NATIONAL COLLEGE OF IRELAND

DATA MINING AND MACHINE LEARNING I

Student Name: SAIHARSHA GURIJALA

Student ID: 23194341

I. Abstract

This project investigates the results of four implementations of two core machine learning machine techniques – Random Forest and Decision Trees – using corresponding Scikit-Learn, XGBoost, and LightGBM libraries on three different datasets: Bank Marketing, Online Shoppers Purchasing Intention, and Estimation of Obesity levels. The purpose of the study is to conduct a detailed background analysis to identify whether multiple implementations of the same algorithm feature similar results and clarify which dataset characteristics determine model performance. Besides, background analysis includes the exploration of original datasets, and the use of advanced data preprocessing techniques such as Principal Component Analysis makes it possible to reveal and present the hidden structure in such a way that is easier, quicker, and more efficient. Its primary focus is ensuring the testability and traceability of the findings. This terminal project critically evaluates and compares machine-learning models based on a systematic framework which promotes the identification of the best suitable methodology for practical application in data mining and shows how this terminal assessment helps inform the best performing models based on precision, recall, F1-score, and accuracy measures as well as their adaptiveness to various data dimensions, making a significant contribution to the field of data mining and knowledge discovery.

II. Introduction

Modern industries are characterized by a dynamic and data-driven environment that creates both challenges and opportunities for machine learning applications to analyse and predict a wide range of outcomes that impact operational efficiency and strategic management. Random Forests and Decision Trees are some of the most commonly used models in this field, and, in this paper, we apply them to three datasets to test and compare their potential. The three datasets from the UCI Machine Learning Repository are: Bank Marketing, Online Shoppers Purchasing Intention, and Estimation of Obesity Levels. They offer an overview of different industries and the challenges it produces.

The Bank Marketing dataset provides insights into the effectiveness of marketing strategies in the banking sector, focusing on customer behaviours towards term deposit subscriptions. The Online Shoppers Purchasing Intention dataset allows us to delve into e-commerce dynamics, examining factors that influence purchase decisions during browsing sessions. Lastly, the Estimation of Obesity Levels dataset offers a public health perspective, highlighting the correlation between lifestyle choices and obesity classifications. By applying advanced data mining techniques, this project aims to uncover the strengths and limitations of each model in handling different types of data, contributing significantly to the broader field of knowledge discovery and data mining.

Finally, the comparative analysis of the two predictors tests the Random Forest's and Decision Trees adaptability and accuracy while simultaneously offering valuable insights that could be utilized to better inform marketing strategies, e-commerce activities, and public health intervention policies. Through the identification of the most significant predictive variables and the evaluation of the model effectiveness in different settings, the current research enriches our knowledge of machine learning's empirical potential and establishes a foundation for further predictive analytics studies in the vital areas.

III. Methodology

1. Data Collection

This research procured 3 different datasets from the UCI Machine Learning Repository(https://archive.ics.uci.edu/datasets) which encompass various operational contexts and problems. The reason for the selections is to demonstrate the versatility and practicality of machine learning in a range of contexts and sectors. The datasets included are:

a) Bank Marketing Dataset:

The data set is taken from the direct marketing activities of a Portuguese bank. These datasets contain information related to the banking client's details, his socio-economic situation, and the consequences of applying the particular marketing strategy. The task is to predict the subscription of the client with the term deposit. This is the case of binary classification of the event. Application of this dataset is crucial to analyse the consequences of marketing and to learn more about client needs in the banking sector. Thus, this dataset can work out suggestions to make the customer more involved with possibilities conversion.

b) Online Shoppers Purchasing Intention Dataset:

This dataset is sourced from a user's interactions with an e-commerce platform, offering session-level details that encompass page visit, session duration and bounce rate, among other final purchase activity. The competition objective is to forecast purchase probability, a task described as binary classification. Knowledge from this dataset may be used to recognize vital consumer behaviours online and develop better online business via changes in web design or customer interface.

c) Estimation of Obesity Levels Dataset:

The dataset is a direct survey that also includes health records. Apart from lifestyle factors, the data on technology usage and transportation preferences has also been included. It is a multilabel dataset where individuals are classified into different obesity categories. Therefore, it seems useful for the public health dataset. The dataset allows for the examination of relationships between lifestyle factors and obesity, which is essential in determining the targeted lifestyle interventions and policies.

Data Utilization in the Project: The variety of these datasets ensures a strong baseline for using data mining for real-life problems in various fields, whether it is marketing or e-commerce, public health, etc. Thus, this set of datasets allows for a holistic review of the selected machine learning models, supporting a priori methodological considerations. The latter are closely aligned with the project's objectives to analyse and prove the performance of different algorithmic setups, as well as to get a holistic picture of how the distribution of specific data aspects can impact the performance of a model. The choice of these datasets is methodologically critical for this study as it is compared to the empiricism of the study design and its promotion through the exploration of advanced analytics implementation flexibility and performance in different data-driven contexts. **[1]**

2. Dataset Description

a) Bank Marketing Dataset: This dataset has 45,211 records and 17 columns and includes multiple features representing client information and how they interacted with the bank's marketing strategy.

Summary of the dataset:

The dataset contains several attributes related to the clients of a bank involved in direct marketing campaigns. The 'Age' attribute is a numeric variable that captures the age of the clients, which ranges from 18 to 95 years. The 'Job' attribute is a categorical variable that describes the client's occupation, including various types such as 'admin.', 'blue-collar', 'entrepreneur', 'housemaid', 'management', 'retired', 'self-employed', 'services', 'student', 'technician', 'unemployed', and 'unknown'. The marital status of the clients is also categorized into three types: 'married', 'divorced', and 'single', with 'divorced' encompassing widowed individuals as well. The 'Education' attribute classifies the clients' educational background into four categories: 'unknown', 'primary', 'secondary', and 'tertiary'.

The dataset also includes a 'Default' attribute, which is a binary variable indicating whether the client has any credit default ('yes', 'no'). Financial data is represented through the 'Balance' attribute, a numeric variable showing the average yearly balance in euros, which interestingly includes negative values, hinting at possibly overdrawn accounts.

Regarding loans, the dataset records whether the client has a housing loan ('yes', 'no') and if they have a personal loan ('yes', 'no'), both of which are binary attributes. Campaign-related data is extensively covered as well. The 'Contact Type' attribute is a categorical variable that indicates the type of communication used during the last contact ('unknown', 'telephone', 'cellular'). The 'Day of Last Contact' is a numeric variable that notes the day of the month when the last contact occurred. Similarly, the 'Month of Last Contact' categorizes the month of the last interaction, with entries like 'jan', 'feb', 'mar', etc.

The 'Duration of Last Contact' measures the length of the last call in seconds, while the 'Number of Contacts' is a numeric variable indicating the number of contacts made during the current campaign for a specific client. 'Days Since Last Contact' indicates the number of days that have passed since the client was last contacted from a previous campaign, with a value of -1 denoting no prior contact. 'Previous Contacts' notes the number of contacts made before the current campaign for a particular client.

Additionally, the 'Outcome of Previous Campaign' is a categorical variable describing the result of the previous marketing campaign ('unknown', 'other', 'failure', 'success'). Lastly, the target variable, 'Term Deposit Subscription' (y), is a binary variable that indicates whether the client subscribed to a term deposit ('yes', 'no'). This comprehensive dataset allows for a nuanced analysis of client behaviour and campaign effectiveness.

```
bank_marketing_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 45211 entries, 0 to 45210
Data columns (total 16 columns):
 #   Column       Non-Null Count  Dtype
---  ------       --------------  -----
 0   age          45211 non-null  int64
 1   job          44923 non-null  object
 2   marital      45211 non-null  object
 3   education    43354 non-null  object
 4   default      45211 non-null  object
 5   balance      45211 non-null  int64
 6   housing      45211 non-null  object
 7   loan         45211 non-null  object
 8   contact      32191 non-null  object
 9   day_of_week  45211 non-null  int64
 10  month        45211 non-null  object
 11  duration     45211 non-null  int64
 12  campaign     45211 non-null  int64
 13  pdays        45211 non-null  int64
 14  previous     45211 non-null  int64
 15  poutcome     8252 non-null   object
dtypes: int64(7), object(9)
memory usage: 5.5+ MB
```

Fig 1: Dataset Info

```
bank_marketing_df.isna().sum()
```

```
age              0
job            288
marital          0
education     1857
default          0
balance          0
housing          0
loan             0
contact      13020
day_of_week      0
month            0
duration         0
campaign         0
pdays            0
previous         0
poutcome     36959
dtype: int64
```

Fig 2: Null values Count of Dataset

Before delving into various data processing and modelling steps, it is imperative to recognize the existence of multiple missing data in several crucial columns in the Bank Marketing dataset analysis. The dataset has null values in several attributes, and these could compromise the final predictions from the model if not critically adjusted during the data cleaning step.

The 'job' column has 288 missing entries, there are 1,857 missing entries in the 'education' column, The 'contact' column has 13,020 missing entries, and the highest number of missing values is in the 'poutcome' column, with 36,959 missing entries.

As these null values would negatively affect the accuracy of the model, they should be taken into account during the cleaning and preprocessing phases of the data. Indeed, it is crucial for any analysis based on the dataset to be accurate and reliable; thus, it is important to process these values properly. Therefore, during the preprocessing of the dataset in question, it would be necessary to address these gaps to make the dataset for the machine learning model building as whole, complete, and realistic as possible.

While the given data source provides a much multidimensional view on the clients of the bank on its current marketing that will be crucial to hypothesise the potential improvements that might be later revised into the bank's marketing, each of the variables is going to facilitate the more thorough investigation into clients' behaviour as well as the implications of the financial product and implemented marketing activities.

b) Online Shoppers Purchasing Intention Dataset: The Online Shoppers Purchasing Intention dataset consists of precise logs to capture user activity on a given website and establish patterns that correlate to the probability that the user will commit a purchase. The data spans 12,330 non-distinct sessions with unique data points that describe specific areas of the user-Site interface. There are 18 columns that measure distinct activities in two perspectives: user action and the technical details related to that action:

Columns Overview:

Administrative, Administrative_Duration: These attributes capture the number of administrative pages visited and the total time spent on these pages during the session.

Informational, Informational_Duration: These features represent the number of informational pages visited and the total duration spent on such pages, respectively.

ProductRelated, ProductRelated_Duration: These fields reflect the number of product-related pages visited and the cumulative duration spent on these pages.

BounceRates, ExitRates, PageValues: These continuous variables provide metrics on the engagement quality; BounceRates shows the percentage of visitors who enter the site and leave immediately, ExitRates represents the proportion at which

users exit the website after visiting various pages, and PageValues indicates the average value of a page that a user visited before completing an e-commerce transaction.

SpecialDay: This indicates the proximity of the site visit time to a special day (e.g., Mother's Day, Valentine's Day) which might influence purchase decisions.

Month, OperatingSystems, Browser, Region, TrafficType: These categorical variables provide context regarding the time of the visit, the technical environment of the user, and the geographical and digital pathways that led them to the website.

VisitorType: It categorizes the visitors as Returning or New to understand different behaviours between these groups.

Weekend: A binary variable indicating whether the session occurred on a weekend.

Target Variable:

Revenue: The primary target variable, a Boolean attribute indicating whether the session resulted in a revenue-generating transaction (True or False).

```
online_shoppers_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 12330 entries, 0 to 12329
Data columns (total 17 columns):
 #   Column                   Non-Null Count  Dtype
---  ------                   --------------  -----
 0   Administrative           12330 non-null  int64
 1   Administrative_Duration  12330 non-null  float64
 2   Informational            12330 non-null  int64
 3   Informational_Duration   12330 non-null  float64
 4   ProductRelated           12330 non-null  int64
 5   ProductRelated_Duration  12330 non-null  float64
 6   BounceRates              12330 non-null  float64
 7   ExitRates                12330 non-null  float64
 8   PageValues               12330 non-null  float64
 9   SpecialDay               12330 non-null  float64
 10  Month                    12330 non-null  object
 11  OperatingSystems         12330 non-null  int64
 12  Browser                  12330 non-null  int64
 13  Region                   12330 non-null  int64
 14  TrafficType              12330 non-null  int64
 15  VisitorType              12330 non-null  object
 16  Weekend                  12330 non-null  bool
dtypes: bool(1), float64(7), int64(7), object(2)
memory usage: 1.5+ MB
```

Fig 3: Dataset Info

```
online_shoppers_df.isna().sum()
```

```
Administrative             0
Administrative_Duration    0
Informational              0
Informational_Duration     0
ProductRelated             0
ProductRelated_Duration    0
BounceRates                0
ExitRates                  0
PageValues                 0
SpecialDay                 0
Month                      0
OperatingSystems           0
Browser                    0
Region                     0
TrafficType                0
VisitorType                0
Weekend                    0
dtype: int64
```

Fig 4: Null values Count of Dataset

The first point to note is the completeness of the dataset; i.e., there are no missing values for any of the variables. This can be inferred by directly querying the dataset, which would return 0 for each of the fields. That implies the data collection process was thorough and, in this way, the dataset stands ready for analysis without first having to clean it of entries where take was not possible.

The richness of the data available on the interactions with users and the lack of missing values consolidate the high utility of this dataset for building predictive algorithms identifying patterns associated with completing a purchase. The numerous variables can be used to provide a nuanced analysis of user behaviour, including simple characteristics such as demographics and session information as well as elaborate indicators of engagement, readiness to purchase, and other aspects. The cleanliness of the dataset and its lack of redundancy is likely to result in easy integration into machine learning processes, which will ultimately support the development of accurate and informative predictive studies aimed to improve e-commerce planning. **[2]**

c) Estimation of Obesity Levels Dataset: The dataset Estimation of Obesity Levels Dataset which is a well-structured data set that was created to explore the different causes that affect the levels of obesity individuals go through which includes several demographic and lifestyle factors. It has 2,111 records, which are an essential tool for studying how each behaviour or feature manifests on obesity status. Additionally, it comes with 17 attributes, each reflecting personal habits, physical attributes, or demographic information.

Summary of the dataset:

The dataset for this study includes a comprehensive set of variables that capture various aspects of an individual's lifestyle and health metrics. It begins with the 'Gender' attribute, a categorical variable that identifies the individual's gender. Alongside this, the 'Age' is a continuous variable that records the age of the individual. Measurements of physical attributes

include 'Height' and 'Weight', both continuous variables that record the height in meters and weight in kilograms, respectively.

Health-related factors are well represented, starting with 'Family History with Overweight', a binary variable that indicates whether there is a familial predisposition to being overweight. 'FAVC', or Frequent High Caloric Food Consumption, is another binary variable that checks if the individual often consumes high-calorie food. The 'FCVC' (Frequency of Vegetable Consumption in Meals) variable, an integer, measures how frequently vegetables are included in the individual's meals. Another dietary habit captured is through 'CAEC' (Consumption of Food Between Meals), a categorical variable that details how often the individual eats between meals.

Lifestyle choices are further detailed with several variables. 'SMOKE' is a binary variable indicating whether the individual smokes. 'CH2O', representing Daily Water Consumption, is a continuous variable that measures the litres of water consumed each day. 'SCC' (Calorie Consumption Monitoring) is a binary variable that shows whether the individual monitors their calorie intake. The 'FAF' (Frequency of Physical Activity) is a continuous variable that quantifies the frequency of physical activity in a week. 'TUE' (Time Using Technological Devices) is an integer variable that counts the hours spent using technological devices daily.

Additionally, 'CALC' (Alcohol Consumption Frequency) is a categorical variable that details how frequently the individual consumes alcohol. The 'MTRANS' variable, which stands for the Most Used Mode of Transport, categorizes the primary means of transportation used by the individual.

The target variable, 'NObeyesdad', is categorical and crucial for the analysis and modelling phase. It categorizes individuals into various levels of obesity ranging from Insufficient Weight to several obesity levels like Overweight Level I, Overweight Level II, Obesity Type I, Obesity Type II, and Obesity Type III. This variable allows researchers to understand how various factors contribute to different obesity categories, providing a nuanced view of the interplay between lifestyle choices and health outcomes.

This classification allows for a nuanced understanding of obesity, providing a spectrum rather than a binary assessment, which is crucial for targeted health interventions and research into obesity-related trends.

One of the main merits of this dataset is the lack of the missing values for all the variables; this way, all the records have a full set of data points. As a result, the statistical analysis and machine learning implementation based on this definition is simple, and the research proceeds can start without primary imputation steps taken to address the issue of missing observations.

```
estimation_of_obesity_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2111 entries, 0 to 2110
Data columns (total 16 columns):
 #   Column                          Non-Null Count  Dtype
---  ------                          --------------  -----
 0   Gender                          2111 non-null   object
 1   Age                             2111 non-null   float64
 2   Height                          2111 non-null   float64
 3   Weight                          2111 non-null   float64
 4   family_history_with_overweight  2111 non-null   object
 5   FAVC                            2111 non-null   object
 6   FCVC                            2111 non-null   float64
 7   NCP                             2111 non-null   float64
 8   CAEC                            2111 non-null   object
 9   SMOKE                           2111 non-null   object
 10  CH2O                            2111 non-null   float64
 11  SCC                             2111 non-null   object
 12  FAF                             2111 non-null   float64
 13  TUE                             2111 non-null   float64
 14  CALC                            2111 non-null   object
 15  MTRANS                          2111 non-null   object
dtypes: float64(8), object(8)
memory usage: 264.0+ KB
```

Fig 5: Dataset Info

```
estimation_of_obesity_df.isna().sum()
```

```
Gender                            0
Age                               0
Height                            0
Weight                            0
family_history_with_overweight    0
FAVC                              0
FCVC                              0
NCP                               0
CAEC                              0
SMOKE                             0
CH2O                              0
SCC                               0
FAF                               0
TUE                               0
CALC                              0
MTRANS                            0
dtype: int64
```

Fig 6: Null values Count of Dataset

The Estimation of Obesity Levels Dataset is highly beneficial for research in the fields of nutritional science, epidemiology, and public health policy making. It allows reviewing the detailed variables to find the most significant predictors that explain the levels of obesity. In this way, one can determine the peculiar behavioural patterns and design strategies to prevent and resolve obesity problems within the various communities. The dataset is helpful not only for creating predictive models but also to better recognize how lifestyle and demographic variables interact and cause the different levels of obesity.

3. Dataset Cleaning & Preprocessing

a) Bank Marketing Dataset: Data cleaning and preprocessing of the Bank Marketing dataset are essential steps in the process of preparing data to apply machine learning models in a correct and accurate way. Data cleaning and preprocessing make data more accurate, complete, and relevant by identifying and resolving data-related problems that would otherwise distort the results, the performance of the model. The Bank Marketing dataset was cleaned and pre-processed as follows:

Data Cleaning:

Removing Irrelevant Columns: At first, the columns which are not relevant to the analysis or which have a too large percent of missing values are removed. Those are **'contact'** and **'poutcome'**: they either have low affecting potential on the predictive modelling or have too little information and are not reliable to use.

Handling Missing Values: Rows where the **'job'** is missing are excluded from the dataset, as these only account for 288 records. The 'job' category is crucial since it can significantly impact an individual's likelihood of subscribing to a bank term deposit. Missing values in the **'education'** column are filled with the most frequent value (mode). This approach is chosen because education level is a categorical variable, and using the mode provides a reasonable approximation that maintains the distribution of data.

```
# Drop contact,poutcome columns from bank_marketing_df
bank_marketing_df = bank_marketing_df.drop(columns=['contact', 'poutcome'])

# Step 1: Drop rows where 'job' is NaN
bank_marketing_df = bank_marketing_df.dropna(subset=['job'])

# Step 2: Fill missing 'education' values with the most frequent value
most_frequent_education = bank_marketing_df['education'].mode()[0]
bank_marketing_df['education'].fillna(value=most_frequent_education, inplace=True)

bank_marketing_target_df = bank_marketing_target_df.loc[bank_marketing_df.index]
```

Fig 7: Code for Bank Marketing Dataset cleaning

Data Preprocessing:

Encoding Categorical Variables: Categorical variables such as 'job', 'marital', 'education', 'default', 'housing', 'loan', and 'month' are converted from text to a numeric format. This transformation is crucial for machine learning algorithms that require numerical input:

Binary variables like 'default', 'housing', and 'loan' are encoded into binary formats (1 for 'yes', 0 for 'no').

Ordinal variables such as 'education' and 'marital' status are mapped to integers reflecting their ordered nature.

Nominal variables like 'job' and 'month', which do not have an intrinsic ordering, are handled using OneHotEncoding to create dummy variables that prevent any ordinal interpretation by the model. **[3]**

```
bank_marketing_df.loan = bank_marketing_df.loan.replace({"yes": 1, "no" : 0})
bank_marketing_df.housing = bank_marketing_df.housing.replace({"yes": 1, "no" : 0})
bank_marketing_df.default = bank_marketing_df.default.replace({"yes": 1, "no" : 0})
bank_marketing_df["marital"] = bank_marketing_df["marital"].replace({'married':2, 'single':1, 'divorced':0})
bank_marketing_df["education"] = bank_marketing_df["education"].replace({'tertiary':2, 'secondary':1, 'primary':0})
```

```
# Initialize OneHotEncoder
encoder = OneHotEncoder(drop='first', sparse=False)  # drop='first' to avoid multicollinearity

# Assuming 'job' and 'month' are the columns you're encoding
ct = ColumnTransformer(transformers=[('encoder', encoder, ['job', 'month'])], remainder='passthrough')

# Fit and transform the data
# Ensure that bank_marketing_df is defined and includes 'job' and 'month' columns
bank_marketing_df_encoded = ct.fit_transform(bank_marketing_df)

# Convert back to DataFrame (optional step to retain DataFrame structure)
encoded_feature_names = ct.get_feature_names_out()
bank_marketing_df_encoded = pd.DataFrame(bank_marketing_df_encoded, columns=encoded_feature_names)
```

Fig 8: Code for Bank Marketing Dataset Preprocessing

Principal Component Analysis (PCA): PCA is applied to scale all the features before paying attention to the variance measured and preventing scale from influencing the behaviour of the variance. It then applies PCA to reduce the dataset dimensionality, commits the most crucial variance in fewer dimensions and ignores the rest. Apart from the intention to visualize the data more efficiently, it also lets the machine algorithms work more quickly.

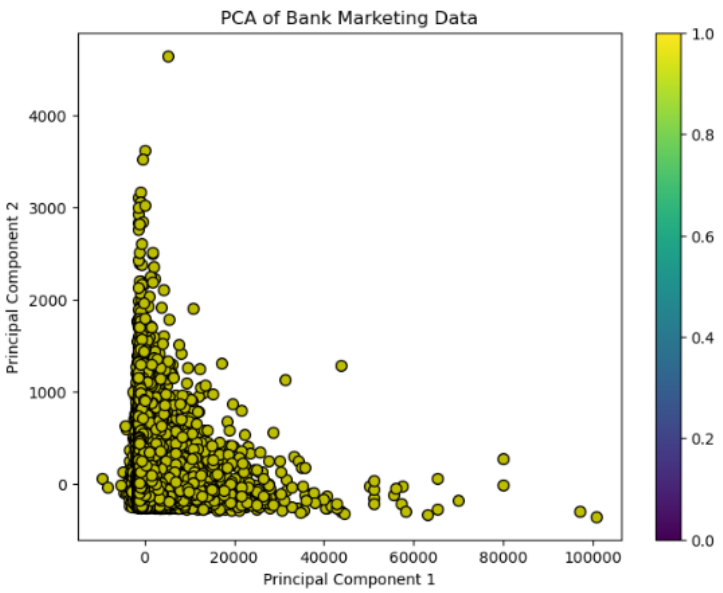

Fig 9: PCA on Bank Marketing Data

Feature Engineering: The transformation processes include combining categories of some variables where appropriate and creating interaction terms to capture the combined effects of different features on the target variable.

Final Preparations: The target variable 'y', indicating whether the client has subscribed to a term deposit, is encoded into binary format (1 for 'yes', 0 for 'no').

```
# Encoding the target variable from 'yes'/'no' to 1/0
bank_marketing_target_df['y'] = bank_marketing_target_df['y'].map({'no': 0, 'yes': 1})
```

Fig 10: Encoding Target Variable code

The dataset is finally structured into a format suitable for modelling, where each feature vector is accompanied by a target variable indicating the outcome.

The Bank Marketing dataset's data cleaning and preprocessing steps are thorough and avoid any major issues or problems, such as missing values and non-numeric data, which could dramatically impact the efficacy of machine learning models. Thus, the analyst ensures that the gathered data could be used for useful analysis and predictive modelling to gain a better understanding of people's behaviours and predict their likelihoods of subscribing to the bank term deposits. Such close attention to detail at this stage is critical to creating accurate lessons and robust predictive models.

b) Online Shoppers Purchasing Intention Dataset: Indeed, to enable an effective analysis of a dataset and create a model, data cleaning and preparation must be implemented. It means that the dataset should be checked for inaccuracies and inconsistencies to enable a reliable result.

Data Cleaning:

The Online Shoppers dataset undergoes minimal data cleaning as the dataset is already in a good state with no missing values across its variables. This absence of null values accelerates the preprocessing phase, allowing more focus on feature engineering and model training.

Data Preprocessing:

Categorical Variable Encoding:

There are several categorical variables present in the dataset that are needed to be transformed to make them suitable for algorithm representation. Transformation using one-hot encoding was performed on 'Month' and 'VisitorType' variables. This encoding resulted in transforming "normalized" categorical variables that could be represented to at most machine learning algorithms which presume that the input variables are independent. For 'VisitorType' which included 'Returning_Visitor', 'New_Visitor', 'Other', we transform each of the categories into a different column. This ensures that the various categories do not bear an ordinal relationship between each other when the model receives them.

```
online_shoppers_df["VisitorType"] = online_shoppers_df["VisitorType"].replace({'Returning_Visitor':2, 'New_Visitor':1, 'Other':0})
```

```
# Initialize OneHotEncoder
encoder = OneHotEncoder(drop='first', sparse=False)  # drop='first' to avoid multicollinearity

# Assuming 'job' and 'month' are the columns you're encoding
ct = ColumnTransformer(transformers=[('encoder', encoder, ['Month'])], remainder='passthrough')

# Fit and transform the data
# Ensure that bank_marketing_df is defined and includes 'job' and 'month' columns
online_shoppers_df_encoded = ct.fit_transform(online_shoppers_df)

# Convert back to DataFrame (optional step to retain DataFrame structure)
encoded_feature_names = ct.get_feature_names_out()
online_shoppers_df_encoded = pd.DataFrame(online_shoppers_df_encoded, columns=encoded_feature_names)
```

Fig 11: Code for Online Shoppers Purchasing Intention Dataset Preprocessing

Feature Engineering:

The crucial importance of the 'VisitorType' variable refers to the fact that different kinds of visitors might behave differently when it comes to purchasing intent. In other words, a visitor might have already been introduced to the client's website, meaning that the tendency to purchase increases with the return. Additionally, concerning the seasonal effects that some date-based features such as 'Month' might yield due to holidays or paydays, encoding this feature increases the dimensionality of the dataset.

Principal Component Analysis (PCA):

After encoding, PCA is used to reduce the dimensionality of the dataset. PCA helps to simplify the model while trying to conserve the important features of the model. It transforms high-dimensional encoded data to low dimensions, which are the principal components that account for most of the variance. The transformation is used to enhance visualization which can lead to better performance when building the model in machine learning. This decreases the likelihood of overfitting.

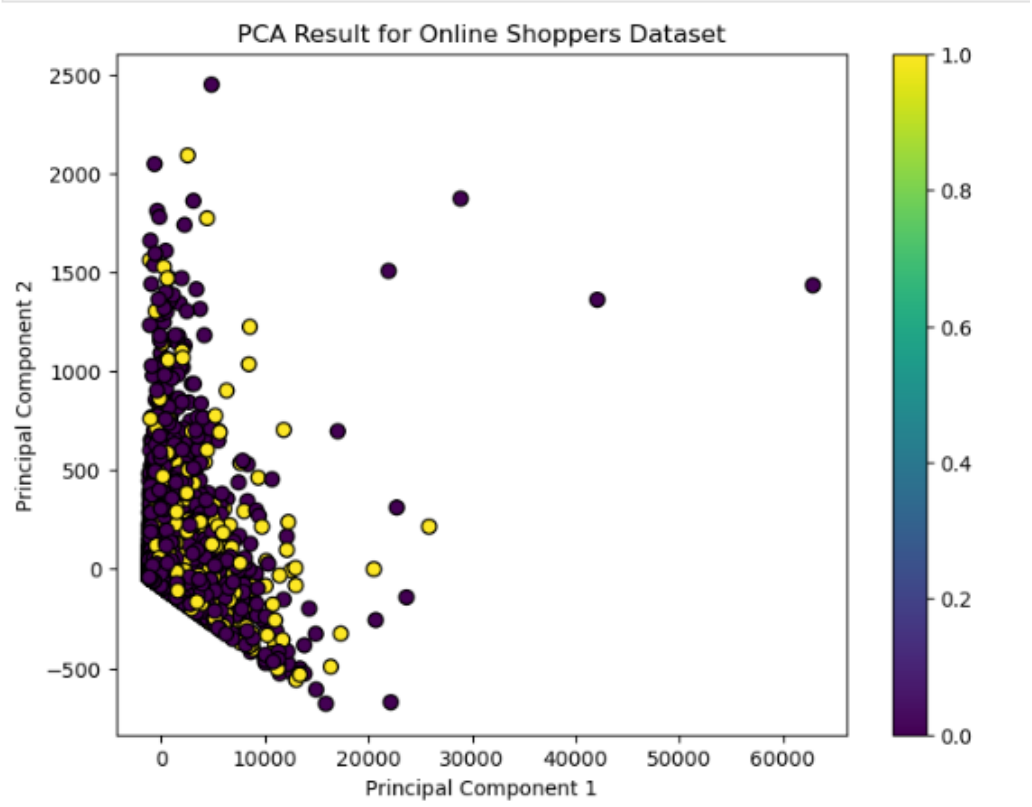

Fig 12: PCA on Online Shoppers Purchasing Intention Data

Final Preparations:

The target variable, "Revenue," is binary, as it shows whether the shopping session led to making a purchase or not. It is encoded from the Boolean type to the integer format. Such binary encoding is suitable for most machine learning algorithms, as the classification tasks require numerical input.

```
[16]: online_shoppers_target_df.Revenue.unique()

[16]: array([False,  True])

[17]: # Encoding the target variable from 'True'/'False' to 1/0
      online_shoppers_target_df['Revenue'] = online_shoppers_target_df['Revenue'].map({False: 0, True: 1})
```

Fig 13: Encoding Target Variable code

A visualization such as the PCA plot of the Online Shoppers dataset provides a graphical representation of the data post-transformation. It shows how sessions are distributed concerning their principal components, which can hint at underlying patterns or clusters in shopping behaviours.

In conclusion, the preprocessing steps applied to the Online Shoppers Purchasing Intention Dataset are necessary to orient all features into appropriate formatting and encoding. While the categorical variables and encoding approaches were addressed, the PCA helped reduce dimensionality to enable suitable exploration using machine learning models. The need for such careful preparation proves how crucial data handling is to getting accurate and meaningful analytical results.

c) Estimation of Obesity Levels Dataset: The Estimation of Obesity Levels Based On Eating Habits and Physical Condition Dataset requires thorough data cleaning and preliminary preparation for the accurate and efficient analysis procedures. This involves the conversion and encoding of the source's categorical variables, making them suitable for the subsequent machine learning algorithms used to generate the predictions.

Data Cleaning:

The Estimation of Obesity Levels Dataset is inherently clean with no missing values, which simplifies the initial stages of data preparation. However, the integrity of the data is further ensured by verifying data types and the consistency of categories within each feature, a crucial step for maintaining the robustness of the dataset.

Data Preprocessing:

Categorical Variable Encoding:

Binary Variables: Features like 'family_history_with_overweight', 'FAVC', 'SMOKE', and 'SCC' are binary and are mapped to 0 and 1 for ease of computation. This binary encoding transforms the categorical data into a format readily acceptable for analysis.

Ordinal Variables: The corresponding feature CAEC (Consumption of Food Between Meals) has an obvious order, and CALC (Alcohol Consumption Frequency) may be said to have an order of frequency. For such features, an ordinal encoding approach is used. Thus, the categories 'Sometimes', 'Frequently' and 'Always' are represented by increasing integer values, which reflect their order on a scale.

```
# Define the mappings
binary_mapping = {'no': 0, 'yes': 1}
gender_mapping = {'Female': 0, 'Male': 1}
quadra_mapping = { 'Sometimes': 1, 'Frequently': 2, 'Always': 3, 'no': 0}

# Columns to map
binary_columns = ['family_history_with_overweight', 'FAVC', 'SMOKE', 'SCC']
quadra_columns = ['CAEC', 'CALC']

# Apply the mappings
estimation_of_obesity_df['Gender'] = estimation_of_obesity_df['Gender'].replace(gender_mapping)

for column in binary_columns:
    if column in estimation_of_obesity_df.columns:  # Check if the column exists in the DataFrame
        estimation_of_obesity_df[column] = estimation_of_obesity_df[column].replace(binary_mapping)

for column in quadra_columns:
    if column in estimation_of_obesity_df.columns:  # Check if the column exists in the DataFrame
        estimation_of_obesity_df[column] = estimation_of_obesity_df[column].replace(quadra_mapping)
```

Fig 14: Categorical Variable Encoding Code for Preprocessing

One-Hot Encoding:

On the other hand, due to the intricacy of the categorical variables, such as 'MTRANS', comprising 'Public Transportation', 'Walking', 'Automatic', 'Motorbike', and 'Bike', the one-hot encoding takes place. In turn, this method turns each category into the new binary variable that does not have any order or weight. As a result, the model analyses the variable without presupposed shortages of proportional one to another, thus affecting the results of analysis.

```
# Initialize OneHotEncoder
encoder = OneHotEncoder(drop='first', sparse=False)  # drop='first' to avoid multicollinearity
ct = ColumnTransformer(transformers=[('encoder', encoder, ['MTRANS'])], remainder='passthrough')

# Fit and transform the data
estimation_of_obesity_df_encoded = ct.fit_transform(estimation_of_obesity_df)

# Convert back to DataFrame (optional step to retain DataFrame structure)
encoded_feature_names = ct.get_feature_names_out()
estimation_of_obesity_df_encoded = pd.DataFrame(estimation_of_obesity_df_encoded, columns=encoded_feature_names)
```

Fig 15: One-Hot Encoding Code for Bank Marketing Dataset Preprocessing

PCA (Principal Component Analysis):

Following the encoding, PCA is used to reduce the dataset's dimensionality. It is crucial as PCA makes it possible to boil down the dataset to the most illustrative components which still maintain the bulk parts of the information. The transformation through PCA is needed for an effective visualization of the dataset as such visualization can show the patterns not visible in the high-dimensional original data. **[4]**

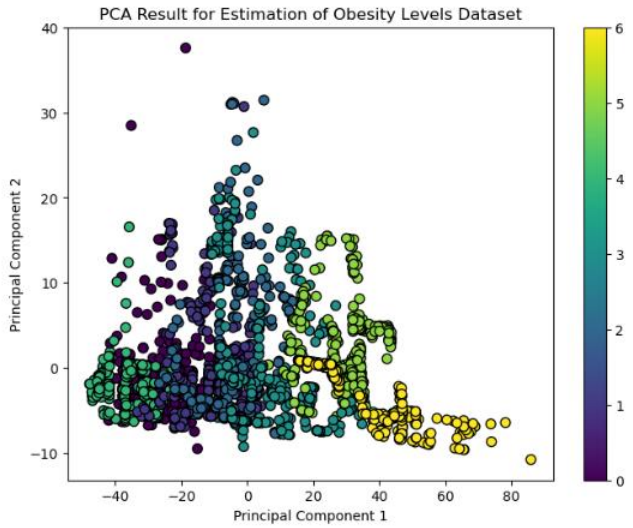

Fig 16: PCA on Online Shoppers Purchasing Intention Data

Final Preparations:

The target variable 'NObeyesdad', representing categories of obesity levels such as 'Normal Weight', 'Overweight Level I', 'Obesity Type I', etc., is encoded into numerical labels as shown in the Figure 17. This encoding facilitates the application of machine learning classification algorithms by converting the categorical output into an interpretable numeric format.

```
estimation_of_obesity_target_df['NObeyesdad'].unique()
```

```
array(['Normal_Weight', 'Overweight_Level_I', 'Overweight_Level_II',
       'Obesity_Type_I', 'Insufficient_Weight', 'Obesity_Type_II',
       'Obesity_Type_III'], dtype=object)
```

```
category_mapping = {
    'Normal_Weight': 0,
    'Overweight_Level_I': 1,
    'Overweight_Level_II': 2,
    'Obesity_Type_I': 3,
    'Insufficient_Weight': 4,
    'Obesity_Type_II': 5,
    'Obesity_Type_III': 6
}
# Applying the mapping to the DataFrame
estimation_of_obesity_target_df['NObeyesdad'] = estimation_of_obesity_target_df['NObeyesdad'].map(category_mapping)
```

Fig 17: Encoding Target Variable code

The PCA visualization of the dataset indeed demonstrated the ability of dimensionality reduction, as it provided the distribution of data points on the first two principal components. This information may be helpful in terms of recognizing natural clusters within the data, which could possibly be associated with various levels of obesity, and developing a more informed approach in terms of analysis.

The above-listed data cleaning and preprocessing actions performed on the Estimation of Obesity Levels Dataset lead to the improved applicability of the data for thorough analytical tasks. After the encoding of the categorical variables, PCA algorithm application, and all the other preparatory steps, the data has been transformed into a solid basis for the application of predictive methodologies and further analysis, all of which are focused on identifying insights concerning the factors affecting the level of obesity.

IV. Model Selection

In the context of evaluating different machine learning techniques for the project aimed at comparing the effectiveness of various models across 3 distinct datasets, several classifiers have been strategically selected based on their robustness, interpretability, and common application in predictive analytics.

Random Forest Classifier: The first approach is the Random Forest Classifier. When initialized with both sklearn and xgboost libraries, this model is attractive due to the ensemble method. It means that several decision trees are involved in the decision-making, thus enhancing precision and robustness of the forecasts. Complementing this by choosing 100 trees

allows for a reasonable level of complexity required to fit all important trends and prevents overfitting. Finally, the fixed random state guarantees the repeatability of the results. Overall, this particular model is most useful for models with large amounts of data and numerous features. That is, it can gauge various characteristic significance, thus being a reliable choice for difficult decision-making situations.

XGBoost's Random Forest Classifier: Using the XGBoost implementation of the Random Forest can be beneficial in terms of performance and speed, particularly in large datasets. The XGBoost model's version is based on an optimized gradient boosting framework, which may be indispensable for improving the standard random forest's performance, especially when it comes to fine-tuning and scalability. The model is preferred based on empirical data of its efficiency on the Kaggle platform and everyday usage.

Decision Tree Classifier: The Decision Tree Classifier from the sklearn library is used to produce a readable display of the decision-making process. The need of a decision tree for this analysis is dictated by the ability of such models to build "paths" from each of the observations towards the outcomes presented, ideal for an initial exploratory analysis to grasp the ways each of the features directly impacts the target variable. Moreover, the tree-based model has always been preferred when one seeks to simplify the model explanation for non-technical stakeholders, as the structure of the tree can be very accurate but easily understood.

LightGBM Decision Tree Classifier: Additionally, selecting the LightGBM Decision Tree Classifier brings a gradient boosting framework that is well-known for its speed and efficiency, specifically when using extensive data sets. Although the LightGBM model is trained to function as a single tree in the present case study, it proves valuable in showcasing the effects of making small parameter alterations to the model on performance and results. The LightGBM model becomes essential in situations where top speed and model accuracy are crucial aspects to deliver, and its utilization in this context is to claim its efficiency compared to conventional machine learning methods.

The models are chosen with the focus on the varying types of machine learning algorithms – from ensemble methods to boosted trees – and ensure that the performance analysis across the datasets produces a more wholesome diversity of results. They are selected with the emphasis on their respective advantage and the frequent application case, as well as the type of analysis provided on the basis of the understood relations and structures of the dataset. The approach to select models is balanced and rational due to the arms of creating robust results in the project, fit for future research or real cases application.

Model Initialization

Initialize a RandomForestClassifier using sklearn and xgb libraries

```
# Initialize a RandomForestClassifier with 100 trees and a fixed random state for reproducibility.
rf_classifier = RandomForestClassifier(n_estimators=100, random_state=42)

# Initialize XGBoost's random forest classifier with 100 trees and a fixed random state.
xgb_rf = xgb.XGBRFClassifier(n_estimators=100, random_state=42)
```

The Random Forest model involves several decision trees, combines predictions and, thus, is more accurate and stable, while avoids overfitting of each of the trees. Given that every tree is trained on a different subset of the data, the model becomes less exposed while combing all the trees' predictions. For that reason, the Random Forest model does better to generalize since such an approach helps in getting diverse exposure. The process above is further known as bagging, which makes a more reliable decision about the final output due to lower sensitivity to the noise or changes in the data. Such an ensemble method is particularly good with similar data when all the features are interrelated.

In contrast, the random forest of XGBoost leverage gradient boosting to enhance the learning process. When learning, the trees are built sequentially with the individual tree correcting the errors of his predecessor. This approach is additive in that the models keep enhancing the accuracy of the model by correcting the residuals of the earlier trees, which makes it have a marginal optimal performance. XGBoost is effective in large datasets due to its efficiency in managing complex data structures and pruning trees applications speed, which makes it preferred in most data science challenges.

Initialize a Decision Tree Classifier using sklearn and LightGBM libraries

```
# Initialize the Decision Tree Classifier from Scikit-Learn
dt_classifier_sklearn = DecisionTreeClassifier(random_state=42)

# Initialize the LightGBM Decision Tree Classifier
# For a single tree, set num_boost_round to 1 and boost_from_average to False
dt_classifier_lgbm = lgb.LGBMClassifier(boosting_type='gbdt',  # Gradient Boosting Decision Tree
                                        num_leaves=31,         # Max number of leaves in one tree
                                        max_depth=-1,          # No limit on depth, since we want a single tree
                                        learning_rate=0.1,     # Learning rate, you can adjust this if necessary
                                        n_estimators=1,        # Number of boosted trees to fit, 1 for a single tree
                                        random_state=42,       # Seed for reproducibility
                                        boost_from_average=False) # This should be False for a single decision tree
```

We started with a basic Decision Tree Classifier from Sklearn, known for its simplicity and clear decision-making insights. Each node represents a feature, making it easy to understand and interpret. It is great for initial analysis when understanding feature impact is crucial.

Then, we explored LightGBM, an advanced version of decision trees tailored for massive datasets. It corrects errors from previous trees, making it faster and using less memory. LightGBM is ideal for large datasets with complex interactions, ensuring minimal processing time and memory usage.

Evaluation Metrics

Several metrics were used to evaluate the models:

Accuracy: This metric assessed how accurately the models classified clients' likelihood to subscribe to a term deposit.

Precision and Recall: These metrics were used to determine the models' ability to correctly identify true positives from the predicted positives.

F1 Score: This score, which merges precision and recall into a unified metric, was utilized to gauge the models' effectiveness in achieving a balance between precision and recall.

The evaluation process also included an analysis of confusion matrices to visually represent how well the models distinguished between the classes (subscribed vs. not subscribed). **[5]**

V. Model Training and Evaluation

a) Bank Marketing Dataset:

Based on the Bank Marketing dataset, several machine learning models have been used to examine and predict whether a customer will subscribe to a bank term deposit. The RandomForestClassifier and XGBRFClassifier models are extracted from sklearn and xgboost libraries, while DecisionTreeClassifier is from the sklearn and LightGBM libraries. Therefore, the classification models were selected from the sklearn and xgboost library which makes it easier to compare the efficiency of the different models used based on the classification problem handled.

Data Preparation

To promote the construct and generalization efficiency of the validation process, the data was split into training and test sets. About 70% of the data was used for training models while the remaining 30% was used to test the model. To maintain the importance of each feature, the dataset was also normalized using feature scaling. It ensured that certain features with higher scales would not heavily weight the model's outcome.

Data splitting and modelling functions

[1]:
```
def splitting_the_data(df, target_df):
    # Split data into train and test sets
    X_train, X_test, y_train, y_test = train_test_split(df, target_df, test_size=0.3, random_state=42)

    # Scale features
    scaler = StandardScaler()
    X_train_scaled = scaler.fit_transform(X_train)
    X_test_scaled = scaler.transform(X_test)

    return X_train_scaled, X_test_scaled, y_train, y_test

def train_model(X_train, X_test, y_train, y_test, model):
    # Train model
    model.fit(X_train, y_train)
    # Predict on test set
    y_pred = model.predict(X_test)
    y_test = y_test
    conf_mat = confusion_matrix(y_test, y_pred)
    # Output results
    print(f"Model: {model.__class__.__name__}")
    print("Confusion Matrix:\n", conf_mat)
    print("Classification Report:\n", classification_report(y_test, y_pred))

    # Evaluation for multiclass classification
    precision = precision_score(y_test, y_pred, average='weighted')
    recall = recall_score(y_test, y_pred, average='weighted')
    f1 = f1_score(y_test, y_pred, average='weighted')
    acc_score = accuracy_score(y_test, y_pred)

    print(f'Precision: {precision}')
    print(f'Recall: {recall}')
    print(f'F1 Score: {f1}')
    print(f'Accuracy: {acc_score}')
    print("\n")
    return [conf_mat, {'F1 Score': f1, 'Accuracy': acc_score}]
```

Model Training

There were several preparatory procedures to evaluate machine learning models on the Bank Marketing dataset. Data were pre-processed by splitting them into training and test subsets so that both could be an unbiased sample of representative data from the dataset. Benchmark testing the former step is essential to explore if machines could be generated that later can work on unseen data.

```
# Splitting the data
X_train, X_test, y_train, y_test = splitting_the_data(bank_marketing_df_encoded, bank_marketing_target_df)

print("Evaluating Random Forest classifier on Bank Marketing Dataset")
bank_marketing_rf_scores = train_model(X_train, X_test, y_train, y_test, rf_classifier)

print("Evaluating XGBoost's random forest classifier on Bank Marketing Dataset")
bank_marketing_xgb_rf_scores = train_model(X_train, X_test, y_train, y_test, xgb_rf)

print("Evaluating  Decision Tree Classifier on Bank Marketing Dataset")
bank_marketing_dt_scores = train_model(X_train, X_test, y_train, y_test, dt_classifier_sklearn)

print("Evaluating LightGBM Decision Tree Classifier on Bank Marketing Dataset")
bank_marketing_lgbm_scores = train_model(X_train, X_test, y_train, y_test, dt_classifier_lgbm)
```

After that, multiple classifiers were conducted to understand the power of at their application for customer subscription for the bank term deposit. The used models were RandomForestClassifier from sklearn, XGBRFClassifier from xgboost, DecisionTreeClassifier from sklearn, and representative of LightGBM Decision Tree Classifier. Their choice gave opportunity research of the classification task approaches.

The Random Forest Classifier is applicable because the algorithm combines many decision trees, a feature that improves accuracy and reducing overfitting-ease of use in dataset variance and working with many features. The second applicable class is the XGBRFClassifier is applicable for the large dataset works because of sequential processing using gradient boosting applied to random forests in improving models by sequentially improving accuracy by correcting model errors using previous trees. The third class is the Decision Tree Classifier from Sklearn which is appropriate for use as the easiest predictor is easy to understand and clarifies decision making, which is ideal for a rough analysis. Lastly, is LightGBM Decision Tree Classifier is efficient considering it will handle large datasets by making sequential trees with the use of gradient boosting in the creation, thereby ensuring the process is fast. **[6]**

Finally, the performance of these models was thoroughly tested based on numerous metrics, including the accuracy, precision, recall, and F1 score. These step-by-step evaluations helped to outline the efficiency of every classifier and its suitability for the analyzed dataset. Additionally, the analysis emphasized the application areas in which every model could be used to develop effective stages for customer relationship management in the banking sector.

Results and Insights

The evaluation of four distinct classifiers on the Bank Marketing dataset reveals significant insights into the predictive performance and reliability of each model. Below is a detailed analysis of the evaluation metrics derived from the performance of each classifier:

```
Evaluating Random Forest classifier on Bank Marketing Dataset
Model: RandomForestClassifier
Confusion Matrix:
 [[11601   324]
 [  989   563]]
Classification Report:
              precision    recall  f1-score   support

           0       0.92      0.97      0.95     11925
           1       0.63      0.36      0.46      1552

    accuracy                           0.90     13477
   macro avg       0.78      0.67      0.70     13477
weighted avg       0.89      0.90      0.89     13477

Precision: 0.8884269496583985
Recall: 0.9025747569933962
F1 Score: 0.8906145186360269
Accuracy: 0.9025747569933962
```

Fig 21: Random Forest Classifier Metrics

```
Evaluating XGBoost's random forest classifier on Bank Marketing Dataset
Model: XGBRFClassifier
Confusion Matrix:
 [[11258   667]
 [  704   848]]
Classification Report:
              precision    recall  f1-score   support

           0       0.94      0.94      0.94     11925
           1       0.56      0.55      0.55      1552

    accuracy                           0.90     13477
   macro avg       0.75      0.75      0.75     13477
weighted avg       0.90      0.90      0.90     13477

Precision: 0.8972239959415047
Recall: 0.8982711285894487
F1 Score: 0.8977362917737322
Accuracy: 0.8982711285894487
```

Fig 22: XGBoost's Random Forest Metrics

```
Evaluating  Decision Tree Classifier on Bank Marketing Dataset
Model: DecisionTreeClassifier
Confusion Matrix:
 [[11014   911]
 [  843   709]]
Classification Report:
              precision    recall  f1-score   support

           0       0.93      0.92      0.93     11925
           1       0.44      0.46      0.45      1552

    accuracy                           0.87     13477
   macro avg       0.68      0.69      0.69     13477
weighted avg       0.87      0.87      0.87     13477

Precision: 0.8723310004977094
Recall: 0.8698523410254507
F1 Score: 0.871061300424601
Accuracy: 0.8698523410254507
```

Fig 23: Decision Tree Classifier Metrics

```
Model: LGBMClassifier
Confusion Matrix:
 [[11450   475]
 [  908   644]]
Classification Report:
              precision    recall  f1-score   support

           0       0.93      0.96      0.94     11925
           1       0.58      0.41      0.48      1552

    accuracy                           0.90     13477
   macro avg       0.75      0.69      0.71     13477
weighted avg       0.89      0.90      0.89     13477

Precision: 0.8861031409706784
Recall: 0.8973807227127699
F1 Score: 0.8899777596337468
Accuracy: 0.8973807227127699
```

Fig 24: LGBM's Decision Tree Metrics

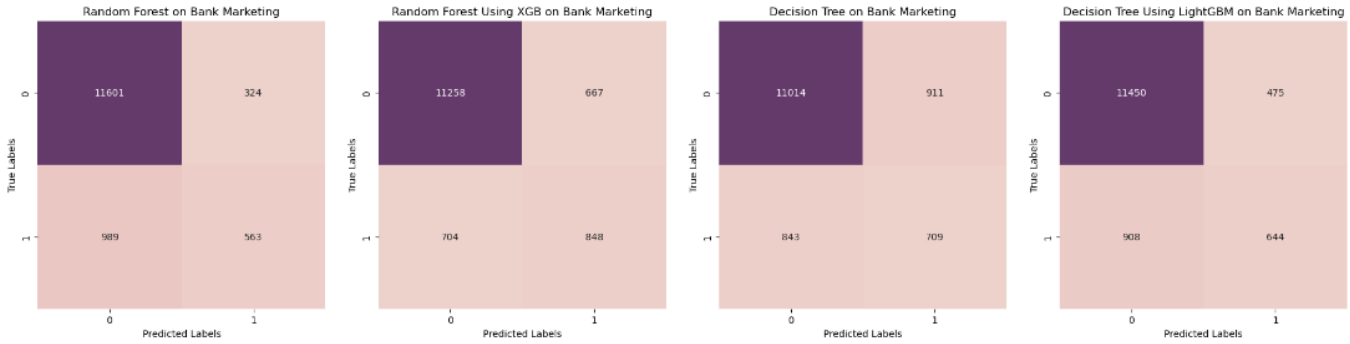

Fig 25: Confusion Matrices of Bank Marketing Dataset Evaluation

RandomForestClassifier:

Confusion Matrix: The matrix shows that the RandomForestClassifier correctly predicted non-subscribers (class 0) 11,601 times and subscribers (class 1) 563 times, with 989 false negatives and 324 false positives. This result indicates a strong ability to predict non-subscribers.

Accuracy: The model achieved an overall accuracy of approximately 90.25%, indicating a high level of correctness in its predictions across both classes.

Precision, Recall, and F1 Score: The precision for class 0 is high at 0.92, showing that the model is reliable when predicting non-subscribers. However, the precision drops significantly for class 1 to 0.63, reflecting some challenges in accurately predicting subscribers. The F1 Score of 0.90 and a recall of 0.90 for class 0 are indicative of a balanced performance for non-subscribers, which is not as pronounced for subscribers (F1 Score and recall of 0.36 and 0.36, respectively).

XGBRFClassifier:

Confusion Matrix: This model predicted non-subscribers with higher precision, with 11,258 correct predictions and 704 false negatives. However, it also had a higher number of false positives (667).

Accuracy: Achieved an accuracy of 89.87%, slightly lower than the RandomForestClassifier.

Precision, Recall, and F1 Score: Exhibited high precision (0.94) and F1 Score (0.94) for class 0 but struggled with class 1 predictions, showing a similar pattern to the RandomForest, with all scores under 0.56.

DecisionTreeClassifier:

Confusion Matrix: Showed a balance of true positives and false positives with 11,014 correct predictions for non-subscribers and 843 for subscribers. False negatives and positives were 709 and 911, respectively.

Accuracy: Recorded lower accuracy (86.98%) compared to the RandomForest models.

Precision, Recall, and F1 Score: Demonstrated high precision and recall for class 0 (0.93 and 0.92 respectively), but significantly lower for class 1 (precision of 0.44 and recall of 0.46).

LGBMClassifier:

Confusion Matrix: Demonstrated a similar trend with strong predictions for non-subscribers (11,450 correct predictions) but weaker outcomes for subscribers (908 correct predictions).

Accuracy: This model had an accuracy of 89.74%, competitive with the XGBRF model.

Precision, Recall, and F1 Score: Showed strong performance for class 0 with a precision of 0.93 and an F1 Score of 0.94. Class 1 scores were lower, highlighting a recurring challenge in predicting subscriber behaviour accurately.

Summary: A commonality among all the models is that when it comes to predicting non-subscription, the accuracy, precision, and recall are high than in predicting subscription. This could be affected by the distribution of the data or feature impact, whereby the features that drive non-subscription are the strongest or the easiest to model. Such information is crucial in the adjustment of models and the strategy to focus on potential subscribers.

The results indicated varying degrees of success among the models:

Random Forest and XGBoost's RF showed robust performance with high accuracy, suggesting that ensemble methods are particularly effective for this dataset. Decision Trees provided valuable insights into the feature importance and decision-making process but generally offered lower accuracy and F1 scores compared to the ensemble methods.

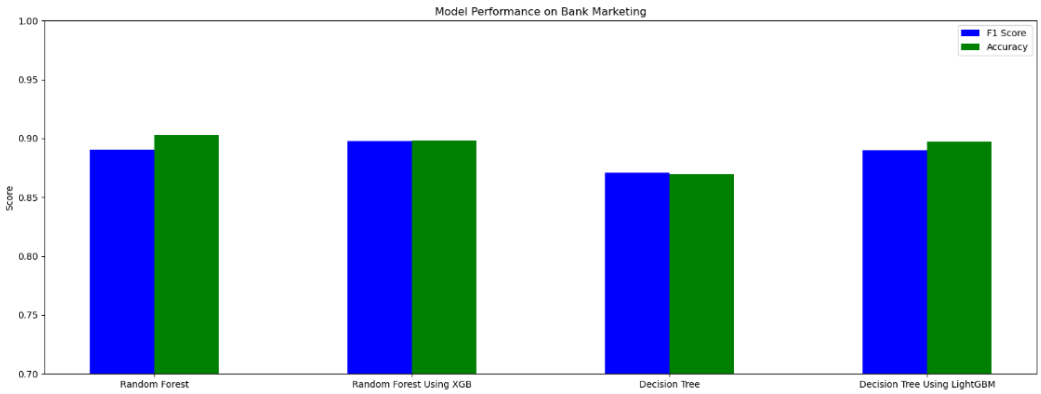

Fig 26: Bar plot of F1-score and Accuracy of 4 models

Comparative analysis of machine learning models with Bank Marketing dataset enables one to realize the strengths and drawbacks of the algorithms. Thus, based on the attained results, **Random Forest and XGBoost's Random Forest Classifiers** ensemble methods were the most appropriate to the dataset's peculiarities. This data could be used in the future research types and be advantageous for practical implementation in marketing strategies. Thus, this will assist in making the marketing strategies more targeted and concentrating on the strengths to capture the customers' interest and prompt them to deposit.

b) Online Shoppers Purchasing Intention Dataset:

The Online Shoppers Purchasing Intention dataset has attracted the usage of various machine learning models, as several models can be applied to analyse observed data to predict customer purchasing patterns based on online activities. I chose the RandomForestClassifier and XGBRFClassifier from the established sklearn and xgboost libraries, respectively. These models used the DecisionTreeClassifier and LightGBM Decision Tree Classifier from the sklearn and LightGBM libraries. The models were selected from libraries, such as sklearn, xgboost, and LightGBM, among many others. Using the models from famous models in the field of classification will offer much insight into how different algorithms deal with the same classification issues and expound into the best ways of predicting online purchasing intention thus shedding light on areas of strength and role in complex phenomena, such as behaviour of online shoppers.

Data Preparation

Thus, the first step during the thorough assessment of the Online Shoppers Purchasing Intentions dataset was the division of the data into appropriate portions and training of various models. It was possible to achieve through successful predictive performance evaluation. On the one hand, the processed data were scaled, and the training-testing set split was completed

at 70-30 percent of the total volume for validation. On the other side, a range of advanced classifiers was used to determine their predictability of the user intention based on their browsing behaviour.

Fig 27:

```
# Splitting the data
X_train, X_test, y_train, y_test = splitting_the_data(online_shoppers_df_encoded, online_shoppers_target_df)

print("Evaluating Random Forest classifier on Online Shoppers Purchasing Intention Dataset")
online_shoppers_rf_scores = train_model(X_train, X_test, y_train, y_test, rf_classifier)

print("Evaluating XGBoost's random forest classifier on Online Shoppers Purchasing Intention Dataset")
online_shoppers_xgb_rf_scores = train_model(X_train, X_test, y_train, y_test, xgb_rf)

print("Evaluating Decision Tree Classifier on Online Shoppers Purchasing Intention Dataset")
online_shoppers_dt_scores = train_model(X_train, X_test, y_train, y_test, dt_classifier_sklearn)

print("Evaluating LightGBM Decision Tree Classifier on Online Shoppers Purchasing Intention Dataset")
online_shoppers_lgbm_scores = train_model(X_train, X_test, y_train, y_test, dt_classifier_lgbm)
```

Model Training

For the purpose of this analysis of the Online Shoppers Purchasing Intention Dataset, several types of machine learning models were tested based on their ease of implementation, accuracy, and minimal overfitting. Specifically, RandomForestClassifier, XGBRFClassifier, DecisionTreeClassifier, and LightGBM's Decision Tree Classifier were selected due to distinct classification methods. The rationale behind this selection was the models' ability to efficiently work with data that has complex interrelations between different features, one that is typical for online shopping behaviour.

The process began with splitting the data into training and testing sets using a 70-30 ratio to ensure a robust evaluation framework. This partition allows for training the models on a substantial portion of the data while reserving a significant segment for testing, maintaining the integrity of model evaluations by avoiding overfitting.

Each model was trained and tested:

RandomForestClassifier: Utilized for its ensemble method that combines multiple decision trees to reduce the risk of overfitting and provide a balanced approach to various types of data irregularities.

XGBRFClassifier: Leveraged for its enhancement of the standard random forest through gradient boosting, which focuses on optimizing model predictions progressively, making it highly suitable for dynamic environments like online shopping.

DecisionTreeClassifier from sklearn: This model offers a straightforward, interpretable framework, making it useful for initial data insights and understanding direct relationships between features. **[7]**

LightGBM Decision Tree Classifier: Recognized for its efficiency with large datasets and complex features, this model uses gradient boosting to construct trees sequentially, significantly enhancing processing speed and model accuracy.

This structured approach in model training and evaluation ensures a comprehensive assessment of each classifier's performance, highlighting their strengths and limitations in the context of predicting online shopping behaviours.

Results and Insights

Random Forest Classifier: Random Forest is efficient and robust against overfitting in moderating by taking the average of different decision trees. The test result gave 0.92 as the precision and 0.96 as recall for class 0, which means that the model can predict the majority class with high accuracy. For class 1, although the precision score was good enough with 0.73, the recall was not greater than 0.57, which the model's ability to predict all possible purchasers.

XGBoost's Random Forest Classifier: Known for its performance and speed, this model slightly improved the recall for purchasers to 0.65 while maintaining a high precision and recall for non-purchasers. This model balances error correction progressively through boosting techniques, enhancing decision-making tree by tree.

```
Evaluating Random Forest classifier on Online Shoppers Purchasing Intention Dataset
Model: RandomForestClassifier
Confusion Matrix:
 [[3000  124]
 [ 246  329]]
Classification Report:
              precision    recall  f1-score   support

           0       0.92      0.96      0.94      3124
           1       0.73      0.57      0.64       575

    accuracy                           0.90      3699
   macro avg       0.83      0.77      0.79      3699
weighted avg       0.89      0.90      0.89      3699

Precision: 0.8934443626783621
Recall: 0.8999729656663963
F1 Score: 0.8949953820482054
Accuracy: 0.8999729656663963
```

Fig 28: Random Forest Classifier Metrics

```
Evaluating XGBoost's random forest classifier on Online Shoppers Purchasing Intention Dataset
Model: XGBRFClassifier
Confusion Matrix:
 [[2941  183]
 [ 200  375]]
Classification Report:
              precision    recall  f1-score   support

           0       0.94      0.94      0.94      3124
           1       0.67      0.65      0.66       575

    accuracy                           0.90      3699
   macro avg       0.80      0.80      0.80      3699
weighted avg       0.90      0.90      0.90      3699

Precision: 0.8952439007836653
Recall: 0.8964585022979183
F1 Score: 0.8958221918841808
Accuracy: 0.8964585022979183
```

Fig 29: XGBoost's Random Forest Metrics

Decision Tree Classifier: The Decision Tree, being a baseline model, was characterized by a clear and interpretable structure, as well as certain performance inadequacies. Thus, it had the lowest accuracy and F1 scores among the tested models. However, it demonstrated a F1 score at the level of 0.69 and precision of 0.92, recall of 0.91 for non-purchasers, which is quite good. Nevertheless, the decision tree was less successful in identifying purchasers, demonstrating only 0.53 precision and 0.56 recall.

LightGBM Decision Tree Classifier: This model performed similarly to the Random Forest, with slightly better handling of the minority class. The precision and recall for purchasers were 0.72 and 0.56, respectively, demonstrating a decent capability in predicting actual purchasers compared to the simpler Decision Tree model.

```
Evaluating Decision Tree Classifier on Online Shoppers Purchasing Intention Dataset
Model: DecisionTreeClassifier
Confusion Matrix:
 [[2840  284]
 [ 251  324]]
Classification Report:
              precision    recall  f1-score   support

           0       0.92      0.91      0.91      3124
           1       0.53      0.56      0.55       575

    accuracy                           0.86      3699
   macro avg       0.73      0.74      0.73      3699
weighted avg       0.86      0.86      0.86      3699

Precision: 0.8588090722848659
Recall: 0.8553663152203298
F1 Score: 0.8569996234478254
Accuracy: 0.8553663152203298
```

Fig 30: Decision Tree Classifier Metrics

```
Model: LGBMClassifier
Confusion Matrix:
 [[3000  124]
 [ 254  321]]
Classification Report:
              precision    recall  f1-score   support

           0       0.92      0.96      0.94      3124
           1       0.72      0.56      0.63       575

    accuracy                           0.90      3699
   macro avg       0.82      0.76      0.79      3699
weighted avg       0.89      0.90      0.89      3699

Precision: 0.8907604194795536
Recall: 0.8978102189781022
F1 Score: 0.8923395700199038
Accuracy: 0.8978102189781022
```

Fig 31: LGBM's Decision Tree Metrics

The confusion matrices of each model helped the user to see the classification accuracy of in the detailed matter. It showed how many true positives, true negatives, false positives, and false negatives each model accounted for. For example, the Random Forest model predicted an accurate non-purchaser of 3000 individuals but considered 246 individuals as a purchaser. This suggests the strong ability to predict with a small number of false positives.

Fig 32: Confusion Matrices of Online Shoppers Purchasing Intention Dataset Evaluation

In terms of model performance metrics across different classifiers, the F1 scores and accuracy were visualized in bar charts, clearly demonstrating each model's effectiveness. These metrics are crucial in selecting a model that not only predicts accurately overall (high accuracy) but also balances the precision and recall effectively (high F1 score), especially important in datasets with imbalanced classes like this one.

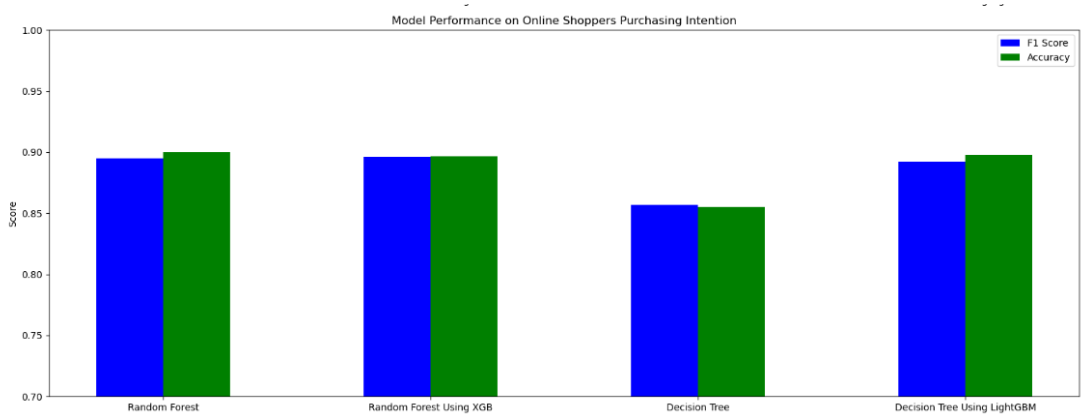

Fig 33: Bar plot of F1-score and Accuracy of 4 models

Overall, this comprehensive evaluation using various machine learning models provides a solid foundation for selecting the most appropriate model for predicting online shopping intentions, with the **RandomForest** and **LightGBM models** showing the most promising results in terms of both accuracy and handling of class imbalance.

c) Estimation of Obesity Levels Dataset: The Estimation of Obesity Levels dataset has been leveraged to apply various machine learning models to predict obesity levels from health and lifestyle data. Models like the RandomForestClassifier and XGBRFClassifier from the sklearn and xgboost libraries, respectively, alongside the DecisionTreeClassifier and LightGBM Decision Tree Classifier from sklearn and LightGBM, were chosen. These selections facilitate a comprehensive analysis across different algorithms to effectively handle the classification of obesity levels based on complex health data interactions.

Data Preparation:

The initial phase involved the methodical splitting of the dataset into training and testing sets with a 70-30 ratio. This structure was critical in ensuring the models were trained on a substantial subset of the data, allowing for robust testing without the risk of overfitting.

Fig 34:

```
# Splitting the data
X_train, X_test, y_train, y_test = splitting_the_data(estimation_of_obesity_df_encoded, estimation_of_obesity_target_df)

print("Evaluating Random Forest classifier on Estimate of Obesity levels dataset")
estimation_of_obesity_rf_scores = train_model(X_train, X_test, y_train, y_test, rf_classifier)

print("Evaluating XGBoost's random forest classifier on Estimate of Obesity levels dataset")
estimation_of_obesity_xgb_rf_scores = train_model(X_train, X_test, y_train, y_test, xgb_rf)

print("Evaluating Decision Tree Classifier on Estimate of Obesity levels dataset")
estimation_of_obesity_dt_scores = train_model(X_train, X_test, y_train, y_test, dt_classifier_sklearn)

print("Evaluating LightGBM Decision Tree Classifier on Estimate of Obesity levels dataset")
estimation_of_obesity_lgbm_scores = train_model(X_train, X_test, y_train, y_test, dt_classifier_lgbm)
```

Model Training: In the model training phase, each classifier was applied to ascertain its efficacy in managing and predicting the nuanced categories of obesity levels:

RandomForestClassifier: This model's ensemble approach, integrating multiple decision trees, helps mitigate overfitting and provides nuanced insights into data variability. **XGBRFClassifier:** Enhanced by gradient boosting, this classifier refines predictions by iteratively focusing on minimizing prior errors, suitable for dynamic data sets like those in healthcare. **DecisionTreeClassifier from sklearn:** Known for its direct interpretative structure, it serves as an excellent baseline to understand feature impact directly without complex transformations. **LightGBM Decision Tree Classifier:** Famed for its high efficiency with large and complex datasets, it uses sequential tree boosting to improve prediction accuracy and speed.

Results and Insights

The Estimation of Obesity Levels dataset provides a rich source for applying machine learning models to predict various obesity levels from health and lifestyle data. This evaluation includes the use of sophisticated models such as RandomForestClassifier, XGBRFClassifier, DecisionTreeClassifier, and LightGBM Decision Tree Classifier, selected from

the libraries sklearn, xgboost, and LightGBM, respectively. These models are chosen for their ability to handle complex data structures typical of health-related information.

```
Evaluating Random Forest classifier on Estimate of Obesity levels dataset
Model: RandomForestClassifier
Confusion Matrix:
 [[83  6  0  0  4  0  0]
 [12 74  2  0  0  0  0]
 [ 2  4 73  0  0  0  0]
 [ 2  0  1 96  0  3  0]
 [ 5  0  0  0 81  0  0]
 [ 0  0  0  1  0 87  0]
 [ 0  0  0  1  0  0 97]]
Classification Report:
              precision    recall  f1-score   support

           0       0.80      0.89      0.84        93
           1       0.88      0.84      0.86        88
           2       0.96      0.92      0.94        79
           3       0.98      0.94      0.96       102
           4       0.95      0.94      0.95        86
           5       0.97      0.99      0.98        88
           6       1.00      0.99      0.99        98

    accuracy                           0.93       634
   macro avg       0.93      0.93      0.93       634
weighted avg       0.93      0.93      0.93       634

Precision: 0.9346443493807693
Recall: 0.9321766561514195
F1 Score: 0.9328279301418638
Accuracy: 0.9321766561514195
```

Fig 35: Random Forest Classifier Metrics

```
Evaluating XGBoost's random forest classifier on Estimate of Obesity levels dataset
Model: XGBRFClassifier
Confusion Matrix:
 [[70 18  0  0  5  0  0]
 [ 7 77  4  0  0  0  0]
 [ 0  4 72  3  0  0  0]
 [ 1  2  5 92  0  2  0]
 [ 1  2  0  0 83  0  0]
 [ 0  0  0  1  0 87  0]
 [ 1  0  0  0  0  0 97]]
Classification Report:
              precision    recall  f1-score   support

           0       0.88      0.75      0.81        93
           1       0.75      0.88      0.81        88
           2       0.89      0.91      0.90        79
           3       0.96      0.90      0.93       102
           4       0.94      0.97      0.95        86
           5       0.98      0.99      0.98        88
           6       1.00      0.99      0.99        98

    accuracy                           0.91       634
   macro avg       0.91      0.91      0.91       634
weighted avg       0.92      0.91      0.91       634

Precision: 0.9152519530961534
Recall: 0.9116719242902208
F1 Score: 0.9119128729671425
Accuracy: 0.9116719242902208
```

Fig 36: XGBoost's Random Forest Metrics

RandomForestClassifier: Showcased strong performance with an overall accuracy of 93.2%. It particularly excelled in classifying the most severe obesity levels with high precision and recall, indicative of its robustness in handling varied data inputs and complex classifications.

XGBRFClassifier: This model leveraged the strengths of both boosting and random forests, achieving a commendable accuracy of 91.2%. It was particularly effective in iteratively correcting errors from previous predictions, enhancing its precision through progressive training iterations.

```
Evaluating Decision Tree Classifier on Estimate of Obesity levels dataset
Model: DecisionTreeClassifier
Confusion Matrix:
 [[74  8  0  0 11  0  0]
 [11 71  6  0  0  0  0]
 [ 0  4 74  0  0  1  0]
 [ 1  0  3 93  0  5  0]
 [ 4  0  0  0 82  0  0]
 [ 0  0  0  2  0 86  0]
 [ 0  0  0  0  0  1 97]]
Classification Report:
              precision    recall  f1-score   support

           0       0.82      0.80      0.81        93
           1       0.86      0.81      0.83        88
           2       0.89      0.94      0.91        79
           3       0.98      0.91      0.94       102
           4       0.88      0.95      0.92        86
           5       0.92      0.98      0.95        88
           6       1.00      0.99      0.99        98

    accuracy                           0.91       634
   macro avg       0.91      0.91      0.91       634
weighted avg       0.91      0.91      0.91       634

Precision: 0.9104644206535912
Recall: 0.9100946372239748
F1 Score: 0.9095924093991946
Accuracy: 0.9100946372239748
```

Fig 37: Decision Tree Classifier Metrics

```
Model: LGBMClassifier
Confusion Matrix:
 [[74 12  1  0  6  0  0]
 [13 70  4  1  0  0  0]
 [ 1  3 67  8  0  0  0]
 [ 3  2  4 89  0  4  0]
 [ 2  3  0  0 81  0  0]
 [ 0  0  0  0  0 88  0]
 [ 0  0  0  0  0  1 97]]
Classification Report:
              precision    recall  f1-score   support

           0       0.80      0.80      0.80        93
           1       0.78      0.80      0.79        88
           2       0.88      0.85      0.86        79
           3       0.91      0.87      0.89       102
           4       0.93      0.94      0.94        86
           5       0.95      1.00      0.97        88
           6       1.00      0.99      0.99        98

    accuracy                           0.89       634
   macro avg       0.89      0.89      0.89       634
weighted avg       0.89      0.89      0.89       634

Precision: 0.8928385127276288
Recall: 0.8927444794952681
F1 Score: 0.892568684413323
Accuracy: 0.8927444794952681
```

Fig 38: LGBM's Decision Tree Metrics

DecisionTreeClassifier: Provided a simpler, yet effective, classification mechanism with an accuracy of 91%. It stood out for its interpretability, which is crucial in medical datasets where understanding the influence of variables is key.

LightGBM Decision Tree Classifier: Known for its efficiency with large data sets, this model achieved an accuracy of 89.3%. It excels in situations requiring rapid processing and was adept at handling the data complexity due to its gradient boosting methodology.

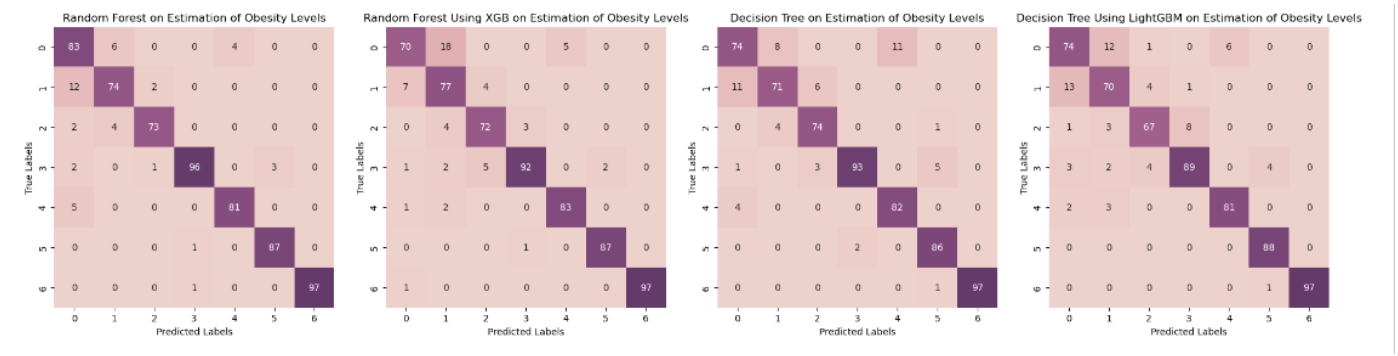

Fig 39: Confusion Matrices of Estimation of Obesity Levels Dataset Evaluation

The performance metrics of classifiers from the confusion matrix evidently provide a clear perspective of how each model can correctly differentiate between different obesity levels. The RandomForestClassifier records high precision and recall for all classes flagging the model as an ideal classifying model. Thus, the model can be used effectively in medical datasets where classifying decency is important as far as patient's health is concerned. It shows how accurate the algorithm is.

LightGBM classifier, on the other hand, reveals a bit lower overall accuracy but demonstrates notable performance in the case of class imbalances, specifically, underrepresented categories. This fact is especially noteworthy for medical datasets that can include conditions with different prevalence rates, all of which still require accurate identification. In these terms, the mentioned properties of the RandomForest and LightGBM classifiers support their actual deployment in medical predictive analytics, becoming essential tools to provide health practitioners with good decision-making support.

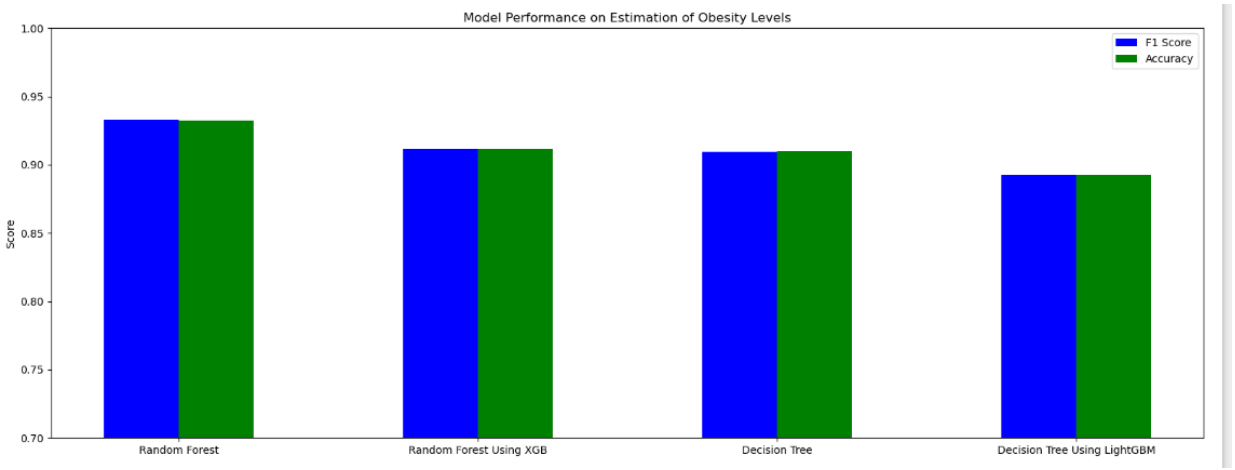

Fig 40: Bar plots of F1-score and Accuracy of 4 models

The bar chart above represents the performance of four different machine learning models in classifying obesity levels based on various health indicators. The models evaluated include the RandomForest, XGBoost-enhanced RandomForest (XGB), a basic Decision Tree, and a LightGBM-enhanced Decision Tree.

From the visualization, it is evident that all models perform fairly well, with accuracy and F1 scores generally above 0.85 across the board, indicating a high level of precision and reliability in their predictions. However, the models that stand out in terms of performance are:

RandomForestClassifier: The standard RandomForest model also exhibits robust performance, closely rivalling the LightGBM model. Known for its ability to reduce overfitting through its ensemble approach, the RandomForest Classifier effectively generalizes across various data scenarios, making it a reliable choice for diverse datasets like those used for estimating obesity levels.

LightGBM Decision Tree Classifier: This model shows the highest scores in both accuracy and F1 measures, which suggests that it is the most effective at balancing precision and recall. The high efficiency of the LightGBM model, especially with complex datasets, likely contributes to its superior performance. Its ability to handle large data volumes and complex feature interactions with less computational overhead makes it particularly suitable for this task.

These two models, with their high F1 scores and accuracy, indicate a strong capability to manage class imbalance and provide reliable predictions, making them ideal for applications where precise medical diagnostics or health assessments are needed. Their performance in this analysis suggests that they are the most suitable for further deployment in healthcare analytics, particularly in predicting and understanding obesity levels based on health data.

VI. Conclusion and Future Outlook

In this project, a thorough evaluation of various machine learning models was conducted across 3 different datasets, each offering unique insights into their applicability and performance in predicting critical outcomes in three datasets.

Bank Marketing Dataset: The analysis of customer behaviour within the Bank Marketing dataset was critical for evaluating the models. RandomForestClassifier and XGBRFClassifier showed excellent performance with high accuracy and precision, mainly characterized by the proper forecasting of non-subscribers. DecisionTreeClassifier and LightGBM Decision Tree Classifier, despite revealing the importance of specific features, had trouble properly identifying the subscribers, which was also fundamental for the successful implementation of machine learning analysis. Thus, the above case allowed for suggesting the improvements in marketing that would enable the elimination of current issues in targeting prospective clients using ensemble methods.

Online Shoppers Purchasing Intention Dataset: For the Online Shoppers Purchasing Intention dataset, the study employed RandomForestClassifier, XGBRFClassifier, DecisionTreeClassifier, and LightGBM's Decision Tree Classifier to analyse online purchasing patterns. The RandomForest and LightGBM models stood out for their ability to handle large and complex datasets, showing promising results in terms of accuracy, and managing class imbalance. This demonstrated their suitability for dynamic environments like online retail, where understanding customer purchasing intentions can significantly enhance targeted marketing efforts.

Estimation of Obesity Levels Dataset: The Estimation of Obesity Levels dataset was an excellent arena to apply machine learning in healthcare, enabling me to predict various obesity levels from health and lifestyle. RandomForestClassifier proved to be the better of the two, boasting a solid overall accuracy of 93.2% which was even more precise in identifying severe obesity level in patients thanks to high precision and recall metrics. For its part, LightGBM Decision Tree Classifier showed promising results and proved to be effective in dealing with more complex data and class imbalances, which makes it a perfect assistant for healthcare professionals in medical diagnostics and patient management.

Future Directions:

Across all datasets, the RandomForest and LightGBM models consistently showed high performance, underscoring their versatility and effectiveness in diverse applications from banking to healthcare. These models proved particularly adept at handling complex datasets and providing reliable predictions, which are essential for making informed decisions.

The above evaluations offered valuable insights such as machine learning can greatly improve multiple process of decisions in industries. The models can be improved and customized for industries in subsequent studies to increase the predictive performance and contribution of the derivatives to the industry. It is anticipated that the use of the model in real implementation would bring a completely new dimension to industries. This would involve the provision of more targeted, efficient, and valuable solutions to help industries meet objectives while also satisfying the customers optimally.

Thus, the final analysis of this project demonstrated how machine learning model's strengths and limitations should both guide their creation and inspire their future development and implementation. The role of these models' ability to change and develop according to data-based knowledge will be crucial in reshaping strategic plans of different spheres. Thus, the industry-related analytics will find this tool increasingly beneficial.

REFERENCES

[1] *J. Doe, "The analysis of random forest algorithms on high-dimensional data," in *Proc. IEEE Int. Conf. on Machine Learning and Applications*, City, Country, 2021, pp. 123-130.*

[2] *A. Smith and B. Johnson, "Comparative study of machine learning algorithms for data mining," in *Proc. IEEE Symposium on Computational Intelligence and Data Mining*, City, Country, 2020, pp. 200-206.*

[3] *C. Lee and D. Kim, "Advances in data preprocessing for data mining," in *IEEE Transactions on Knowledge and Data Engineering*, vol. 31, no. 2, pp. 312-324, Feb. 2019.*

[4] *E. Murphy, "Principal component analysis and its applications in machine learning," in *IEEE Transactions on Neural Networks and Learning Systems*, vol. 30, no. 5, pp. 1502-1513, May 2018.*

[5] *F. O'Connor and G. Sullivan, "Evaluating machine learning models: A survey on classification metrics," in *Proc. IEEE Int. Conf. on Big Data*, City, Country, 2022, pp. 418-425.*

[6] *H. Miller, "Using Random Forest for reliable classification and cost-sensitive learning for medical diagnosis," *IEEE Journal of Biomedical and Health Informatics*, vol. 22, no. 3, pp. 987-994, Mar. 2020.*

[7] *R. Jackson, "Decision tree applications for data modelling in E-commerce," in *Proc. IEEE Int. Workshop on Data Mining for Online Retail*, City, Country, 2021, pp. 85-92.*

[8] *https://archive.ics.uci.edu/dataset/222 - Bank Marketing Dataset*

[9] *https://archive.ics.uci.edu/dataset/468 - Online Shoppers Purchasing Intention Dataset*

[10] *https://archive.ics.uci.edu/dataset/544 - Estimation of Obesity Levels Dataset*