**CCT College Dublin**

**Assessment Cover Page**

| **Module Title:** | Machine Learning & Data Preparation |
| --- | --- |
| **Student Full Name:** | Izaias de Oliveira Gomes Junior |
| **Lecturer Name:** | Dr. Muhammad Iqbal & David McQuaid |
| **Assessment Title:** | CA2 Project |
| **Assessment Due Date:** | 02nd January 2023 |
| **Date of Submission:** | 02nd January 2023 |
| **Student Number:** | 2023232 |

****

**Declaration**

| **By submitting this assessment, I confirm that I have read the CCT policy on Academic Misconduct and understand the implications of submitting work that is not my own or does not appropriately reference material taken from a third party or other source. I declare it to be my own work and that all material from third parties has been appropriately referenced. I further confirm that this work has not previously been submitted for assessment by myself or someone else in CCT College Dublin or any other higher education institution.** |
| --- |

[**Introduction 3**](#_3317zzcog7f)

[**Objective 4**](#_v5dkbdxabw6g)

[**Data Understanding 4**](#_3eceqdc2pfsk)

[Figure 1 - Target Variable Distribution 5](#_n1s7o330v2rk)

[Figure 2 - Age Distribution 6](#_mompc4deo93d)

[Figure 3 - Gender Distribution 6](#_lhkwkhcxp11w)

[Figure 4 - Boxplot with the highest concentration of outliers 7](#_pzv0zyk8mmnb)

[**Success Criteria 8**](#_dtz6ex69s6s9)

[**Dimensionality Reduction 9**](#_or0mnvkdqouz)

[**Imbalanced Data 10**](#_2e0sk1pql5tn)

[**Models - Linear Regression 10**](#_al7h82skthn)

[Table 1 - Regression with LDA and PCA 11](#_i9vwo2d3ltnm)

[Table 2 - Regression with LDA, PCA and Oversampling 12](#_1uumbf4h5x9n)

[Table 3 - Regression with LDA, PCA and Undersampling 13](#_jo6prske4wdm)

[**Models - Classification 14**](#_48ge0x9eh2me)

[Table 4 - Classification with LDA and PCA 15](#_yjbr7wwaat58)

[Table 5 - Regression with Original Data 15](#_6588ynt45wqk)

[Table 6 - Classification with Original Data 16](#_kd1xt1b0vpdl)

[**Models - Overall Analysis 17**](#_ul6fd45e455)

[Table 7 - Overall Analysis 17](#_507u1enfd7gj)

[**Conclusion 18**](#_knbzyd8gqzoj)

[**Reference 19**](#_jpkvvvgmrt2w)

[**Appendix 21**](#_88r5kwgbwojb)

# 

# Introduction

Health is a major concern among all societies, rich and poor, impacting on the quality of life, economic productivity and social stability. Since the beginning of time humanity has tried to overcome diseases and even death, looking for answers in nature and using all the available resources around.

The oldest record about medical conditions is an ancient Egyptian work produced by Imhotep and dated from around 3000 BC. Hippocrates was one of the first to suggest that diseases could have natural causes instead of supernatural (University of Glasgow, 2020). Data has been a key feature of the evolution of science, especially in the health field and The National Center for Health Statistics has contributed hugely for the american society through its exceptional work in gathering and organising data.

Health is not the same for all age groups and the predominance of diseases vary on different stages of life. It is important to study how health conditions and ageing are related to get a better understanding of how diseases behave with different ages so health authorities can draw prevention strategies and allocate resources to researchers for more effective treatments.

# Objective

The objective is to predict in which age group (Senior or Non-Senior) the respondents will be, given their health and nutritional information. Even though the problem can be dealt with Regression, Classification or Clustering, this project will focus on the first two and their performances will be compared. Two dimensionality reduction techniques will be performed - PCA and LDA - and two methods for imbalanced data - SMOTE and Near-Miss.

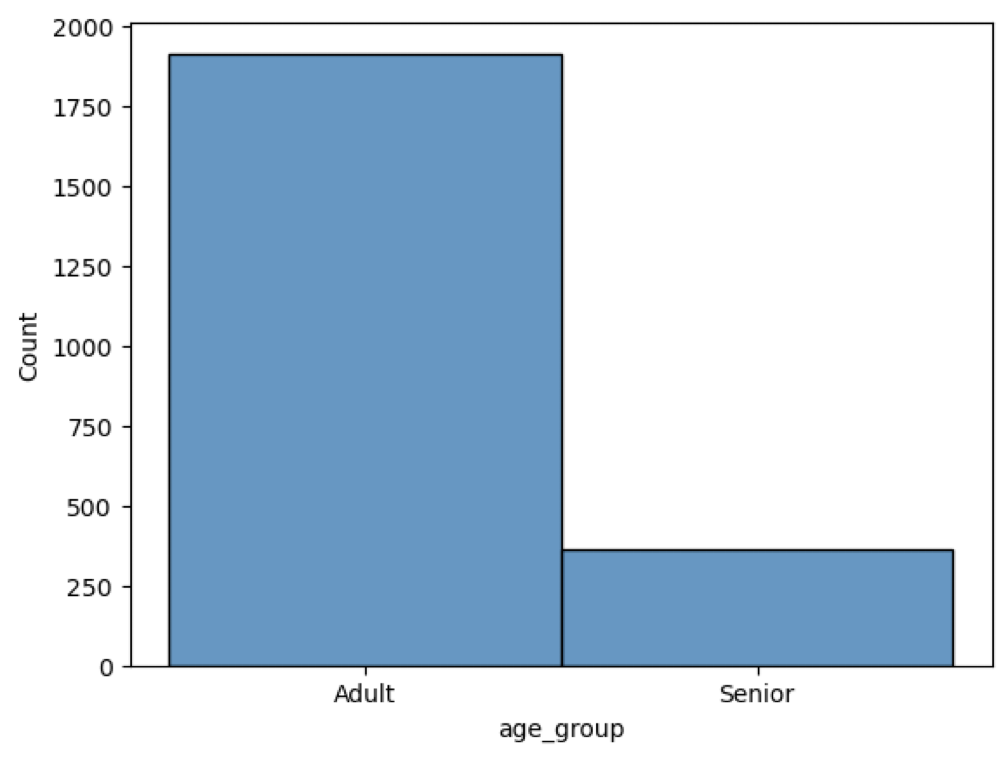
# 

# Data Understanding

The National Center for Health Statistics (NCHS) at the Center for Disease Control and Prevention (CDC) is responsible for collecting a wide-ranging set of nutritional and health information through surveys across The United States territory. The sub-dataset provided for this project was extracted in the UC Irvine Machine Learning Repository with DOI [10.24432/C5BS66](https://doi.org/10.24432/C5BS66) and is licensed under a Creative Commons Attribution 4.0 International (CC BY 4.0) licence. It contains 2278 observations and 10 features related to lifestyle choices and biochemical markers (such as Blood Insulin Levels) and they were selected based on their hypothetical correlation with age, which is the target variable labelled into two categories.

The dataset has no missing values and the column “SEQN”, which contain the respondent sequence number, was dropped for its insignificance for the analysis. This information is only useful for gathering and organising the dataset during the survey fase. The target variable in “age\_group” is divided into two groups: individuals over 65 years were labelled as “senior” while the “non-senior” contains the individuals under 65 years. The One-Hot Encoding was used to transform the target variable from categorical to numerical to avoid to force an ordinal relationship between the groups (Brownlee, 2020). This method create two new features where “age\_goup\_adult" will be 1 when the "adult" value is true and 0 when is false and “age\_group\_Senior” will be the opposite. The new variable “age\_group\_Senior” will be dropped so we have just one target variable and the Machine Learning models can perform their mathematical procedures.

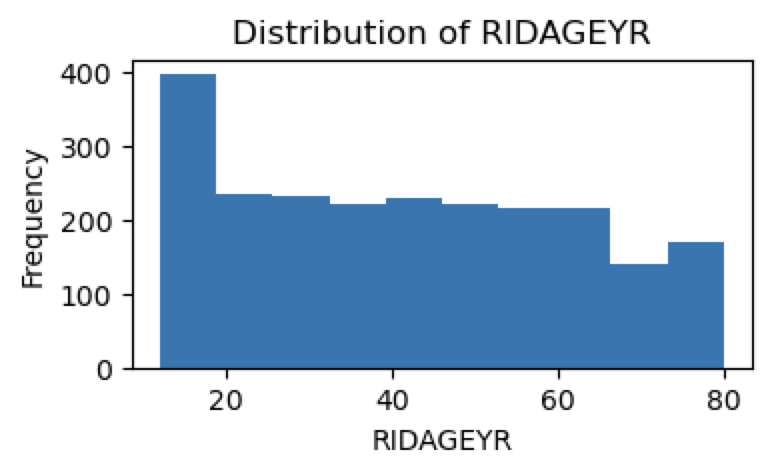
## Figure 1 - Target Variable Distribution



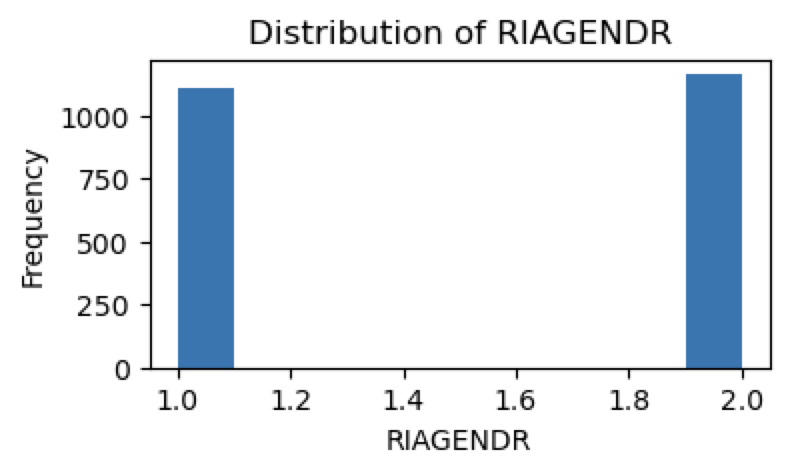
The graphic above shows that the target variable is imbalanced where the amount of “Adult” represents 84.02% against 15.98% of the “Senior” group. The imbalance of the dataset might affect the performance of the Machine Learning algorithms, when the minority class tends to be ignored over the majority leading to an overfitting model (Truong, 2022). There are two ways to make the number of observations in the groups equal: by creating synthetic data of the minority group or reducing the number of the majority group (Imarticus, 2021). Both ways will be explored through the SMOTE (Synthetic Minority Oversampling Technique) and Near-Miss techniques.

The column “RIDAGEYR” contains the ages of the respondents, the distribution is not normal, even though the media and the median are the same, with a higher concentration on the left side. The average is 41 years old with a standard deviation of approximately 20, the youngest person is 12 years old and the oldest is 80. The dataset is quite balanced for genders, with 1113 males and 1165 females.

## Figure 2 - Age Distribution

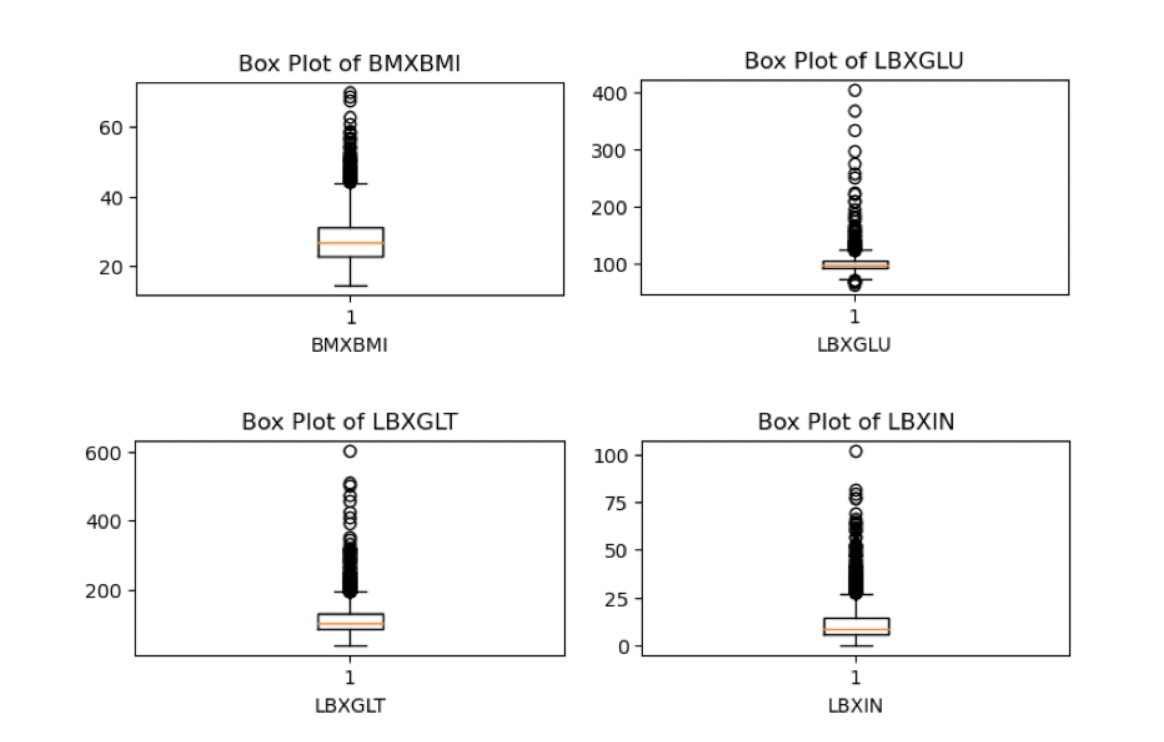


## Figure 3 - Gender Distribution



The presence of outliers are significant in four columns: “BMXBMI” (Body Mass Index), “LBXGLU” (Blood Glucose after fasting), “LBXGLT” (Oral Health) and “LBXIN” (Blood Insulin Levels).

## Figure 4 - Boxplot with the highest concentration of outliers



# Success Criteria

The success of the regression models can be evaluated using the following methods: the R-squared (R2), the Adjusted R-squared, Mean Squared Error (MSE), the Cross-Validation Score, the comparison between the MSE on the train and on the test data and Huber Loss, from Sklearn and Tensorflow libraries (Pedregosa et al., 2011) (TensorFlow Developers, 2023).

The R-squared is usually used to measure and to indicate the quality of a regression model, it is a proportion of the variance in the target variable that is explained by the features of the model (Frost, 2018). Meanwhile the Adjusted R-squared takes into consideration the numbers of features used in the model and measures if adding additional features would be better, penalising if it does not (Bhandari, 2020). The MSE is a method to calculate the error with the squared difference between the actual and the predicted value (M, 2021).

The Cross-Validation Score is a helpful technique to reduce bias and to avoid overfitting by estimating how well the model will perform and generalise in new unseen data (Brownlee, 2018). The comparison between the MSE on the train test data is also helpful to check overfitting by having those two as close as possible (Bishop, 2006). The Huber Loss has the same goal as the MSE and it is normally used in a dataset with the presence of outliers (M, 2021).

When dealing with classification models the success will be measured by the Accuracy score, Precision, Recall, f1-score and Cross-Validation Accuracy. The Accuracy score is the ratio of the total of corrected predictions to the total predictions of the model. It is a very straightforward concept and it is usually visualised with the help of a confusion matrix, which summarises the model’s predictions and organises in a table by class with the correctly predicted values and their errors (Brownlee, 2019).

Precision and Recall are two simple and useful score measures. The first one takes the true positives and divides them for all the positives while the second one is the true positives divided by the sum of the true positives and the false negatives, giving us a ratio of the true predictions by the model (Huilgol, 2020). The f1-score is the harmonic mean between precision and recall as a higher value indicating a better performance considering both precision and recall (Pedregosa et al., 2019).

The ultimate goal of any machine learning algorithm is to achieve such a performance that predicts new and unseen data as it did on the original one. Most of the time this unseen data is not available in such cases the train test split for validation is a good choice to test if the model generalises well (Galarnyk, 2022). It is a good practice to experiment with different sizes of train test split, in this project the percentages that will be used are: 10%, 20% and 30%.

A tuning in the hyperparameters of the algorithms are also important in the pursuit of the best model. There are several ways to do it but the one chosen for this project was the Grid Search from Sklearn library. After specifying the range for the hyperparameters, the method tries different combinations and calculates their performance; it might be time-consuming depending on the number of hyperparameters that is being tested (Shah, 2021).

# Dimensionality Reduction

Principal Component Analysis and Linear Discriminant Analysis are both dimensionality reduction techniques that help with complex datasets. PCA is an unsupervised machine learning useful to deal with the curse of dimensionality, which is the difficulty to get meaningful patterns within the data due its large numbers of features (dimensions). This method is commonly used when it is important to reduce the amount of time that the algorithm takes to process high-dimensional data and also reduces the amount of noise by reducing the overfitting and improving the generalisation of the model. Its importance lays down on the simplicity of its applicability and the useful resource of keeping as much of the original variance as possible (S, 2022).

Another dimensionality reduction technique that will be implemented in this project is Linear Discriminant Analysis. LDA is generally used in classification problems for feature extraction patterns. The idea is to reduce the number of dimensions by reducing the variability within classes while maximising it between the classes. Different from PCA, LDA is a supervised method which considers the classes labels to identify the directions that best discriminate them (Dash, 2021).

It is important to test both techniques in different models and compare their results through the success criteria chosen.

# Imbalanced Data

The imbalance in the target variable of a dataset might cause overfitting and bias on the success criteria. This problem can be dealt with by equalising both classes through the augmentation of the minor class or the reduction of the majority.

The Synthetic Minority Over-Sampling Technique, or SMOTE, balances the dataset by generating synthetic observations of the minority class with its nearest neighbours. While this technique is compatible with various machine learning algorithms and simple to use, the synthetic data might have the opposite desirable effect and generate an overfitting model (Brownlee, 2020b).

Undersampling methods do the opposite as described above, in order to balance the dataset they remove part of the observations from the majority class. Although it is an interesting technique it should be handled carefully because useful information might be lost due to the randomness of the process. The Near-Miss is one of the methods available and works based on the distance between the majority class examples to the minority class (Brownlee, 2020c).

Both techniques are worthwhile testing as experiment is key to good science, however a closer look into the results are demanded to check if they are reliable.

# Models - Linear Regression

For Regression 07 models were performed:

* Linear Regression;
* Linear Regression Polynomial Second Degree;
* Random Forest;
* Support Vector Regression;
* K-Nearest Neighbours;
* Bagging Regressor with KNN;
* Decision Tree.

When applying LDA all the models performed better with a 10% split and the best result was KNN with a R2 of 0.9221, Cross-Validation score of 0.8622 and no overfitting as the difference between the MSE on the train and on the test was small.

For PCA 04 out of 07 models performed better with a 10% split and the other 03 with a 30% split. The best performance overall was Random Forest with R2 of 0.9895 and 10% split, followed by Decision Tree with 30% split and R2 of 0.9738. Even though the Grid Search was used to find the best hyperparameters in both cases, the first model cannot be used due its small Adjusted R2 of -0.229 and big difference between the R2. The second model is more reliable with no overfitting, no difference between R2 and Adjusted R2 and a Cross-Validation score of 0.9473.

Overall PCA had better results than LDA in 4 models: Random Forest, KNN, Bagging Regressor with KNN and Decision Tree. Linear Regression had the worst performance in all scenarios but improved considerably with its derivation when Polynomial with second degree is applied.

## Table 1 - Regression with LDA and PCA

| LDA | | | |  | PCA | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 10% | 20% | 30% | BEST |  | 10% | 20% | 30% | BEST |
| R2 LN: 0.5618881510394962 | R2 LN: 0.5041900982150748 | R2 LN: 0.4927866255820207 | 10% |  | R2 LN: 0.5282121813594496 | R2 LN: 0.4753742755353033 | R2 LN: 0.46568286690897653 | 10% |
| R2 LNP2: 0.8071489417811208 | R2 LNP2: 0.7612939803295771 | R2 LNP2: 0.7510945668752906 | 10% |  | R2 LNP2: 0.7993926928925605 | R2 LNP2: 0.7795515454599702 | R2 LNP2: 0.7698297738540906 | 10% |
| R2 RF: 0.8957097930140704 | R2 RF: 0.8624796667270428 | R2 RF: 0.8547897694606474 | 10% |  | R2 RF: 0.9895033026113671 | R2 RF: 0.9735693971063258 | R2 RF: 0.970571667544113 | 10% |
| R2 SVR: 0.8945388097791866 | R2 SVR: 0.8352821121231669 | R2 SVR: 0.8334051561957369 | 10% |  | R2 SVR: 0.8652449665458792 | R2 SVR: -1.0195804195804197 | R2 SVR: 0.8354034987443396 | 10% |
| R2 KNN: 0.9221710189452125 | R2 KNN: 0.8639440559440559 | R2 KNN: 0.8606565562216599 | 10% |  | R2 KNN: 0.9545997610513739 | R2 KNN: 0.942916342916343 | R2 KNN: 0.959343104121733 | 30% |
| R2 KNN BGG: 0.9027336280134173 | R2 KNN BGG: 0.8567059607059607 | R2 KNN BGG: 0.8606001870884047 | 10% |  | R2 KNN BGG: 0.9378162324837769 | R2 KNN BGG: 0.9292749155606298 | R2 KNN BGG: 0.9429404765659469 | 30% |
| R2 DT: 0.8603816908227705 | R2 DT: 0.8268961504398012 | R2 DT: 0.8420032937063441 | 10% |  | R2 DT: 0.9686922003824572 | R2 DT: 0.9626003626003626 | R2 DT: 0.9738634240782569 | 30% |

As mentioned before, the target variable is imbalanced and there are two ways to deal with it: oversampling the minority class or undersampling the majority one. Both techniques were performed with LDA and PCA and the best results, considering only the R2, were obtained with oversampling, the highlights are: Random Forest, K-Nearest Neighbours, Bagging Regressor with KNN and Decision Tree with R2 higher than 0.97 in all three splits with PCA. These models did not show overfitting when the Training Mean Square Errors and the Validation MSE were compared but they all performed poorly for Cross-Validation with a score no higher than 0.58, which might indicate that the models do not generalise well to new unseen data. The results for the Cross-Validation score with undersampling were even lower, with LDA and PCA, with some being negative.

## Table 2 - Regression with LDA, PCA and Oversampling

| LDA - Over Sampling | | | |  | PCA - Over Sampling | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 10% | 20% | 30% | BEST |  | 10% | 20% | 30% | BEST |
| R2 LN: 0.7635925495609067 | R2 LN: 0.7475267940774699 | R2 LN: 0.7439193356990967 | 10% |  | R2 LN: 0.7369995676749912 | R2 LN: 0.7373150824242898 | R2 LN: 0.7286196676676804 | 20% |
| R2 LNP2: 0.8283614540695288 | R2 LNP2: 0.7950646052829744 | R2 LNP2: 0.8028222702205406 | 10% |  | R2 LNP2: 0.8230878657126647 | R2 LNP2: 0.8217220680819086 | R2 LNP2: 0.8177087367073441 | 10% |
| R2 RF: 0.9063086092105765 | R2 RF: 0.8788981374045148 | R2 RF: 0.8846655331310294 | 10% |  | R2 RF: 0.9906451612903225 | R2 RF: 0.9886071871009943 | R2 RF: 0.9868253859348199 | 10% |
| R2 SVR: 0.9151153467813481 | R2 SVR: 0.8904595217125015 | R2 SVR: 0.8879652120337375 | 10% |  | R2 SVR: 0.0 | R2 SVR: 0.0 | R2 SVR: 0.0 |  |
| R2 KNN: 0.924826762246117 | R2 KNN: 0.8906438746438746 | R2 KNN: 0.9045092433771679 | 10% |  | R2 KNN: 0.98805256869773 | R2 KNN: 0.9851851851851852 | R2 KNN: 0.9824661711454165 | 10% |
| R2 KNN BGG: 0.9025916172920782 | R2 KNN BGG: 0.884436420722135 | R2 KNN BGG: 0.8943396226415095 | 10% |  | R2 KNN BGG: 0.9816282642089094 | R2 KNN BGG: 0.9849345892203035 | R2 KNN BGG: 0.9824937865369132 | 20% |
| R2 DT: 0.8941198420825917 | R2 DT: 0.8528596405798787 | R2 DT: 0.8561793160501606 | 10% |  | R2 DT: 0.9885752688172043 | R2 DT: 0.9763532763532764 | R2 DT: 0.9858967028778349 | 10% |

# 

# 

# 

# 

# 

# 

## Table 3 - Regression with LDA, PCA and Undersampling

| LDA - Under Sampling | | | |  | PCA - Under Sampling | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 10% | 20% | 30% | BEST |  | 10% | 20% | 30% | BEST |
| R2 LN: 0.7524543977902298 | R2 LN: 0.6786802216427132 | R2 LN: 0.663615555682719 | 10% |  | R2 LN: 0.7214445089990282 | R2 LN: 0.7400765044385769 | R2 LN: 0.7305304112238034 | 20% |
| R2 LNP2: 0.7277140678345341 | R2 LNP2: 0.6956373357267776 | R2 LNP2: 0.6859051023776452 | 10% |  | R2 LNP2: 0.6476387214490104 | R2 LNP2: 0.7690086279751207 | R2 LNP2: 0.7559104093017317 | 20% |
| R2 RF: 0.8201000619402062 | R2 RF: 0.7501886326820648 | R2 RF: 0.7603797062261229 | 10% |  | R2 RF: 0.965654335451356 | R2 RF: 0.9581946807072087 | R2 RF: 0.9468944805747759 | 10% |
| R2 SVR: 0.8567145391833357 | R2 SVR: 0.7558895475676481 | R2 SVR: 0.7681677863063485 | 10% |  | R2 SVR: 0.0 | R2 SVR: 0.0 | R2 SVR: 0.0 |  |
| R2 KNN: 0.870899470899471 | R2 KNN: 0.7671380471380471 | R2 KNN: 0.7624642464246425 | 10% |  | R2 KNN: 0.9470899470899471 | R2 KNN: 0.9595959595959596 | R2 KNN: 0.9361936193619362 | 20% |
| R2 KNN BGG: 0.8411467444120505 | R2 KNN BGG: 0.7382374768089053 | R2 KNN BGG: 0.768729036168923 | 10% |  | R2 KNN BGG: 0.9312925170068027 | R2 KNN BGG: 0.9428509585652443 | R2 KNN BGG: 0.9358698727015559 | 20% |
| R2 DT: 0.7369733290141454 | R2 DT: 0.7030605134871009 | R2 DT: 0.731904057185764 | 10% |  | R2 DT: 0.9828042328042328 | R2 DT: 0.9663299663299664 | R2 DT: 0.9603960396039604 | 10% |

# 

# Models - Classification

For Classification 07 models were performed:

* Logistic Regression;
* Decision Tree;
* Random Forest;
* Support Vector Machine;
* K-Nearest Neighbours;
* Naive Bayes;
* Artificial Neural Networks.

The LDA and PCA were applied again but with no over and under sampling techniques once all the accuracy scores without them were higher than 0.95. The 10% split was the most common but the differences were not so significant, suggesting that the models were stable with different train test splits. The Cross-Validation score was close to the accuracy score in all scenarios.

The presence of overfitting was evaluated by the difference between precision and recall for both classes, in this case the models with PCA performed better with almost no difference while for the ones with LDA the range of the difference was between 0.07 and 0.15. The best models with LDA, with high accuracy and low overfitting, were Logistic Regression, Support Vector Machine and K-Nearest Neighbours. When PCA is applied the best performances were obtained with Decision Tree and Random Forest.

## Table 4 - Classification with LDA and PCA

| LDA | | | |  | PCA | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 10% | 20% | 30% | BEST |  | 10% | 20% | 30% | BEST |
| Accuracy\_lg: 0.9868 | Accuracy\_lg: 0.9759 | Accuracy\_lg: 0.9781 | 10% |  | Accuracy\_lg: 0.9781 | Accuracy\_lg: 0.9868 | Accuracy\_lg: 0.9898 | 30% |
| Accuracy\_dt: 0.9605 | Accuracy\_dt: 0.9627 | Accuracy\_dt: 0.9561 | 20% |  | Accuracy\_dt: 1.0000 | Accuracy\_dt: 0.9956 | Accuracy\_dt: 0.9971 | 10% |
| Accuracy\_rf: 0.9605 | Accuracy\_rf: 0.9649 | Accuracy\_rf: 0.9605 | 20% |  | Accuracy\_rf: 1.0000 | Accuracy\_rf: 0.9978 | Accuracy\_rf: 0.9956 | 10% |
| Accuracy\_svm: 0.9868 | Accuracy\_svm: 0.9759 | Accuracy\_svm: 0.9781 | 10% |  | Accuracy\_svm: 0.9956 | Accuracy\_svm: 0.9956 | Accuracy\_svm: 0.9956 | TODOS |
| Accuracy\_knn: 0.9825 | Accuracy\_knn: 0.9737 | Accuracy\_knn: 0.9795 | 10% |  | Accuracy\_knn: 0.9956 | Accuracy\_knn: 0.9956 | Accuracy\_knn: 0.9942 | 10% ou  20% |
| Accuracy\_gnb: 0.9825 | Accuracy\_gnb: 0.9737 | Accuracy\_gnb: 0.9751 | 10% |  | Accuracy\_gnb: 0.9825 | Accuracy\_gnb: 0.9912 | Accuracy\_gnb: 0.9912 | 20% ou  30% |
| Accuracy\_ann: 0.9868 | Accuracy\_ann: 0.9781 | Accuracy\_ann: 0.9766 | 10% |  | Accuracy\_ann: 0.9912 | Accuracy\_ann: 0.9978 | Accuracy\_ann: 0.9927 | 20% |

The last experiment performed was with the original data with no dimensionality reduction or sampling techniques. For Regression all models got R2 smaller than the best ones in the previous scenarios, the exceptions were Random Forest and Decision Tree. Although the first one got a R2 of 1.00, the Adjusted R2 was -0.014 making the model not a good choice. However, the second one got a 1.00 score for R2, Adjusted R2 and Cross-Validation with no presence of overfitting across all train test splits.

## Table 5 - Regression with Original Data

| Regression | | |
| --- | --- | --- |
| 10% | 20% | 30% |
| R2 LN: 0.5618881510394962 | R2 LN: 0.5041900982150747 | R2 LN: 0.49278662558202013 |
| R2 LNP2: 0.8051823695564234 | R2 LNP2: 0.7865151627989853 | R2 LNP2: 0.7737980964040417 |
| R2 RF: 1.0 | R2 RF: 1.0 | R2 RF: 1.0 |
| R2 SVR: 0.8082321417670297 | R2 SVR: 0.7962631481558756 | R2 SVR: 0.7923535186535011 |
| R2 KNN: 0.9069943676395289 | R2 KNN: 0.9097684537684537 | R2 KNN: 0.9063471630181886 |
| R2 KNN BGG: 0.9039487131258033 | R2 KNN BGG: 0.9000622234907949 | R2 KNN BGG: 0.9161898985431851 |
| R2 DT: 1.0 | R2 DT: 1.0 | R2 DT: 1.0 |

Once more the classification models performed better than the regression models on an overall view with scores higher than 0.96. Decision Tree and Random Forest were the best ones again but this time both of them got good results in all scores: accuracy, precision, recall, Cross-Validation and no overfitting.

## Table 6 - Classification with Original Data

| Classification | | |
| --- | --- | --- |
| 10% | 20% | 30% |
| Accuracy\_lg: 0.9781 | Accuracy\_lg: 0.9868 | Accuracy\_lg: 0.9868 |
| Accuracy\_dt: 1.0000 | Accuracy\_dt: 1.0000 | Accuracy\_dt: 1.0000 |
| Accuracy\_rf: 1.0000 | Accuracy\_rf: 1.0000 | Accuracy\_rf: 1.0000 |
| Accuracy\_svm: 0.9956 | Accuracy\_svm: 0.9956 | Accuracy\_svm: 0.9956 |
| Accuracy\_knn: 0.9868 | Accuracy\_knn: 0.9934 | Accuracy\_knn: 0.9927 |
| Accuracy\_gnb: 0.9649 | Accuracy\_gnb: 0.9715 | Accuracy\_gnb: 0.9591 |
| Accuracy\_ann: 0.9912 | Accuracy\_ann: 0.9912 | Accuracy\_ann: 0.9708 |

# Models - Overall Analysis

In an overview with the best model for each combination taking into account just the two main success criteria - R2 for regression and accuracy for classification - the most recurrent ones are: K-Nearest Neighbours, Decision Tree and Random Forest. A more in depth analysis, considering others success criteria, was debated on the previous section leaving this one to explain about the highlighted models.

## Table 7 - Overall Analysis

| Overall Analysis | |
| --- | --- |
| Combination | Model |
| Regression and LDA | KNN |
| Regression and PCA | Decision Tree |
| Regression, LDA and Undersampling | KNN |
| Regression, PCA and Undersampling | Decision Tree |
| Regression, LDA and Oversampling | KNN |
| Regression, PCA and Oversampling | Random Forest |
| Classification and LDA | Logistic Regression, Support Vector Machine and Artificial Neural Networks |
| Classification and PCA | Decision Tree and Random Forest |
| Regression with Original Data | KNN |
| Classification with Original Data | Decision Tree and Random Forest |

The K-Nearest Neighbours is known for using proximity in order to make classifications or predictions with the assumption that similar points are found close to each other. The Euclidean distance is often used in both regression and classification models. In the first case, the average of the distance of the continuous values are used while for classification the distance between the data points and the class label. KNN is easy to implement due its simplicity with few hyperparameters. The disadvantages are the proneness to overfitting and the curse of dimensionality. Its usage has been broad, for example: on finances (risk of a loan), healthcare (risk of a heart attack) and pattern recognition (handwriting) (IBM, 2022).

Decision Tree is another supervised machine learning algorithm that can be used for classification and regression problems. Through a built flowchart, the algorithm splits the training data into subsets in accordance with the attributes values and just stops with the set criteria, for example: maximum depth and minimum number of samples. The entropy or Gini impurity is used to help to optimise the model, maximising information gain and minimising impurity. Its usage is very popular due the similarity with the human process of decision making and due the power of thinking in all possible outcomes. Decision Tree often faces the overfitting problem and computational complexity for a broader range of class labels, which can be dealt with using Random Forest (GeeksForGeeks, 2017).

Random Forest is an aggregation of Decision Trees that reaches one single result by combining their outputs. Classification problems are solved with the most frequent class while regression problems are solved with the computed mean of predictions, in both cases the random feature selection is used at each node of the decision tree bringing diversity to the model. The often high accuracy and low overfitting make this algorithm a powerful one, even though with its computational expensiveness and less interpretability when compared with a Decision Tree (IBM, 2023).

# Conclusion

On average the models performed better with a 10% train test split, with higher R2 or accuracy, smaller difference between the Training and Validation MSE (no overfitting), good Cross-Validation score and a lower Huber Loss. All the experiments performed in the pursuit of the best models and their results can be found in separated Jupyter Notebooks or in the Github Repository, the links are provided in the main file and on the appendix.

A total of 14 machine learning algorithms were tested, 07 for regression and 07 for classification, along with two dimensionality reduction methods and two techniques to balance the target variable. Among all the combinations made, Decision Tree and Random Forest were the ones with the highest frequency of good results. Even though LDA is more appropriate for classification problems, this method was tested on regression algorithms with the hypothesis that the results would be considerably low but it turned out to be surprisingly good. The results with PCA were predominantly better than the ones with LDA among all combinations. The imbalance of the dataset was not a crucial problem once the models without the balance techniques performed better in all success criteria.

The classification algorithms outperformed the regression ones and one of the hypotheses is the low correlation between the variables. As the target variable is separated into two classes and the values are not continuous, a classification approach might be more straightforward in regards to the relationship between the independent and the dependent variables.

# Reference List

Bhandari, A. (2020). Key Difference between R-squared and Adjusted R-squared for Regression Analysis. [online] Analytics Vidhya. Available at:<https://www.analyticsvidhya.com/blog/2020/07/difference-between-r-squared-and-adjusted-r-squared/#:~:text=R2%20represents%20the%20proportion%20of>.

Bishop, C.M. (2006). Pattern Recognition and Machine Learning. Springer.

Brownlee, J. (2018). A Gentle Introduction to k-fold Cross-Validation. [online] Machine Learning Mastery. Available at: [https://machinelearningmastery.com/k-fold-Cross-Validation/](https://machinelearningmastery.com/k-fold-cross-validation/).

Brownlee, J. (2019). Failure of Classification Accuracy for Imbalanced Class Distributions. [online] Machine Learning Mastery. Available at:<https://machinelearningmastery.com/failure-of-accuracy-for-imbalanced-class-distributions/#:~:text=Classification%20accuracy%20is%20a%20metric>.

Brownlee, J. (2020a). Ordinal and One-Hot Encodings for Categorical Data. [online] Machine Learning Mastery. Available at:<https://machinelearningmastery.com/one-hot-encoding-for-categorical-data/>.

Brownlee, J. (2020b). SMOTE for Imbalanced Classification with Python. [online] Machine Learning Mastery. Available at:<https://machinelearningmastery.com/smote-oversampling-for-imbalanced-classification/>.

Brownlee, J. (2020c). Undersampling Algorithms for Imbalanced Classification. [online] Machine Learning Mastery. Available at:<https://machinelearningmastery.com/undersampling-algorithms-for-imbalanced-classification/>.

Dash, S.K. (2021). Linear Discriminant Analysis | What is Linear Discriminant Analysis. [online] Analytics Vidhya. Available at:<https://www.analyticsvidhya.com/blog/2021/08/a-brief-introduction-to-linear-discriminant-analysis/>.

Frost, J. (2018). How To Interpret R-squared in Regression Analysis. [online] Statistics By Jim. Available at:<https://statisticsbyjim.com/regression/interpret-r-squared-regression/>.

Galarnyk, M. (2022). Train Test Split: What it Means and How to Use It | Built In. [online] builtin.com. Available at:<https://builtin.com/data-science/train-test-split>.

GeeksForGeeks (2017). Decision Tree - GeeksforGeeks. [online] GeeksforGeeks. Available at:<https://www.geeksforgeeks.org/decision-tree/>.

Huilgol, P. (2020). Precision vs Recall | Precision and Recall Machine Learning. [online] Analytics Vidhya. Available at:<https://www.analyticsvidhya.com/blog/2020/09/precision-recall-machine-learning/>.

IBM (2022). What is the k-nearest neighbors algorithm? | IBM. [online] www.ibm.com. Available at:<https://www.ibm.com/topics/knn#:~:text=Related%20solutions->.

IBM (2023). What is Random Forest? | IBM. [online] www.ibm.com. Available at:<https://www.ibm.com/topics/random-forest#:~:text=Random%20forest%20is%20a%20commonly>.

Imarticus (2021). Using Near-Miss Algorithm For Imbalanced Datasets! [online] Finance, Tech & Analytics Career Resources | Imarticus Blog. Available at:<https://blog.imarticus.org/using-near-miss-algorithm-for-imbalanced-datasets/>.

M, P. (2021). A Comprehensive Introduction to Evaluating Regression Models. [online] Analytics Vidhya. Available at:<https://www.analyticsvidhya.com/blog/2021/10/evaluation-metric-for-regression-models/#:~:text=Mean%20Absolute%20Percentage%20Error%20(MAPE)> [Accessed 25 Nov. 2023].

Pedregosa et al. (2019). sklearn.metrics.f1\_score — scikit-learn 0.21.2 documentation. [online] Scikit-learn.org. Available at:<https://scikit-learn.org/stable/modules/generated/sklearn.metrics.f1_score.html>.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M. and Duchesnay, É. (2011). Scikit-learn: Machine Learning in Python. Journal of Machine Learning Research, [online] 12(85), pp.2825–2830. Available at:<https://jmlr.csail.mit.edu/papers/v12/pedregosa11a.html>.

S, P. (2022). An Introductory Note on Principal Component Analysis. [online] Analytics Vidhya. Available at:<https://www.analyticsvidhya.com/blog/2022/07/principal-component-analysis-beginner-friendly/>.

Shah, R. (2021). GridSearchCV |Tune Hyperparameters with GridSearchCV. [online] Analytics Vidhya. Available at:<https://www.analyticsvidhya.com/blog/2021/06/tune-hyperparameters-with-gridsearchcv/>.

Truong, A. (2022). Imbalanced Data ML: SMOTE and its variants. [online] TotalEnergies Digital Factory. Available at:<https://medium.com/totalenergies-digital-factory/imbalanced-data-ml-smote-and-its-variants-c69a4b32f7e7>.

University of Glasgow (2020). A Brief History Of Medicine. [online] FutureLearn. Available at:<https://www.futurelearn.com/info/courses/study-medicine/0/steps/147884#:~:text=We%20do%20know%20that%20from>.

# Appendix

All the experiments performed in the pursuit of the best models and their results can be found in separated Jupyter Notebooks or in the Github Repository, the links are provided in the main file.

LDA with Regression Algorithms: <http://localhost:8888/notebooks/Documents/GitHub/ML/LDA.ipynb>

LDA with Classification Algorithms: <http://localhost:8888/notebooks/Documents/GitHub/ML/LDA_Classification.ipynb>

LDA with Oversampling and Regression: <http://localhost:8888/notebooks/Documents/GitHub/ML/LDA_Over.ipynb>

LDA with Undersampling and Regression: <http://localhost:8888/notebooks/Documents/GitHub/ML/LDA_Under.ipynb>

PCA with Regression Algorithms: <http://localhost:8888/notebooks/Documents/GitHub/ML/PCA.ipynb>

PCA with Classification Algorithms: <http://localhost:8888/notebooks/Documents/GitHub/ML/PCA_Classification.ipynb>

PCA with Oversampling and Regression: <http://localhost:8888/notebooks/Documents/GitHub/ML/PCA_Over.ipynb>

PCA with Undersampling and Regression: <http://localhost:8888/notebooks/Documents/GitHub/ML/PCA_Under.ipynb>

Original Data with Regression and Classification Algorithms: <http://localhost:8888/notebooks/Documents/GitHub/ML/Original%20Data.ipynb>

Summary of R2 and Accuracy of all Models: <https://docs.google.com/spreadsheets/d/1_jueJid0ULbd2rgHKxMRD1Jy-Ke8bujUO5ozr00IkL4/edit#gid=0>

Github Repository: <https://github.com/izazaka/ML>