, Machine Learning, Deep Learning

Abstract – In an increasingly complex world, with greater interactions between users and computer data systems, a great need is being generated by several business sectors for professionals specialized in Data Science and Data Analysis. In order to respond to this growing need, the Data Science and Engineering Club was born at the Autonomous University of Barcelona, this being a web platform that allows students to expose specific audience how to solve different problems that cover large amounts of data through the use of Machine Learning and Deep Learning tools. That is to say the main objective of this project is to study and propose a solution to several problems using different Computational Learning techniques, showing the knowledge acquired, as well as the techniques used and results obtained, on the Club's web platform.

Keywords – Data Analysis, Data Science, Machine Learning, Deep Learning

1 Introduction

With the arrival of new technologies and information systems that move large amounts of data, the concepts of Machine and Deep Learning appeared, disciplines within the field of Artificial Intelligence that allow the automation of analytical models that are able to predict behaviors through the search for common patterns between data features [1]. The quick rise of these new technologies caused many companies the urge of hiring specialized personnel in the field.

Despite this high demand, since these technologies are in their early years, people specialized in this particular field are scarce. For such reasons, a blog-type web platform named after Data Club Engineering [2] was created at the Autonomous University of Barcelona, allowing Computer Science students interested in the field of AI and Data Science to participate in Machine and Deep Learning competitions as well as propose solutions to real problems using different tools and techniques learned throughout the years.

In addition, this platform not only works as a learning tool in which students can show their progress in the field and learn from others, but also, since it is public, it may work as an intermediary between students and companies in the field of AI that may be interested in hiring specialized

• E-mail de contacte: ferran.rodriguezmi@e-campus.uab.cat
• Menció realitzada: Computació
• Treball tutoritzat per: Jordi Gonzàlez (Ciències de la computació)
• Curs 2019/20

personnel in Data Science after seeing what students are capable of.

That said, the main objective of this work is to propose different solutions to various problems using Machine and Deep Learning techniques, creating content for the Blog [2]. Moreover, improvements will be made to the Blog in order to facilitate user interactions.

2 Dataset Selection

In order to carry out this work, a dataset selection process has been strictly followed based on different criteria. Those datasets that have been selected are found to be interesting enough to be studied and obtain useful applicable knowledge in different research areas and, in turn, each dataset gives the opportunity to be proposed a solution by using a different technique. The chosen datasets are the following ones:

1. **Breast Cancer Wisconsin** [3]. The purpose of this problem is to predict whether a breast tumor is benign or malignant through data analysis since breast cancer is the most common type of cancer in women around the world and its early detection may save the life of patients by giving them medication as soon as possible.

2. **Predicting a Pulsar Star** [4]. Pulsars are a very rare type of neutron star that produce radio emission detectable on Earth and they are of considerable scientific interest as probes of space-time and states of matter. However in practice almost all detections are caused by radio frequency interference and noise, making legitimate signals hard to find, which is why the aim of this problem is to predict whether a detected signal comes from a pulsar star or from any other source.

3. **Red Wine Quality** [5]. The main goal of this problem is to find out which features of a red wine variant from Portugal are the most important ones when creating a good wine. We will also try to make a prediction of wines' qualities and check if they match with the real qualifications.

4. **Mall Customer Segmentation** [6]. In this dataset the main approach we are taking is to learn the purpose of customer segmentation concepts, also known as market basket analysis, trying to understand customers and separate them in different groups according to their preferences. This is very useful since this segmentation can be given to any marketing team so they can plan a customer strategy accordingly.

As specified above, each dataset allowed to be exposed to a different Machine Learning technique and this is what has been done. For the first two datasets, different classification techniques have been used, which will be specified more accurately later. For the third dataset, regression techniques have been used since it required a numerical continuous prediction. Finally, for the fourth and last dataset a solution based on unsupervised learning has been applied.

3 Blog Posts

In order to shape each of the studies carried out during the realization of this work, a post has been created for each problem to be solved, which has been uploaded to the Blog [2]. These posts have been created using Python 3.6 [7] onto the Jupyter Notebook Tool [8] and, subsequently exported as HTML since it is the format used in the Blog.

These posts are intended to contribute content to the Blog and show step by step, thanks to small code snippets and explanations, how to provide solutions to these types of problems. In addition, to facilitate the understanding of each of them, they follow a specific format:

First of all the dataset is loaded and a pre-processing of data process is executed to avoid having null or unwanted values. A visual study of the data is then carried out via graphic sampling of different features that are considered of interest. Secondly, Machine Learning algorithms that are believed to give a better solution to the problem are applied. Finally, the results obtained are collected and analyzed in order to extract knowledge and conclusions from the data.

3.1 Breast Cancer Detection

In order to provide a solution to this problem, the Breast Cancer Wisconsin (Diagnostic) dataset [3] has been used. In this post [9] we are going to predict whether a tumor is benign or malignant building Machine Learning classifiers as well as trying to understand which features are important when detecting breast cancer through data analysis.

Once these models are built, they are going to be tested and used to make predictions that will be compared with the real values so we can get to know how accurate they are.

3.1.1 Data Analysis

When it comes to pre-processing data in any database, it seems interesting we first study its size to see how many columns and rows it contains, in this case the database consisted of 569 rows and 33 columns. Once we had an idea of the size the database that we are going to deal with had, it was checked what type of value each feature had and we observed that there was a column whose values were all null, so it was decided to discard the full column. All other values were numerical, except for the column that referred to the tumor diagnosis. It contained B in case the tumor was benign and M in case it was malignant. Since Machine Learning algorithms work better with numerical data, it was decided to change these values to 1 if the tumor was malignant and 0 if the tumor was benign, thus indicating that the patient did not suffer from cancer.

After the dataset was free from undesired values, we proceeded to study it graphically. Taking into account that the target variable in this problem was the diagnosis of the patient we analyzed that column to see what kind of data it contained and have a broader view of the problem. The results were that a large part of tumors, 357, were benign, which made us believe an early detection of a tumor can greatly improve chances of survival. On the other hand, 212 tumors were malignant.

With this information, the Pearson correlation coefficient was computed in order to know how strong the correlation between features and the diagnosis column was, so we

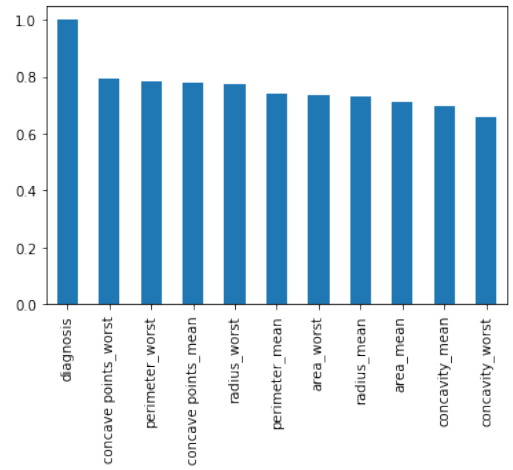


Fig. 1: Correlation with Diagnosis

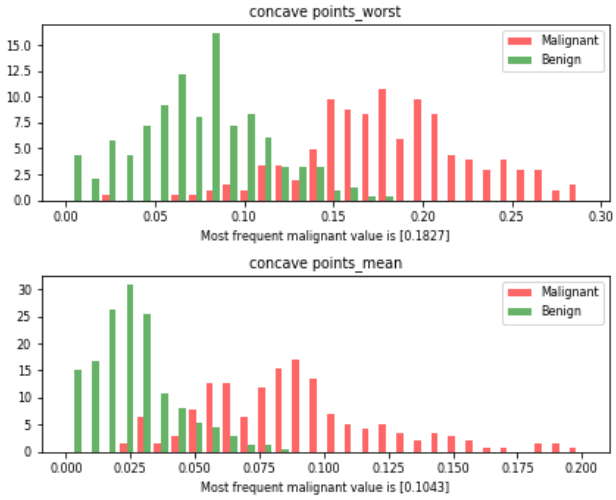


Fig. 2: Histogram of Concave Points Worst and Mean

could get to know how important each of them was when it came to deciding whether a tumor was benign or malignant. After doing so, those characteristics whose correlation exceeded 0.6 were selected, since they clearly were the ones that could provide most information. From a total of ten different computed features for each nucleus cell, these being the radius, texture, perimeter, area, smoothness, compactness, concavity, concave points, symmetry and fractal dimension, only the worst measure of the perimeter, concave points, area, radius and concavity of the tumor as well as the mean of these values have been selected as seen in the Figure 1.

Having selected the most important features of the problem, it was interesting to see each of their histograms, separating values in benign and malignant so it could be seen from which values the tumors began to seem malignant. Since there are a total of ten different characteristics, only two are shown in Figure 2 although they can all be seen in the Blog.

After visualizing the different histograms, it is clear that the difference in values between features is quite large, so if all the data together is to be shown in a graph it is necessary to normalize it first. After doing this normalization, the distribution of values for each feature can be seen in the

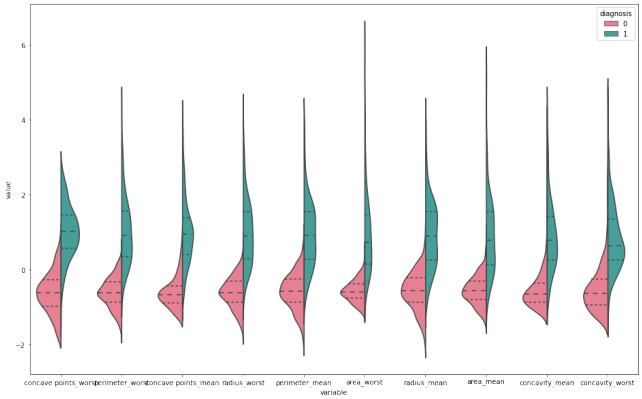


Fig. 3: Normalized Data Distribution

Figure 3.

Thanks to the Figures 2 and 3, which can be seen in the Blog post with greater detail, it can be said that the selection of features has been successfully done, since we can see how the probability of a tumor being malignant increases when these feature values also increase. Additionally, in the Blog it is shown from which values a tumor is almost surely malignant.

3.1.2 Machine Learning

In this section of the post, different models of classifiers will be built and get their performance checked to see which one works best for this type of problem.

In the case of this particular problem, something interesting has been put to the test. First, Machine Learning models have been built to act on the features we had selected before based on the correlations. Then, a Machine Learning algorithm that used automatic feature selection has been used. Finally their performance has been compared and some conclusions have been drawn.

That said, before applying any computing technique, as explained in the post, it is important to divide our dataset into two main groups: training data and test data. The training data will serve us as a way to model the behavior of our model. Once this modeling process is finished, the performance of our model will be checked using the test data to make predictions, since this data has not been "seen" by the model yet.

Having already divided our data, we proceed to apply Machine Learning algorithms on the features that we had previously selected. The first one is a parameterized SVM (Support Vector Machine) that uses Cross-Validation, this allows to automatically select the parameters that have provided the best performance and avoid Overfitting. These parameters have been a regularization parameter "C" equal to 10, and the used kernel has been a linear one. Next, the Gaussian Naive Bayes algorithm has been applied, which is quite simple but very powerful in solving classification problems. Finally, a classification method based on Decision Trees has been used.

Then, a Random Forest algorithm has been used with automatic feature selection which, surprisingly, in most cases selected 22 as the optimal number of features to solve the problem with. That said, in the following section results are going to be explained in greater detail.

TABLE 1: SVM AND RF METRICS

Algorithm	Accuracy	Recall	F1Score
SVM	0.970	0.968	0.960
RF	0.959	0.920	0.943

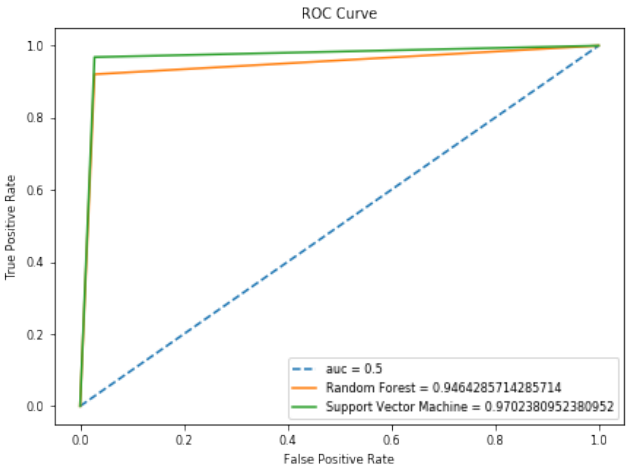


Fig. 4: ROC Curves for SVM and RF

3.1.3 Results

Once models have been trained, we proceed to evaluate them through different metrics, these being Accuracy, Recall and F1Score in order to assess which of these algorithms works best for this problem.

In this case, as expected, from the three techniques we applied onto the manually selected features, the SVM algorithm has proven to be the best, which makes sense since it was allowed to chose the best SVM parameters thanks to parametrization. In addition, surprisingly, these metrics resulted to be slightly better than the ones obtained in Random Forest with automatic feature selection. These results are shown in Table 1.

Additionally, apart from these metrics, predictions have been made using both the SVM and Random Forest models. In order to visually see these predictions, two confusion matrices have been created, one per model. In these confusion matrices it can be clearly seen that out of a total of 171 test samples, the SVM incorrectly classifies a total of 5 patients while the Random Forest a total of 7. It seems that these samples were misclassified because the value of each feature was in the lower or upper limit of the class, acting as an outlier, resulting in the algorithm not being able to decide with certainty to which class they belonged.

This reaffirms what has been stated before, the SVM algorithm works slightly better. The ROC curve for each of the two algorithms is shown in the Figure 4, where it can be seen again how the SVM algorithm has an AUC (Area Under Curve) superior to the Random Forest. In general terms, the ROC curve indicates that there is a 97% chance that the SVM algorithm and a 94% that the RF algorithm will be able to correctly classify a sample.

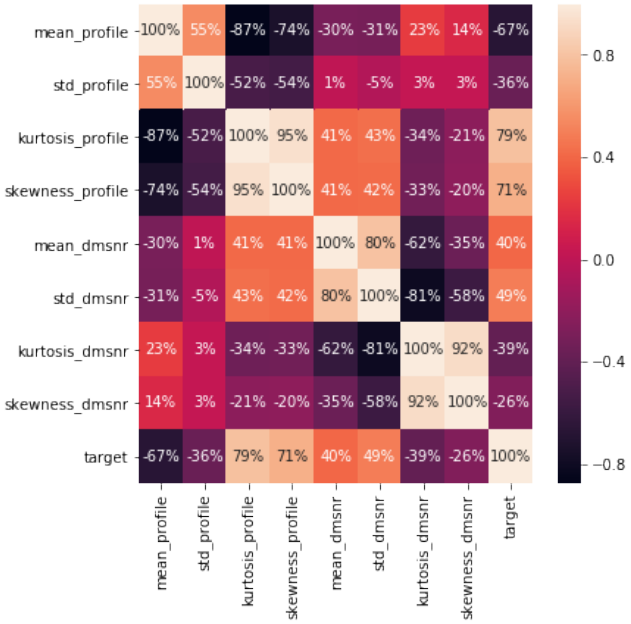


Fig. 5: Correlation Map in HTRU2 Dataset

3.2 Pulsar Stars Detection

In this post [10] the HTRU2 dataset [4], which describes samples of Pulsar candidates collected during the High Time Resolution Universe Survey, is going to be used in order to provide a solution for this problem. The main aim of this post is to build a classifier able to detect whether a radio emission comes from a pulsar star or by radio frequency or noise interference.

To do this, in the post it will be shown how to build a fully connected ANN (Artificial Neural Network) step by step using the Keras library [11] provided by Tensorflow, introducing the reader to the world of Deep Learning.

3.2.1 Data Analysis

In this dataset each candidate is described by 8 continuous variables, and a single class variable. The first four are simple statistics obtained from the integrated pulse profile and the remaining four variables are the same ones similarly obtained from the DM-SNR curve [12]. These features are the mean, standard deviation, skewness and kurtosis. The ninth variable, which is the class, indicates 1 if it belongs to a star or 0 if it comes from another source.

The dataset used here contains a total of 17,898 samples, 16,259 of which are RFI/noise, and 1,639 are real pulsar stars.

Once the size of the database is determined, the correlation map between variables is shown to have a better understanding of which features may be more useful, although no manual feature selection will be done for this problem since the neural network will do the job. This correlation map is shown in the Figure 5 and it can be observed how four of the eight features correlate positively with the target variable whilst the other four correlate negatively with it.

The next thing that is done in the post is to create a violin plot for each of the variables using the target variable as a distinction class in order to see if it can be visually detected from which specific value it can be already decided

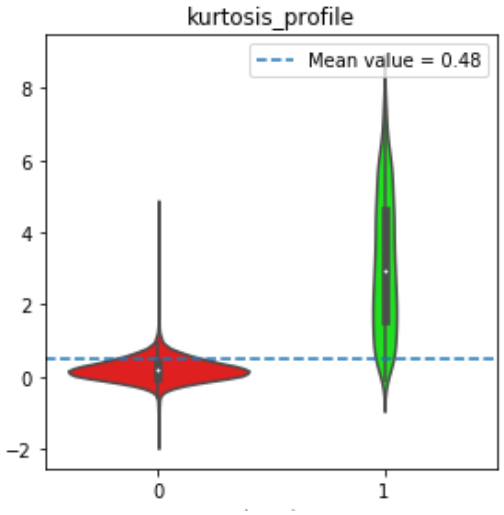


Fig. 6: Violinplot of Kurtosis Profile

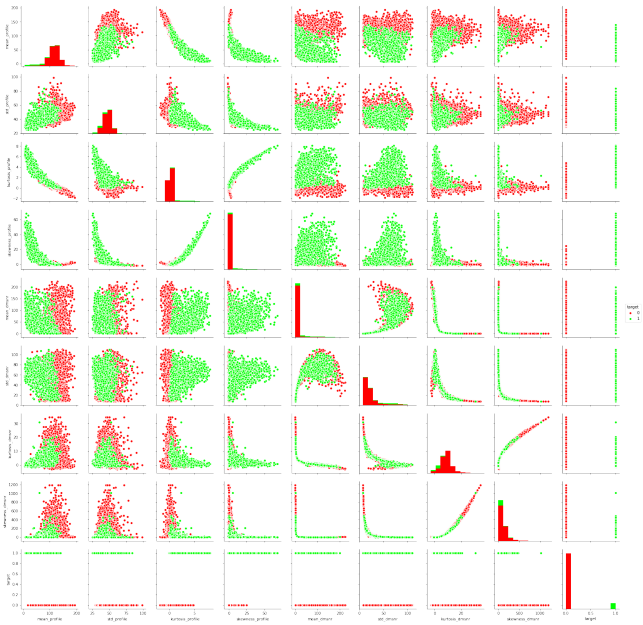


Fig. 7: Pairplot of HTRU2 Dataset

the source of the signal. An example of these violin plots, specifically the one from kurtosis profile, is shown in Figure 6 where it can be observed how the majority of samples whose kurtosis profile value is above the mean value belong to the group of pulsar stars while, with some outliers that break the rules, lower values than the mean come from other signals. In addition, the distribution of values from the "non pulsar" group is pretty similar, meaning the range of values for those signals is quite narrow. However, the opposite happens to the pulsar group, values tend to spread in a range between 0.48-8.

With this information, the last step is to create a pair plot between all variables, again using the target variable as a distinction. In this graphic, which can be seen in the Figure 7, it is shown how well the two classes can be separated in the dataset, thus giving a small clue that it will be easy to build a good classifier.

3.2.2 Deep Learning

This section of the post begins by dividing data into training data and test data, also applying some scaling to the data to avoid overwhelming differences between the values. Next, it is explained how to build an artificial neural network from scratch, layer by layer, following a sequential model.

That said, for this problem a fully connected neural network with a total of 3 different Dense layers is going to be used, two of them will function as hidden layers and one of them as an output layer. In addition, between these Dense layers there will be Dropout hidden layers with a rate of 0.3 in order to prevent Overfitting.

The input dimension is 8 since it matches the number of features and output dimension is 1 since only one label is being received. The number of neurons per layer has been chosen randomly without being excessively high after having tried with bigger and lower numbers and observing the results barely variate. It is also important to mention that, after having tested other activation functions for hidden layers, we are using ReLU as activation function since it provided slightly better results, and Sigmoid for the output layer since this is a binary classification problem. For the output layer a Softmax activation function could have been used as well but, since this is not a multi class classification problem but a binary, there is no real use of using it since it decreased accuracy by 2-3%.

Having already the model defined there comes the time to compile it. As loss function we are using binary cross-entropy since, as it has already been stated before, this is a binary classification problem. As far as optimizer is concerned, Adam optimizer is being used since it adapts the learning rate as the training progresses as opposed to the classical Stochastic Gradient Descent. Finally as far as metrics are concerned, Accuracy is the chosen one.

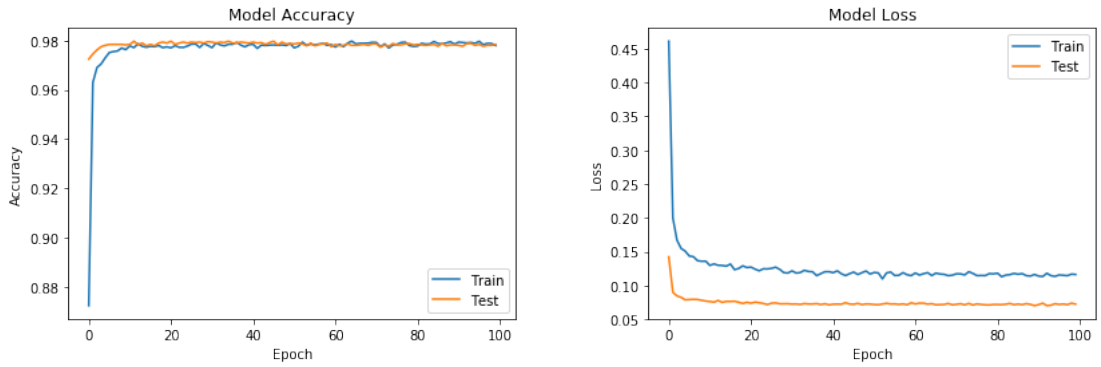
After building the model to be used, before feeding the neural network with our data, it has been decided to adjust the weight of each of the different output classes, establishing a weight x2 for the positive pulsar class. This is because it is more of our interest that the neural network learns better what makes a signal come from a pulsar star because, even if there are some false positives, it is important to detect the maximum amount of signals that come from them. This approach is usually taken when there are way more samples from a class than the other.

With all the parameters set, we proceed to fit the model with a number of epochs equal to 100 and a batch size of 64. During the fitting process, the test data is being used as validation data as well.

3.2.3 Results

The Figure 8 provides some interesting information about the performance of the model that has been built:

On the one hand, it is observed how the accuracy of the model increases during approximately the first ten epochs, while after this quick increase, it barely varies leaving a constant accuracy of approximately 98.3% for both the training and validation data. On the other hand, taking a look at the graphic that shows the loss of the model, keeping in mind that binary cross entropy has been used as loss function, it can be observed that its behavior is quite similar to accuracy, it minimizes during the first twenty epochs and



(a) Accuracy (b) Loss

Fig. 8: Model Accuracy and Loss for Training and Validation Data

then it remains constant at a value of approximately 0.11 for the training data and 0.08 for the validation data.

After plotting the previous graphics, the next step followed in the post is making a prediction using the test data, which has a total of 5,370 samples. From this prediction a total of 119 samples have been misclassified, 62 that were originally stars have been classified as noise and 57 that were not stars have been classified as stars.

It is likely that some pulsars from the test sample were missed due to one of these two reasons: poor training of the neural network, or having an unbalanced training set. In order to slightly improve these results, future implementations should avoid unbalanced training sets or bias the training data toward the objects of interest. However having a large number of false positives is beneficial, as stated before, when looking for rare values in datasets.

In Kaggle [13] it is also shown how some Machine Learning algorithms can obtain an accuracy similar to that obtained here, at least with this amount of data.

3.3 Predicting Red Wine Quality

Red Wine Quality [5] is a dataset from 2009 that contains information of a Portuguese red wine variant called "Vinho Verde" that originated in the historic Minho province in the far north of the country.

The main purpose of this post [14] is, through data analysis, to investigate what makes a good wine. Furthermore, Machine Learning regression techniques are going to be used in order to make a prediction of wines' qualities.

3.3.1 Data Analysis

The first thing that is done in the post is to investigate the size of the database, which has a total of 1599 samples and 12 features, one of which refers to the quality of the wine. Then we proceed to search for null or unwanted values in the database in order to eliminate them so that it does not affect the learning process.

Once it has been reviewed that the data is correct and there are no values that can impede the correct functioning of the models, the data is visually analyzed in order to extract knowledge and information.

Then, in the post it is explained that the target variable of the problem is quality, therefore it is the first one studied.

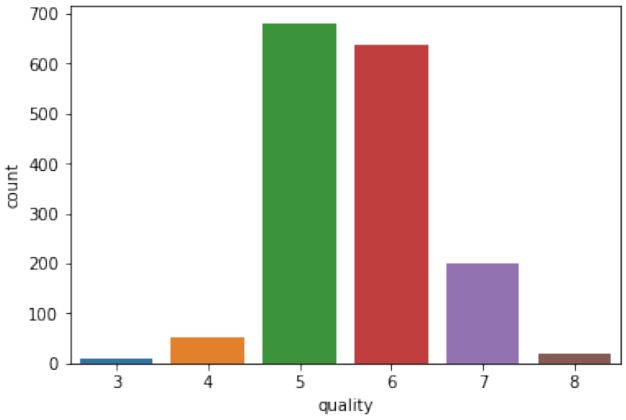


Fig. 9: Red Wine Quality Values

In the Figure 9 it can be seen how there are a total of 6 different qualifications and the vast majority of wines are mediocre, obtaining a rating of 5 or 6. It seems that in this database there are not many bad wines or many excellent wines either. Therefore, it is concluded that this is, again, an unbalanced dataset.

The next step is to study the correlation of different features with the quality of wine, in order to do this, the correlation map between all the variables is shown. After plotting the correlation of features with our target variable, it is observed that the dataset does not have a strong correlation between features, which implies that the results when applying Machine Learning will not be extremely good if we do not engineer any solution.

Then, we proceed to select those features whose correlation with the quality variable is greater than 0.2, thus leaving us, as a result, with four variables: alcohol percent, sulfates, citric acid and volatile acidity. The first three features correlate positively while the last one does so negatively.

Once the most important features of the problem have been selected, we proceed to study each of them using box plots, in the Figure 10 it is shown an example of these graphics for alcohol percent. As it was estimated based on the correlation, the greater amount of alcohol percent, the higher quality the wine gets.

In addition, in order to provide a little bit more of data analysis and visualization, we proceed to split wines into three different categories: low, medium and high rated

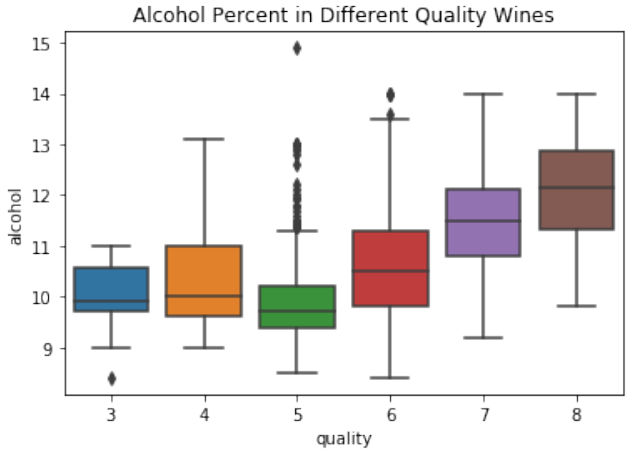


Fig. 10: Alcohol Percent Boxplot for Different Quality Values

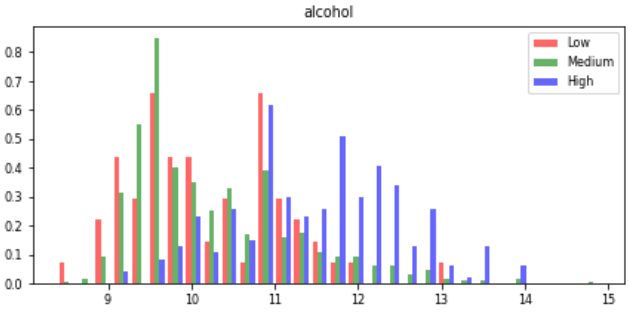


Fig. 11: Alcohol Percent Histogram for Different Qualities

wines. Low rated wines are those with a quality of 3 and 4, medium with 5 and 6 and, finally high rated wines are those that have a quality equal to 7 or 8.

Having done this split, histograms for each feature have been plotted making a class distinction using the target variable where it can be observed, again, how wines with higher values of features that correlated positively belong to the highest rated class. In the Figure 11 it is shown the histogram built for the alcohol percent feature.

3.3.2 Machine Learning

As in the previous problems, this section of the post begins by splitting the dataset into training data and test data. The four previously chosen features are also selected and separated from the others.

Once data is split, we proceed to build the regression models that will be used. In this particular problem Linear Regression, Regressive Decision Trees and a Random Forest with a number of estimators equal to 10 have been used.

In the following section there are only going to be explained the results obtained by the Linear and Random Forest Regression algorithms, since those are the two that have given the best results, although the Decision Tree Regressor can also be seen in the post.

3.3.3 Results

After building these models, predictions have been made on the test data to evaluate their performance. With these predictions it has been observed that results were not very good since there were quite some prediction errors. Moreover, the RMSE measure has been calculated. This metric is the standard deviation of the residuals (prediction errors), which are a measure of how far from the regression line data points are. In other words, it tells how concentrated data is around the line of best fit. The RMSE values for both the Linear Regression and Random Forest were 0.70 approximately, meaning there was room for improvement.

In order to improve these results, we proceeded to round the values predicted and apply a concept called 1-off accuracy, which states that if the distance between our predicted value and the ground truth is 1, in absolute value, the prediction is considered correct.

After applying this function, the RMSE values have been calculated again obtaining an RMSE of 0.27 for the Linear Regressor and 0.35 for the Random Forest. In addition, confusion matrices for both models have been plotted as shown in Figure 12.

As seen in the confusion matrices, the results obtained after applying 1-off accuracy are quite good, since we obtained an accuracy of approximately 0.98 for the Linear Regressor and 0.97 for the Random Forest Regressor. However, there are still some values our regressors are not able to predict correctly, these being wines whose true quality lays in the range of low or high. This can be due to having an unbalanced dataset since the models can predict pretty well wines with a medium quality value of 5 o 6 while good and bad wines seem to be harder to classify as it can be seen on the confusion matrices.

This being said, since there are only six different quality values in this dataset, it would have been clever treating this problem as a classification problem, grouping samples in two groups: wines with a quality less or equal than 5 would belong to the low rated wines, whilst wines with quality equal or greater than 6 would belong to high rated wines. Had we treated the problem like that [15], results might have improved since it would have been a binary classification problem and the 1-off accuracy concept would not have been needed.

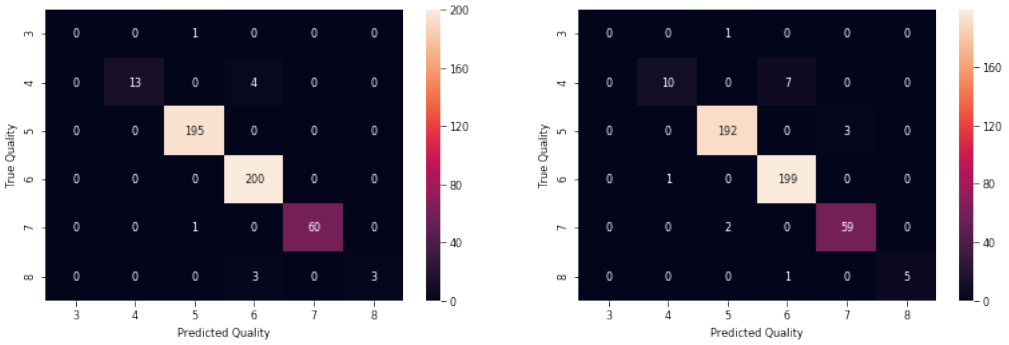
3.4 Mall Customer Segmentation

Let's imagine you are owning a supermarket mall and through membership cards, you have some basic data about your customers like customer ID, age, gender, annual income and spending score, which is something that is assigned to the customer based on defined parameters such as customer behavior and purchasing data.

In this post [16] we are going to be using unsupervised Machine Learning techniques in order to group customers in different groups or clusters taking into account specific features from the dataset [6], something that is also known as market basket analysis and helps the marketing team when planning customer strategies.

3.4.1 Data Analysis

Although this is a very different type of problem from the previous ones, the data analysis is done in the same way. First, the size of the dataset is checked, which contains a total of 200 samples and the 5 features mentioned above.



(a) Linear Regression (b) Random Forest Regression

Fig. 12: Confusion Matrices of Linear and Random Forest Regressors

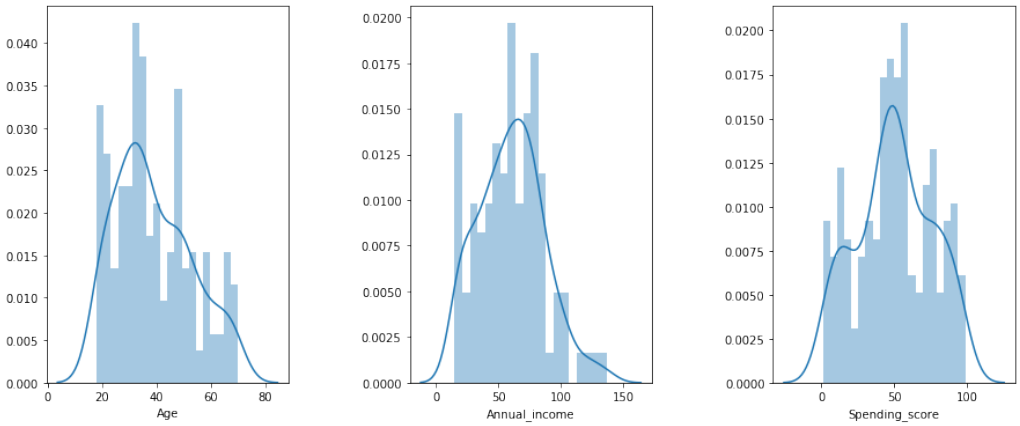
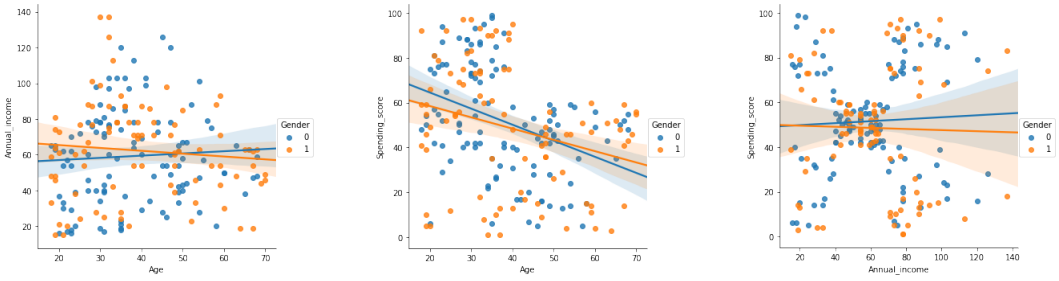


Fig. 13: Age, Annual Income and Spending Score Values Distribution



(a) Annual Income - Age (b) Spending Score - Age (c) Spending Score - Annual Income

Fig. 14: Relation Between the Three Variables Separated by Gender

Next, we look for null values we might want to discard and the values of Male or Female are replaced by 1 or 0 in the gender column, to facilitate the handling of the data.

From the five features the dataset has, we proceed to discard the client ID since it does not provide any information and it is decided that the gender variable will work as a separator when graphing the other three remaining variables: age, annual income and spending score. In the Figure 13 we can see the distribution of values for each of these three features. In these histograms it can be observed that the shape of these values resembles a Gaussian distribution, where the vast majority of the values lay in the middle with some exceptions in the extremes.

The next thing that is done in the post is to count how many men and women there are in the dataset, resulting in a total of 112 women and 88 men. Next, we proceed to show a pair plot between the three variables previously selected, but now using the gender variable to differentiate between men and women. This pair plot can be seen in the post. However, in the Figure 14 it can be seen the relation between these three features and in the post there is a brief explanation for each of them.

3.4.2 Machine Learning

This section of the post explains that, in order to solve this problem, the KMeans algorithm will be used, an unsupervised Machine Learning algorithm that aims to segment a set of N observations into K groups in which each observation belongs to the cluster with the nearest mean. This section also explains how to find the optimal K or number

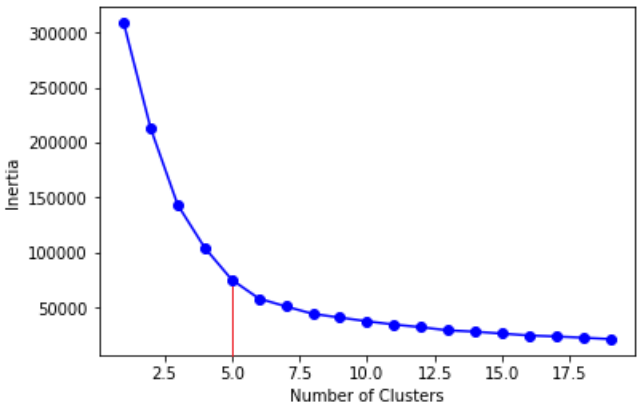


Fig. 15: Elbow Method

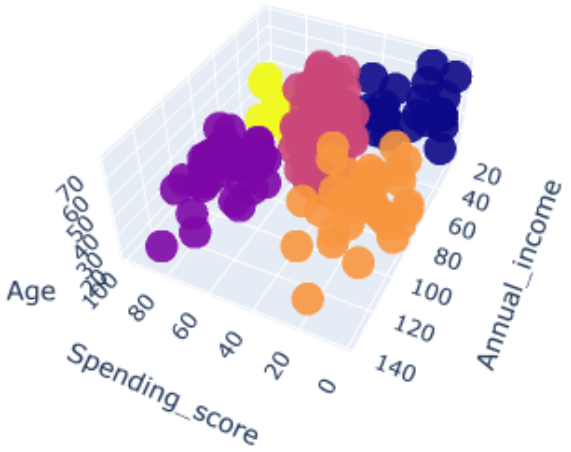


Fig. 16: Clusters Obtained from KMeans

of clusters to divide our data in.

In order to find the optimal number of clusters, the value of a cost function is plotted for each of the different values of K. Once this graph is available, which is shown in Figure 15, the elbow method [17] is used as a heuristic, choosing a number of clusters so that adding another cluster does not give much better modeling of the data. Looking at this particular example, if we imagine the line in the graphic is an arm, the elbow can be found, approximately, where the number of clusters is equal to 5. Therefore we are selecting 5 as the number of clusters to divide our data in.

In this problem, we are using the inertia as cost function in order to identify the sum of squared distances of the samples to the nearest cluster center.

3.4.3 Results

Once the clustering process has finished, the result obtained is an array of labels, one label per sample. These labels have values between 0 and 4, a total of 5 different values corresponding to the number of selected clusters.

In order to visualize these results, it has been chosen to create a 3D plot using this array of labels as a class separator, similar to what has been done with gender before, thus allowing us to paint in the same color those samples that belonged to the same cluster. The result of this plot can be seen in the Figure 16 where the 5 groups are clearly distinguished:

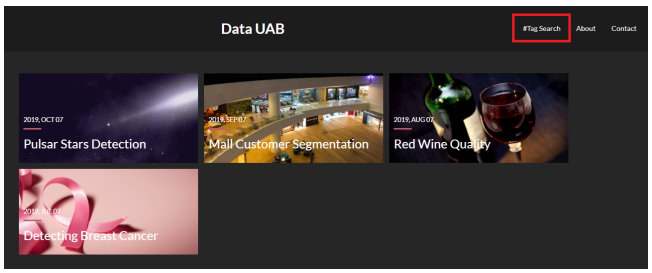


Fig. 17: Data UAB Blog with Posts and Link to Tags

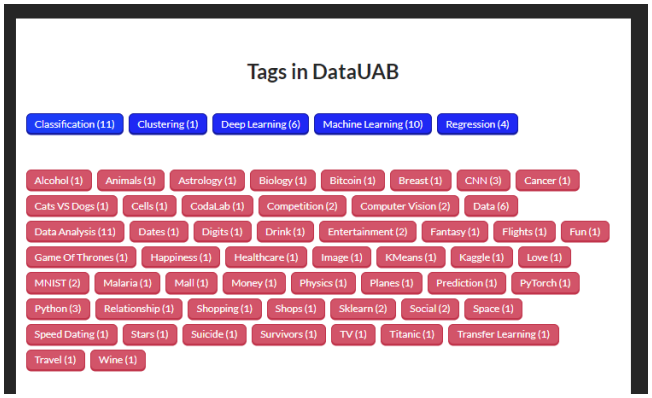


Fig. 18: Tag Page

- The **yellow** cluster groups young people with moderate to low annual income who actually spend a lot.
- The **purple** cluster groups reasonably young people with pretty decent salaries who spend a lot.
- The **pink** cluster groups people of all ages whose salary is not pretty high and their spending score is moderate.
- The **orange** cluster groups people who actually have pretty good salaries and barely spend money, their age usually lays between thirty and sixty years.
- The **blue** cluster groups those whose salary is pretty low and do not spend much money in stores, they are people of all ages.

4 DataUAB Blog

An important part of this project was to make aesthetic improvements in the DataUAB Blog [2], a blog-type web platform that, as previously stated, aims to encourage students to solve Machine Learning problems and acts as a bridge between students and companies of the field.

As far as the Blog improvements are concerned, the Figure 17 shows the four problems discussed in this paper after they have been uploaded, in HTML format, to the Blog so that everyone who wants to read them can. In addition, in the same figure, a link to the Blog tag page [18] can be seen inside the red square. The purpose of this tag page, which is observed in the Figure 18, is to group posts depending on the type of problem and topics they address.

The Blog code with all changes can be found in the DataUAB GitHub [19].

5 Conclusions

The main purpose of this project has been to conduct a study on four problems using different Machine Learning techniques. Likewise, improvements have also been made to the DataUAB Blog, creating new content and facilitating user interactions through the new tag system. On top of that, down below there are some conclusions drawn from each problem:

1. In Breast Cancer Detection [9], features that belong to "mean" and "worst" are the ones that provide most valuable information, especially: radius, area, perimeter, concave points and concavity, making redundant a good amount of variables.

2. Pulsar Stars [10] are hard to detect, with the majority of radio emissions detected belonging to noise or interference. Building an ANN (Artificial Neural Network) and setting higher weights for the positive class can greatly improve the chances of finding positives, regardless of whether they are true or negative positives.

3. In the Red Wine Prediction [14] there does not seem to be either very bad or good wines. However, if a wine contains higher alcohol percent, sulphates, citric acid while at the same time its volatile acidity is low, it is likely that it gets a higher qualification.

4. Mall Customer Segmentation [16] can be easily achieved using unsupervised learning techniques. Five groups of different kinds of people have been found in order to be able to plan a marketing strategy according to their annual income, age and spending behavior.

It is quite likely that the platform will be on the rise in the near future, and more students interested in Machine Learning and AI in general participate by creating new content with their own solutions to a wide variety of problems.

Acknowledgements

First of all, I would like to thank my tutor, Jordi Gonzàlez Sabaté, for the help and guidance he has given to me and the trust placed in me during these months this project has lasted.

Also, I would like to thank my family for the unconditional support they show every day, helping me to become a better person in all aspects. Finally, I would like to personally thank my friends, and former students, Laura Planas Simón, Elliot Ribas Garcia and Óscar Sánchez Bocero for being by my side every day during these last years, helping me in everything I needed.

References

[1] Dataversity, "A Brief History of Machine Learning": https://www.dataversity.net/a-brief-history-of-machine-learning/. [Last Visit: 2020-02-02].

[2] DataUAB Blog: https://datauab.github.io/. [Last Visit: 2020-02-02].

[3] Kaggle, "Breast Cancer Wisconsin (Diagnostic) Data Set": https://www.kaggle.com/uciml/breast-cancer-wisconsin-data. [Last Visit: 2020-02-02].

[4] Kaggle, "Predicting a Pulsar Star": https://www.kaggle.com/pavanraj159/predicting-a-pulsar-star. [Last Visit: 2020-02-02].

[5] Kaggle, "Red Wine Quality": https://www.kaggle.com/uciml/red-wine-quality-cortez-et-al-2009. [Last Visit: 2020-02-02].

[6] Kaggle, "Mall Customer Segmentation Data": https://www.kaggle.com/vjchoudhary7/customer-segmentation-tutorial-in-python. [Last Visit: 2020-02-02].

[7] Python, "Python 3 Documentation": https://docs.python.org/3/. [Last Visit: 2020-02-02].

[8] Jupyter Notebook: https://jupyter.org/. [Last Visit: 2020-02-02].

[9] F. Rodríguez Mir, "Detecting Breast Cancer": https://datauab.github.io/breast_cancer/. [Last Visit: 2020-02-02].

[10] F. Rodríguez Mir, "Pulsar Stars Detection": https://datauab.github.io/pulsar_stars/. [Last Visit: 2020-02-02].

[11] Keras, "Keras Documentation": https://keras.io/. [Last Visit: 2020-02-02].

[12] Jodrell Bank Centre for Astrophysics: http://www.jb.man.ac.uk/distance/frontiers/pulsars/section4.html. [Last Visit: 2020-02-02].

[13] Kaggle, "Solution by Pavan Raj": https://www.kaggle.com/pavanraj159/predicting-pulsar-star-in-the-universe. [Last Visit: 2020-02-02].

[14] F. Rodríguez Mir, "Red Wine Quality": https://datauab.github.io/red_wine_quality/. [Last Visit: 2020-02-02].

[15] Kaggle, "Solution by Vishalyu990": https://www.kaggle.com/vishalyo990/prediction-of-quality-of-wine. [Last Visit: 2020-02-02].

[16] F. Rodríguez Mir, "Mall Customer Segmentation": https://datauab.github.io/mall_segmentation/. [Last Visit: 2020-02-02].

[17] Scikit-yb, "Elbow Method": https://www.scikit-yb.org/en/latest/api/cluster/elbow.html. [Last Visit: 2020-02-02].

[18] DataUAB Tag Page: https://datauab.github.io/tags/. [Last Visit: 2020-02-02].

[19] DataUAB GitHub: https://github.com/DataUAB. [Last Visit: 2020-02-02].