**CCT College Dublin**

**Assessment Cover Page**

|  |  |
| --- | --- |
| **Module Title:** | Data Preparation Machine Learning |
| **Assessment Title:** | CA2\_DP\_ML\_HDip\_Lvl8 |
| **Lecturer Name:** | Muhammad Iqbal, David McQuaid |
| **Student Full Name:** | Natalia Gomes |
| **Student Number:** | 2020350 |
| **Assessment Due Date:** | 21st November 2023 |
| **Date of Submission:** | 2nd Jan 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. |

Data Analytics: Dynamics of Data Preparation and Machine Learning

Introduction

When it comes to data-driven decision-making, the foundation of understanding and preparing datasets is essential to drawing insightful conclusions and building effective machine learning models. We begin by thoroughly analyzing the properties of the dataset, navigating through dimensions, missing values, and outliers. We move beyond basic data cleaning to considering the reasoning behind every choice. Histograms and scatter plots are two examples of visualizations that show the path to insightful conclusions.

Encoding, scaling, and feature engineering are strategic maneuvers aimed at fortifying datasets for ensuing modeling challenges. Dimensionality reduction using Linear Discriminant Analysis (LDA) and Principal Component Analysis (PCA) serves as a pinnacle, showcasing their impacts on classification and clustering tasks.

The transition into the realm of machine learning involves strategic choices in selecting the learning approach—supervised or unsupervised—based on the dataset's intrinsic nature. Feature selection, hyper-parameter tuning, training, testing, and final comparison and visualization stages form the conclusive chapters, portraying a holistic picture of modeling outcomes.

This report delves into a thorough exploration of the data preparation and machine learning processes, encapsulating pivotal steps where data transforms into insights and models stand as sentinels of informed decision-making in the realms of data science and artificial intelligence.

*Characterization of the Dataset*

The dataset under consideration, derived from the NHANES\_age\_prediction.csv file, plays a foundational role in our analytical journey. Comprising 2278 entries and 10 columns, this dataset serves as the canvas upon which we paint our insights. The initial exploration reveals the absence of missing values, affirming the integrity of our dataset. Its structure, characterized by columns such as SEQN, age\_group, RIDAGEYR, and others, forms the backbone of our analysis.

Upon closer inspection, we gain valuable insights into the dataset's unique characteristics. The age distribution, encapsulated in the 'RIDAGEYR' column, spans a range from 12 to 80 years, with a mean age of approximately 41.8 years. Further, the gender distribution, indicated by 'RIAGENDR,' presents a categorical classification of values Adult and Senior.

Detailed statistics for numerical attributes, such as 'BMXBMI' (Body Mass Index), 'LBXGLU' (Glucose levels), and 'LBXGLT' (Total Cholesterol), provide a comprehensive understanding of the dataset's composition. Notably, the 'DIQ010' column, representing Diabetes Status, offers a glimpse into the prevalence of diabetes within the dataset.

This comprehensive dataset classification and analysis set the stage for the rest of our analysis, directing our choices for feature engineering, scaling, encoding, and data cleaning. The details of the dataset become the compass that guides our analytical efforts as we set out on this data analytics exploration (Chumbe, 2022).

*Data Cleaning and Exploratory Data Analysis*

In the critical phase of Data Cleaning and Exploratory Data Analysis (EDA), our primary objective is to ensure the integrity and reliability of our dataset. This process begins with handling of missing values, achieved by the removal of incomplete entries, thus enhancing the overall data quality. At the same time, we address the issue of duplicates, employing a systematic approach to eliminate redundant data points that might distort the analytical process.

A pivotal aspect of our data preparation involves the handling of outliers, a potential source of skewed insights. Leveraging statistical measures such as quartiles, we thoroughly filter out data points lying beyond acceptable thresholds. This ensures that our dataset is robust, representative, and resilient to the influence of extreme values.

In tandem with data cleaning, we prioritize clarity in our dataset by renaming variables, fostering a more intuitive and accessible understanding of the information at hand. This optimization contributes to the overall interpretability and user-friendliness of our data.

Transitioning into the area of Exploratory Data Analysis, we harness the power of visualization to unravel complex patterns within the dataset. Our collection includes scatter plots, histograms, and box plots, each offering a unique perspective on the relationships and distributions within the data. The scatter plot of Age against BMI is a versatile exploratory tool that provides a visual narrative of the relationship between these two variables. By visualizing the distribution of these data points, the plot provides an at-a-glance overview of how BMI is distributed across different age groups. This insight is valuable for understanding the overall health profile of the population under study.

A key component of our exploratory journey is the Boxplot of Age Distribution by Diabetes Status. This visual representation allows us to discern patterns and variations in age concerning different diabetes statuses, aiding in identifying potential trends or anomalies.

Concluding our exploration, we present a pie chart illustrating the distribution of age groups within our dataset. This concise visual encapsulates the prevalence of Adult and Senior categories, offering a snapshot of the demographic composition. In this specific instance, the Adult category dominates with an 84.0% share, while the Senior category accounts for the remaining 16.0%.

Throughout this data preparation and exploratory analysis, transparency remains paramount. At each step, we provide a rationale for our decisions, ensuring a thorough understanding of the methodologies applied and insights derived (Stedman, Burns and Pratt, 2022).

*Encoding, Scaling, and Feature Engineering*

In the process of data preprocessing, the presented code snippet focuses on three key aspects: encoding categorical variables, scaling numerical features, and performing feature engineering. Firstly, the 'age\_group' column, containing categorical data representing different age groups, is encoded using the LabelEncoder from scikit-learn. The resulting 'encoded\_column' provides a numerical representation of the 'age\_group' for compatibility with machine learning algorithms.

Moving on to numerical features, the 'RIDAGEYR' column, representing the age of individuals, is standardized using StandardScaler. The standardized values are stored in the 'scaled\_column,' ensuring that numerical variables are on a comparable scale, preventing any dominance by variables with larger magnitudes.

Feature engineering is demonstrated by creating a new feature, 'new\_feature,' which is a product of the 'RIDAGEYR' and 'BMXBMI' columns. This new feature is designed to capture potential interactions between age and body mass index, providing the model with additional information that might enhance predictive performance.

The decisions made in this process align with best practices in data preprocessing. Encoding categorical variables ensures compatibility with machine learning models, scaling numerical features maintains consistency in the impact of each variable, and feature engineering introduces potential predictors that may contribute valuable insights (Bhandari, 2023). The resulting data frame reflects these transformations, with the encoded, scaled, and newly engineered columns integrated seamlessly into the dataset. This approach lays the foundation for subsequent stages in the machine learning, enhancing the model's ability to derive meaningful patterns and relationships from the data.

*Dimensionality Reduction*

In this section, the code employs two dimensionality reduction techniques, Linear Discriminant Analysis (LDA) and Principal Component Analysis (PCA), on a given dataset. The dataset is loaded, comprising features denoted as X and corresponding target variables denoted as y. To facilitate the analysis, string labels in the target variable are encoded into numerical values using the LabelEncoder.

Following the encoding process, the features are standardized using StandardScaler to ensure consistent scales for accurate comparison. Subsequently, the code applies LDA and PCA. The LDA results are visualized in a scatterplot, where 'LDA Component 1' is plotted on the x-axis, and the 'Target Variable' is on the y-axis. The scatterplot displays a linear separation of classes, with yellow dots concentrated at 1.0 on the y-axis and dark-purple dots at 0.0.

For PCA, a two-dimensional scatterplot is generated with 'PCA Component 1' on the x-axis and 'PCA Component 2' on the y-axis. The plot illustrates a dense cluster of yellow and dark-purple dots in the bottom left, while sparse dark-purple dots appear in the top right. This distribution indicates a distinct separation of classes, especially in the lower-dimensional space defined by the principal components.

The differences between LDA and PCA lie in their underlying mathematical principles. LDA aims to maximize class separation, making it particularly effective for classification tasks. On the other hand, PCA focuses on capturing overall variance and is commonly used for data visualization and noise reduction.

The outcomes of Linear Discriminant Analysis (LDA) and Principal Component Analysis (PCA) can significantly influence classification and clustering tasks. In the context of classification, LDA's explicit focus on maximizing class separation is reflected in the linear patterns observed in the scatterplot, suggesting enhanced discriminatory power. Utilizing LDA-transformed features in a classification model is likely to improve accuracy and robustness.

On the other hand, PCA, although not designed for class discrimination, captures essential features, as indicated by dense clusters in its scatterplot. PCA-transformed features can contribute to improved generalization and model performance. In clustering tasks, LDA's linear separation might be less conducive to identifying natural groupings within the data. Conversely, PCA's ability to capture underlying patterns, evident in distinct clusters, makes it a valuable choice for clustering purposes. The selection between LDA and PCA depends on the specific objectives of the classification or clustering task, with LDA excelling in class discrimination, and PCA offering broader insights and potential for dimensionality reduction (Anello, 2022).

Machine Learning

*Choice of Machine Learning Approach*

In this section, the initial steps involve loading a dataset using the `ucimlrepo` library, converting it into a data frame, and exploring its characteristics. The dataset is then divided into features (X) and targets (y), which sets the stage for the subsequent decision-making process.

To determine the nature of the machine learning task, the number of classes in the target variable is assessed. If there are fewer than or equal to two classes, the task is classified as "Classification"; otherwise, it is deemed "Multiclass Classification." This distinction lays the foundation for the subsequent choice of machine learning approach.

The dataset is further split into training and testing sets using the `train\_test\_split` function, facilitating model evaluation. The choice between supervised and unsupervised learning is contingent on the task type determined earlier. If the problem is a classification task, the script advocates for supervised learning, particularly highlighting algorithms such as Logistic Regression, Random Forest, and Support Vector Machines as suitable choices. On the other hand, for regression tasks, the recommendation remains supervised learning, with Linear Regression, Random Forest Regression, and Gradient Boosting being endorsed.

The output of the code indicates a clear decision – the chosen machine learning approach is "Supervised Learning (Classification)," aligning with the identified task type. The selection of classification algorithms, namely Logistic Regression, Random Forest, and Support Vector Machines, follows logically from the classification task at hand. This systematic approach ensures a thoughtful and justified strategy for initiating the machine learning journey tailored to the dataset's characteristics.

*Feature Selection and Hyper-parameter Tuning*

In the process of Feature Selection and Hyper-parameter Tuning, our approach involves a two-fold strategy aimed at optimizing model performance. Initially, we employ a Random Forest classifier to assess the importance of each feature within the dataset. The resulting feature importance are then meticulously scrutinized and ranked, offering valuable insights into the significance of individual attributes. In the presented output, 'LBXGLT' emerges as the most influential feature, followed closely by 'LBXIN' and 'BMXBMI,' collectively contributing to the model's predictive power.

The focus shifts to the selection of a subset of the most impactful features. In this case, we choose the top five features based on their importance scores. This strategic feature selection ensures a streamlined model that leverages the most relevant information for accurate predictions.

The second phase of our strategy involves Hyper-parameter Tuning, where we utilize GridSearchCV to explore various combinations of hyper-parameters and identify the optimal configuration for the Random Forest classifier. The hyper-parameters under consideration encompass essential aspects such as the number of estimators, maximum depth, minimum samples split, and minimum samples leaf. The output unveils the best combination of hyper-parameters, providing a blueprint for an optimized Random Forest model.

Upon implementing the identified optimal hyper-parameter values, the model is evaluated on an independent test set. The accuracy achieved, prominently displayed as "Best Model Accuracy on Test Set: 0.8377," signifies the success of our meticulous tuning process. This result encapsulates the culmination of Feature Selection and Hyper-parameter Tuning, affirming the effectiveness of our approach in enhancing the model's accuracy and predictive capabilities. The reported accuracy serves as a tangible metric of the model's performance, validating the significance of thoughtful feature selection and hyper-parameter optimization in the realm of machine learning (Vegibit, 2022).

*Training and Testing*

In the approach of training and testing the machine learning models, our approach is two-fold, combining supervised and unsupervised learning paradigms. In supervised learning, we split the dataset into training and testing sets, allocating 80% for training and 20% for testing, utilizing the `train\_test\_split` function from the `sklearn.model\_selection` module. Employing a Random Forest Classifier with 100 estimators and a maximum depth of 10, we achieve a commendable accuracy of 84.43% on the test set. This result is obtained through the `accuracy\_score` metric, demonstrating the model's proficiency in making accurate predictions.

Our exploration into unsupervised learning involves K-Means clustering, a popular algorithm for partitioning data into distinct groups. With a predetermined number of clusters set at 2, we utilize the `KMeans` class from the `sklearn.cluster` module, presenting a silhouette score of 0.6066. This metric gauges the quality of clustering, indicating a substantial separation between the identified clusters.

To further enhance the robustness of our models, we examine the K-fold cross-validation, a pivotal step in evaluating their generalization capabilities. For supervised learning, the Random Forest Classifier undergoes cross-validation with 5 folds, resulting in a mean accuracy score of 84.37%. This metric provides a more comprehensive understanding of the model's performance, considering variations in the training and testing splits.

In the area of unsupervised learning cross-validation, we employ K-Means clustering once again. However, this time, we opt for the `cross\_val\_predict` function and a K-Fold strategy. The silhouette score derived from this process, at 0.2192, adds an additional layer of validation to our unsupervised learning endeavors.

According to Bhardwaj (2022), this extensive evaluation not only highlights the effectiveness of our supervised learning model in predicting outcomes but also emphasizes the reliability of our unsupervised learning approach in discerning inherent patterns within the data. Through this comprehensive analysis, we pave the way for an apparent understanding of our models' performance, crucial for their deployment in real-world scenarios.

*Comparison and Visualization*

In this part, the focus is on comparing and visually representing the outcomes of machine learning models. The process begins by splitting the dataset into training and testing sets, a fundamental step in model evaluation. Two main approaches are explored: supervised learning using a Random Forest Classifier and unsupervised learning through K-Means clustering. The Random Forest Classifier is trained on the training set, and predictions are made on the test set, leading to a supervised accuracy score.

K-Means clustering is applied as an unsupervised learning technique. The silhouette score, a metric that evaluates the quality of clustering, is utilized to assess the effectiveness of the unsupervised model. The code further incorporates K-Fold Cross-Validation to enhance model evaluation, ensuring robustness by splitting the dataset into multiple folds. Both supervised and unsupervised learning methods undergo cross-validation, providing a mean accuracy score and silhouette score, respectively.

The output of the code is then presented in a tabular format for a concise comparison of the two models. The table includes columns for 'Model/Method,' 'Accuracy/Silhouette,' and 'Cross-Validation Mean.' 'Supervised Learning' and 'Unsupervised Learning (K-Means)' are compared in terms of accuracy and silhouette scores, providing a clear overview of their respective performances.

To enhance the interpretability of the results, a bar plot visualization is generated. The y-axis represents the 'Accuracy/Silhouette Score' ranging from 0.0 to 0.8, while the x-axis displays the 'Model/Method.' The bar plot employs distinct colors for better differentiation, with blue indicating 'Supervised Learning' and green representing 'Unsupervised Learning (K-Means).' The visual representation serves as an intuitive summary, allowing stakeholders to quickly grasp the comparative performance of the two models. According to Bhardwaj (2021), his approach enables a comprehensive understanding of the statistical outcomes, promoting informed decision-making in the realm of machine learning.

Conclusion

This essay explores into data analytics complexities, highlighting machine learning and data preparation dynamics. A complex dataset review laid the foundation, ensuring integrity and paving the way for subsequent analysis. Data cleaning and EDA handled outliers, duplicates, and missing values, justified transparently. Visualization methods revealed nuanced data trends.

Encoding, scaling, and feature engineering prepared the dataset for machine learning, with LDA and PCA serving vital roles in dimensionality reduction. Transitioning to machine learning involved judicious choices, emphasizing feature selection and hyper-parameter tuning for a finely-tuned Random Forest model.

Training and testing showcased the efficacy of Random Forest in prediction accuracy and K-Means clustering in pattern identification. K-Fold cross-validation added robustness. Comparative performance visualization through tabular and bar plot formats facilitated intuitive understanding, guiding informed decision-making in machine learning.

This thorough examination of data analytics, which includes everything from dataset categorization to machine learning results, essentially emphasizes the need of careful data preparation and developing models. A better understanding of the dataset is made possible by the clear reasoning behind each step, which guarantees replicability and leads to informed decision-making in the fields of data science and machine learning.

References

Anello, E. (2022) *What is the difference between PCA and LDA?* https://365datascience.com/tutorials/python-tutorials/lda-vs-pca/.

Bhandari, A. (2023) *Feature Engineering: Scaling, Normalization, and Standardization (Updated 2023)*. https://www.analyticsvidhya.com/blog/2020/04/feature-scaling-machine-learning-normalization-standardization/.

Bhardwaj, A. (2021) 'Silhouette Coefficient - towards data science,' *Medium*, 14 December. https://towardsdatascience.com/silhouette-coefficient-validating-clustering-techniques-e976bb81d10c.

Bhardwaj, A. (2022) 'Training and Test data | Splitting of data  | Medium,' *Medium*, 14 July. https://iaviral.medium.com/training-and-test-data-27b7e166e5d5.

Chumbe, K. (2022) 'A comprehensive guide to a classification project - towards Data science,' *Medium*, 6 January. https://towardsdatascience.com/a-comprehensive-guide-to-a-classification-project-data-cleaning-and-exploration-88edd5617ce2.

Stedman, C., Burns, E. and Pratt, M.K. (2022) *What is data preparation? An in-depth guide to data prep*. https://www.techtarget.com/searchbusinessanalytics/definition/data-preparation.

Vegibit (2022) *SciKit-Learn: Hyperparameter tuning and feature selection*. https://vegibit.com/scikit-learn-hyperparameter-tuning-and-feature-selection/.