A logo for college computing

Description automatically generated

**Assessment Cover Page**

|  |  |
| --- | --- |
| *Student Full Name: Cristhian Elson Pereira Macedo* |  |
| *Student Number: 2024104* |  |
| *Module Title: Data Preparation - Machine Learning* |  |
| *Assessment Title: Integrated CA2* |  |
| *Assessment Due Date: 24th May 2024* |  |
| *Date of Submission: 24th May 2024* |  |

**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.



**Health Care**

**Integrated CA2 - Data Preparation and Machine Learning**

**Cristhian Elson Pereira Macedo**

**2024104**

Higher Diploma in Science in Data Analytics for Business

Data Preparation (DPDA) and Machine Learning (MLDA)

Lecturer: David McQuaid and Dr. Muhammad Iqbal

CCT College Dublin

Dublin, Ireland

2024



**SUMMARY**

[**1.** **Introduction** 4](#_Toc167487422)

[**2.** **Data UNDERSTANDING** 4](#_Toc167487423)

[**3.** **Data preparation** 4](#_Toc167487424)

[**4.** **Machine Learning** 7](#_Toc167487425)

[**References** 11](#_Toc167487426)

# **Introduction**

This project aims to complete the second Assessment Task for the module Data Preparation and Machine Learning from the course Higher Diploma in Science in Data Analytics for Business by CCT College Dublin, this is a document to describe the project itself whose name is “Hdip\_CristhianMacedo\_2024.ipynb” and should be always with this document. This Data Preparation and Machine Learning project uses the programming language Python, the environment of Anaconda Navigator with Jupyter Notebook, with CRISP-DM methodology as project management.

# **Data UNDERSTANDING**

After getting and loading the database using the Pandas library using “.head()” it was possible to get the first impression of the data, returning that this dataset has 15 columns (features) and after using “.shape” returning the shape of it, in this case, 5000 rows (observations) to be analysed.

Also used “.info()” to get a summary of the Data Frame, and next “.describe()” to see statistical information of the numerical features, in this case, having 10 numerical and 5 categorical. Checked the missing values and plotted them in a graphic using the “missingno” library, also, checked the existence of duplicate values, and in this case, there are no duplicates to be dealt with and checked unique values, got an overview of all data set.

# **Data preparation**

Until this moment there are no features to be dropped, it was renamed one feature, because the others it were well typed already, replaced values (Syntax Errors), the feature "Disease" when it appears NaN values could not be considered as a "True Missing Value", but a patient diagnosed with no disease, that is, there is no type of disease analysed for that specific patient, so this approach it will be considered as a Syntax Error and be fixed, according to the result obtained from the command ".unique()".

Handled with missing values, in the features “BMI, BloodPressure and Cholesterol” using the method “bfill” instead of the "Median" to get values from the next below rows, even Median being a better representation of the majority of the values in the variable, no duplicate rows do be dropped, encoded the features: PhysicalActivity, IncomeLevel, StressLevel, HealthIns and Disease. There is no type conversion to be dealt with, all features are numerical already.

Next, in Exploratory Data Analysis, it was created a heatmap of all features, a pair plot, figured out that the features “Glucose, Insulin and HeartRate” have outliers, created a comparison using "dist plot" and "kde plot” for the numerical features to check their distribution, statistical Mean, Median and Skew values, also an “kde plot” with two different parameters to the feature that used be categorical. Next, get a box plot of all features, and one side by side, also in a “hist” and “box plot”, create an “kde plot” of Age and Disease as hue.

Create an example to deal with outliers but opt not to delete them, once these outliers could be no errors in the data, but just a usual value in the observations with no frequency. Create an example of feature scaling testing Min Max Scaler, Standard Scaler, Normalisation and Robust Scaler.

As a feature engineering step, created after analyses a new data frame where it would be necessary to split the feature "Disease" using a dummy method, next not deal with text data processing, or time series because there is none in the dataset.

In Feature Selection, it used Univariate Feature Selection (UFS) and Recursive Feature Elimination (RFE), also created new hypothesis tests and used Box Plots to visualise and compare the features and the ANOVA test to confirm. After the analyses, following some hypotheses could work, but in a specific approach and others not so much, for example:

1. Using the "Disease" feature from the "df" data frame as a target to predict Diseases could **not** be advised, because, after analysis, there is no significant evidence of a relationship between the features and the target feature in question, in this case, more specific data for each Disease should be collected to better results.
2. Using the "Disease A, B, C and N" features separately from the "disease" data frame as a target to predict specific Diseases could be better interpreted by avoiding noise, once there is significant evidence of a relationship between the features and the target feature in question with this data set.
   1. "Disease A" there is evidence of a significant relationship between the features: Glucose, SleepTime, and other Diseases, (could be advised).
   2. "Disease B" there is evidence between the features: Glucose, and other Diseases, (could be advised).
   3. "Disease C" there is evidence between the features: The other Diseases, (could **not** be advised, needs more data).
   4. "Disease N" there is evidence between the features: The other Diseases, (could **not** be advised, needs more data).
3. Using the "Smoker" feature from the "df" data frame as a target to predict Smokers could be advised, because, after analysis, there is significant evidence of a relationship between the features and the target feature in question, in the features: "Age, Glucose, HeartRate and HealthIns".
4. Using the "PhysicalActivity " feature from the "df" data frame as a target to predict Physical Activities could be advised, because, after analysis, there is significant evidence of a relationship between the features and the target feature in question, in the features: "Age and Glucose".
5. Using the "StressLevel" feature from the "df" data frame as a target to predict Diseases could **not** be advised, because, after analysis, there is no significant evidence of a relationship between the features and the target feature in question, in this case, more specific data for Stress Levels should be collected to better results.

Next in Feature Extraction, it was tested LDA (Linear Discriminant Analysis), PCA (Principal Component Analysis) for comparison between both and t-DSNE (t-Distributed Stochastic Neighbor Embedding) to test, and not get good results using the feature “Disease” as a target that was known already their performance after use ANOVA test, but it opted to try until Machine Learning step.

Since the beginning, it was possible to understand that this data set is considered Supervised, due to the categorical features: "Smoker, PhysicalActivity, StressLevel and Disease" presents.

In this case, the PCA (Principal Component Analysis) unsupervised method is mostly used for Regression and Classification and the LDA (Linear Discriminant Analysis) supervised method for Classification Problems, both could work well in this situation as a Classification Problem, the PCA focus on get the variance/signal in the data removing noise, variance, redundancy and bias while LDA focus in maximise class separability, that can fix better in this classification problem with the existence of labels.

After tests, to create new features, and predict results, LDA got better since the beginning while PCA needed more features to work better, and LDA just with 3 got it, also TSNE showed a good performance and distribution.

# **Machine Learning**

After the first analysis of the data frame, it was possible to understand that this data set is considered Supervised Learning, due to the categorical features: "Smoker, PhysicalActivity, StressLevel and Disease" present, there exist labels, but also might be used as Unsupervised learning approaches to test this data set if desired.

Once it is discovered that the data set is Supervised learning might use some approaches such as:

1. Supervised learning Regression (Continuous)
   1. Regression model: dependent variable and independent variable
   2. Linear regression (Multiple Linear/Polynomial)
   3. Decision Tree
   4. Random Forest
   5. Neural Network
   6. Others
2. Supervised learning Classification (Discrete)
   1. Logistic regression
   2. Support vector machine
   3. Naive Bayes
   4. Decision Tree
   5. Random forest
   6. Neural Network
   7. Others

And if might try the Unsupervised approach might use some approaches such as:

1. Unsupervised learning
   1. Unsupervised learning (Patterns from input data without references to labelled outcomes)
   2. Clustering (K-means, Hierarchical, Mean Shift, Density-Based)
   3. Dimensionality Deduction
   4. PCA
   5. Others

In this case, this Assessment Task will be used just for Supervised learning. Once the pros of the data set are labelled already, then we can pick up a Target to test like: "Smoker, PhysicalActivity, StressLevel and Disease" features, once it does not exist it will be necessary to use the Unsupervised learning approaches or use feature engineering method such as Feature Creation to generate a Target Feature if it is possible, cons might be a waste of time once the data set is labelled already.

However once the data set feature does not have a strong correlation between the features, it might be good to use the Unsupervised learning method, to pick up the strong signal in the data to use.

Next continuous to try to use the First Hypothesis Disease as a Target, tested with the Machine Learning Algorithms Decision Tree, Random Forest, SVM (Support vector machine) and GNB (Gaussian Naive Bayes) with a split of 30% of testing and also with 20% of testing, Accuracy: 0.25, plotted a tree graphic, next used a Grid Search to Find Optimal Hyperparameters do not change the result as much getting an Accuracy: 0.24.

Random Forest: with a split of 30% of testing and also with 20% of testing, Accuracy: 0.25, plotted a tree graphic, next with Random Forest Regressor plotted another tree graphic with just three as max depth, next used a Grid Search to Find Optimal Hyperparameters do not change the result as much getting an Accuracy: 0.24, tried with another approach and got Accuracy: 0.23, tried with Hyperparameters GridSearchCV and KFold and got Accuracy: 0.23.

SVM: with a split of 30% of testing and also with 20% of testing, Accuracy: 0.24, next used a Grid Search to Find Optimal Hyperparameters do not change the result as much getting an Accuracy: 0.25.

GNB: with a split of 30% of testing and also with 20% of testing, Accuracy: 0.23, and not used a Grid Search to Find Optimal Hyperparameters.

Next the Second Hypothesis Disease A, B, C and N as Target, Decision Tree: with a split of 30% of testing and also with 20% of testing, Accuracy: 0.75, next used a Grid Search to Find Optimal Hyperparameters do not change the result as much getting an Accuracy: 0.76, also used LDA to try others Machine Learning Algorithms and got: Logistic Regression 0.74, LDA 0.74, KNeighbors Classifier 0.69, Decision Tree Classifier 0.63, Gaussian NB 0.74, SVC 0.74, Random Forest Classifier 0.63 and MLP Classifier 0.74.

Third Hypothesis Smoker as Target, Decision Tree: with a split of 30% of testing and also with 20% of testing, Accuracy: 0.50, next used a Grid Search to Find Optimal Hyperparameters do not change the result as much getting an Accuracy: 0.51.

Fourth Hypothesis PhysicalActivity as Target, Decision Tree: with a split of 30% of testing and also with 20% of testing, Accuracy: 0.34, next used a Grid Search to Find Optimal Hyperparameters do not change the result as much getting an Accuracy: 0.31.

Fifth Hypothesis StressLevel as Target, Decision Tree: with a split of 30% of testing and also with 20% of testing, Accuracy: 0.34, next used a Grid Search to Find Optimal Hyperparameters do not change the result as much getting an Accuracy: 0.33.

**First Hypothesis Disease as Target**

|  |  |  |  |
| --- | --- | --- | --- |
| **Decision Tree** | **Random Forest** | **SVM (Support Vector Machine)** | **GNB (Gaussian Naive Bayes)** |
| Accuracy: 0.25 | Accuracy: 0.25 | Accuracy: 0.24 | Accuracy: 0.23 |
| Hyperparameters  Accuracy: 0.24 | Hyperparameters  Accuracy: 0.24, Accuracy: 0.23 | Hyperparameters  Accuracy: 0.25 | Hyperparameters  - |

**Other Hypothesis**

|  |  |  |
| --- | --- | --- |
| **Second Hypothesis Disease A, B, C and N as Target** | Decision Tree  Accuracy: 0.75 | Decision Tree  Hyperparameters  Accuracy: 0.76 |
| **Third Hypothesis Smoker as Target** | Decision Tree  Accuracy: 0.50 | Decision Tree  Hyperparameters  Accuracy: 0.51 |
| **Fourth Hypothesis PhysicalActivity as Target** | Decision Tree  Accuracy: 0.34 | Decision Tree  Hyperparameters  Accuracy: 0.31 |
| **Fifth Hypothesis StressLevel as Target** | Decision Tree  Accuracy: 0.34 | Decision Tree  Hyperparameters  Accuracy: 0.33 |

In the First Hypothesis Disease as Target, all algorithm tests have the approximate Accuracy as a result, reinforcing once again that more information for each different Disease, for the Second Hypothesis Disease A, B, C and N as Target, the Decision Tree and the others tested through LDA showed interesting to be used and improve more, same to the Third Hypothesis Smoker as Target, could be improved, and the Fourth Hypothesis PhysicalActivity as Target and Fifth Hypothesis StressLevel as Target, showed similarity in results.

For the hypothesis that got results as not desired (lower results) might be considered to collect more data. The collection of data across multiple departments is probably causing noise to the data itself and not allowing it to work as desired, or the data set is missing specific features to better analyse the other hypothesis.

# **References**

Albon, C. (2018). Machine learning with Python cookbook practical solutions from preprocessing to deep learning. Cambridge O’reilly.

Cereniyim (2019). Distribution plot with mean and median. [online] GitHub Gist. Available at: https://gist.github.com/cereniyim/df1c8646411cc34eb9720a79400a200f [Accessed 29 Apr. 2024].

DataCamp (2023). Python t-SNE with Matplotlib. [online] www.datacamp.com. Available at: https://www.datacamp.com/tutorial/introduction-t-sne [Accessed 6 May 2024].

Domino Data Lab (2024). What is Feature Engineering? | Domino Data Science Dictionary. [online] domino.ai. Available at: https://domino.ai/data-science-dictionary/feature-engineering [Accessed 27 Apr. 2024].

GeeksforGeeks (2018). Principal Component Analysis(PCA). [online] GeeksforGeeks. Available at: https://www.geeksforgeeks.org/principal-component-analysis-pca/ [Accessed 5 May 2024].

Hai Huang (2024). What are the best practices for feature selection in machine learning? [online] www.linkedin.com. Available at: https://www.linkedin.com/advice/3/what-best-practices-feature-selection-machine-u9crc [Accessed 4 May 2024].

Hashmi, F. (2021). Advertisement click classification case study in Python - Thinking Neuron. [online] Thinking Neuron. Available at: https://thinkingneuron.com/advertisement-click-classification-case-study-in-python/ [Accessed 24 May 2024].

Mahadevan, M. (2022). Step-by-Step Exploratory Data Analysis (EDA) using Python -. [online] Analytics Vidhya. Available at: https://www.analyticsvidhya.com/blog/2022/07/step-by-step-exploratory-data-analysis-eda-using-python/ [Accessed 27 Apr. 2024].

Marangon, J.D. (2023). 🏋🏼‍♂️ How to use cross-validation to train models in scikit-learn. [online] Medium. Available at: https://medium.com/@johnidouglasmarangon/%EF%B8%8F-how-to-use-cross-validation-to-train-models-in-scikit-learn-f94518de8e2d [Accessed 6 May 2024].

Michael Waskom (2024). seaborn.pairplot — seaborn 0.10.1 documentation. [online] seaborn.pydata.org. Available at: https://seaborn.pydata.org/generated/seaborn.pairplot.html [Accessed 28 Apr. 2024].

Płoński, P. (2020). How to visualize a single Decision Tree from the Random Forest in Scikit-Learn (Python)? [online] MLJAR. Available at: https://mljar.com/blog/visualize-tree-from-random-forest/ [Accessed 6 May 2024].

Saturn Cloud (2023). How to Replace None with NaN in Pandas DataFrame | Saturn Cloud Blog. [online] saturncloud.io. Available at: https://saturncloud.io/blog/how-to-replace-none-with-nan-in-pandas-dataframe/ [Accessed 26 Apr. 2024].

scikit-learn developers (2024a). Recursive feature elimination with cross-validation. [online] scikit-learn. Available at: https://scikit-learn.org/stable/auto\_examples/feature\_selection/plot\_rfe\_with\_cross\_validation.html [Accessed 4 May 2024].

scikit-learn developers (2024b). sklearn.preprocessing.RobustScaler — scikit-learn 0.24.2 documentation. [online] scikit-learn.org. Available at: https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.RobustScaler.html [Accessed 29 Apr. 2024].

scikit-learn developers (2024c). Univariate Feature Selection. [online] scikit-learn. Available at: https://scikit-learn.org/stable/auto\_examples/feature\_selection/plot\_feature\_selection.html [Accessed 4 May 2024].

Seaborn (2012). seaborn.kdeplot — seaborn 0.9.0 documentation. [online] Pydata.org. Available at: https://seaborn.pydata.org/generated/seaborn.kdeplot.html [Accessed 29 Apr. 2024].

Shafi, A. (2023). Sklearn Random Forest Classifiers in Python Tutorial. [online] www.datacamp.com. Available at: https://www.datacamp.com/tutorial/random-forests-classifier-python [Accessed 6 May 2024].

The Matplotlib Development Team (2024a). 3D scatterplot — Matplotlib 3.7.1 documentation. [online] matplotlib.org. Available at: https://matplotlib.org/stable/gallery/mplot3d/scatter3d.html.

The Matplotlib Development Team (2024b). Choosing Colormaps in Matplotlib — Matplotlib 3.8.0 documentation. [online] matplotlib.org. Available at: https://matplotlib.org/stable/users/explain/colors/colormaps.html [Accessed 1 May 2024].

The Matplotlib Development Team (2024c). matplotlib.pyplot.axvline — Matplotlib 3.8.4 documentation. [online] matplotlib.org. Available at: https://matplotlib.org/stable/api/\_as\_gen/matplotlib.pyplot.axvline.html#matplotlib.pyplot.axvline [Accessed 29 Apr. 2024].

The Pandas Development Team (2024a). pandas.DataFrame.reset\_index — pandas 1.4.2 documentation. [online] pandas.pydata.org. Available at: https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.reset\_index.html [Accessed 6 May 2024].

The Pandas Development Team (2024b). pandas.DataFrame.T — pandas 2.1.3 documentation. [online] pandas.pydata.org. Available at: https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.T.html [Accessed 29 Apr. 2024].

The scikit-yb developers (2019). Recursive Feature Elimination — Yellowbrick v1.1 documentation. [online] www.scikit-yb.org. Available at: https://www.scikit-yb.org/en/latest/api/model\_selection/rfecv.html [Accessed 4 May 2024].

Yashmeet Singh (2022). Boxplot With Separate Y-Axis for Each Column | Proclus Academy. [online] proclusacademy.com. Available at: https://proclusacademy.com/blog/quicktip/boxplot-separate-yaxis/ [Accessed 29 Apr. 2024].