Data Glacier

Your Deep Learning Partner

Exploratory Data Analysis

Bank Marketing Campaign

14th April 2023

Data Glacier

Your Deep Learning Partner

Group Name: One

Name: Hira Fahim

Email: hirashahidd26@yahoo.com

Country: United Kingdom

Company: Unemployed

Specialization: Data Science

Batch Code: LISUM19

Submission Date: 14th April 2023

Submitted to: Data Glacier

Background – Bank Marketing Campaign

Problem Description:

ABC Bank wants to sell its term deposit product to customers and before launching the product they want to develop a model which help them in understanding whether a particular customer will buy their product or not (based on customer's past interaction with bank or other Financial Institution).

Approach

The analysis has been divided into six parts:

- Data Understanding
- Exploratory Data Analysis
- Univariate Analysis
- Correlation Analysis
- Bivariate Analysis
- Proposed Model Technique

Data Understanding

Dataset Information

The data is related with direct marketing campaigns of a Portuguese banking institution. The marketing campaigns were based on phone calls. Often, more than one contact to the same client was required, to access if the product (bank term deposit) would be ('yes') or not ('no') subscribed.

Datatype of Columns and Non-null values

```
In [19]: # Datatypes of columns and non-null values
         d.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 45211 entries, 0 to 45210
Data columns (total 17 columns):
 #   Column     Non-Null Count  Dtype
---  ------     --------------  -----
 0   age        45211 non-null  int64
 1   job        45211 non-null  object
 2   marital    45211 non-null  object
 3   education  45211 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    45211 non-null  object
 9   day        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   45211 non-null  object
 16  y          45211 non-null  object
dtypes: int64(7), object(10)
memory usage: 5.9+ MB
```

Numerical and categorical Features

```
Numeric Features:
Index(['age', 'balance', 'day', 'duration', 'campaign', 'pdays', 'previous'], dtype='object')
================================================================================
Categorical Features:
Index(['job', 'marital', 'education', 'default', 'housing', 'loan', 'contact',
       'month', 'poutcome', 'y'],
      dtype='object')
```

Hira Fahim

Exploratory Data Analysis

Step:1 Drop Duplicate Rows

```
# Remove duplicate rows
d=d.drop_duplicates()
d
```

	age	job	marital	education	default	balance	housing	loan	contact	day	month	duration	campaign	pdays	previous	poutcome	y
0	58	management	married	tertiary	no	2143	yes	no	unknown	5	may	261	1	-1	0	unknown	no
1	44	technician	single	secondary	no	29	yes	no	unknown	5	may	151	1	-1	0	unknown	no
2	33	entrepreneur	married	secondary	no	2	yes	yes	unknown	5	may	76	1	-1	0	unknown	no
3	47	blue-collar	married	unknown	no	1506	yes	no	unknown	5	may	92	1	-1	0	unknown	no
4	33	unknown	single	unknown	no	1	no	no	unknown	5	may	198	1	-1	0	unknown	no
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
45206	51	technician	married	tertiary	no	825	no	no	cellular	17	nov	977	3	-1	0	unknown	yes
45207	71	retired	divorced	primary	no	1729	no	no	cellular	17	nov	456	2	-1	0	unknown	yes
45208	72	retired	married	secondary	no	5715	no	no	cellular	17	nov	1127	5	184	3	success	yes
45209	57	blue-collar	married	secondary	no	668	no	no	telephone	17	nov	508	4	-1	0	unknown	no
45210	37	entrepreneur	married	secondary	no	2971	no	no	cellular	17	nov	361	2	188	11	other	no

45211 rows × 17 columns

Step:2 Drop Unnecessary Column

```
# The duration is not known before a call is performed. Also, after the end of the call y is obviously known.
#Thus, this input should be discarded for a realistic predictive model.
d= d.drop(['duration'], axis=1)
```

```
d.shape
```

```
(45211, 16)
```

Step:3 Change Datatype of Categorical features

```
# change datatype of categorical columns into "category"
d["job"]=d["job"].astype("category")
d["marital"]=d["marital"].astype("category")
d["education"]=d["education"].astype("category")
d["default"]=d["default"].astype("category")
d["housing"]=d["housing"].astype("category")
d["loan"]=d["loan"].astype("category")
d["contact"]=d["contact"].astype("category")
d["month"]=d["month"].astype("category")
d["poutcome"]=d["poutcome"].astype("category")
d["y"]=d["y"].astype("category")
```

Final Dataset

```
d.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 45211 entries, 0 to 45210
Data columns (total 16 columns):
 #   Column     Non-Null Count  Dtype
---  ------     --------------  -----
 0   age        45211 non-null  int64
 1   job        45211 non-null  category
 2   marital    45211 non-null  category
 3   education  45211 non-null  category
 4   default    45211 non-null  category
 5   balance    45211 non-null  int64
 6   housing    45211 non-null  category
 7   loan       45211 non-null  category
 8   contact    45211 non-null  category
 9   day        45211 non-null  int64
 10  month      45211 non-null  category
 11  campaign   45211 non-null  int64
 12  pdays      45211 non-null  int64
 13  previous   45211 non-null  int64
 14  poutcome   45211 non-null  category
 15  y          45211 non-null  category
dtypes: category(10), int64(6)
memory usage: 2.8 MB
```

Hira Fahim

Univariate Analysis

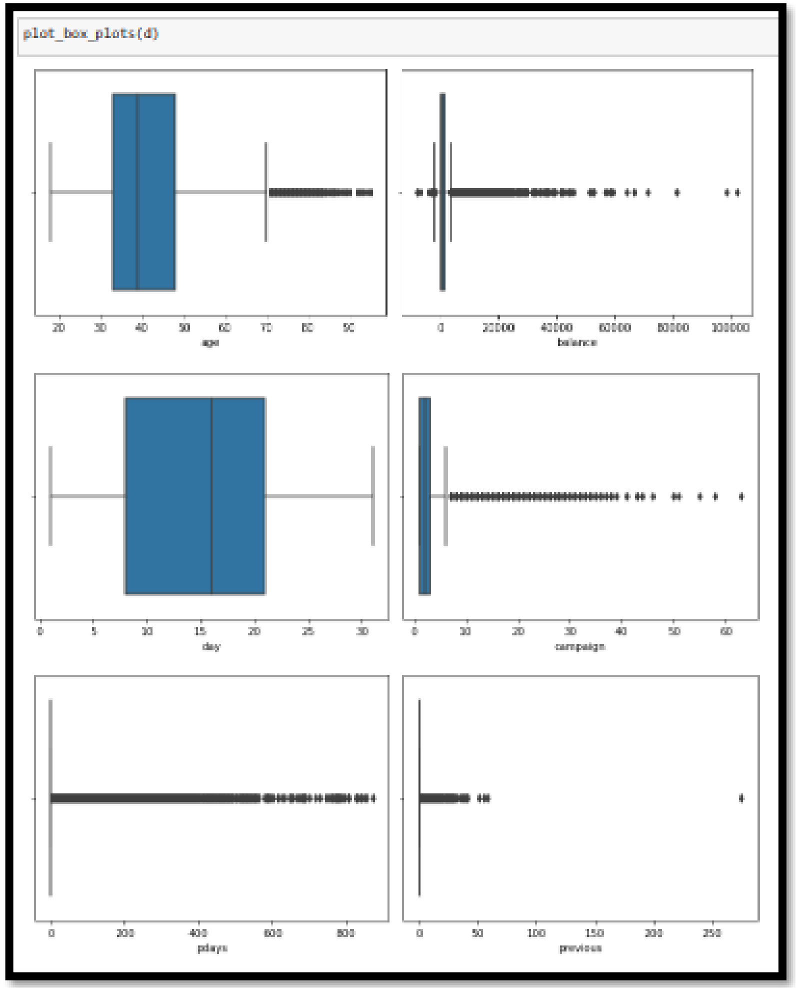
Description of the data

```
# Description of numerical columns
d.describe()
```

	age	balance	day	duration	campaign	pdays	previous
count	45211.000000	45211.000000	45211.000000	45211.000000	45211.000000	45211.000000	45211.000000
mean	40.936210	1362.272058	15.806419	258.163080	2.763841	40.197828	0.580323
std	10.618762	3044.765829	8.322476	257.527812	3.098021	100.128746	2.303441
min	18.000000	-8019.000000	1.000000	0.000000	1.000000	-1.000000	0.000000
25%	33.000000	72.000000	8.000000	103.000000	1.000000	-1.000000	0.000000
50%	39.000000	448.000000	16.000000	180.000000	2.000000	-1.000000	0.000000
75%	48.000000	1428.000000	21.000000	319.000000	3.000000	-1.000000	0.000000
max	95.000000	102127.000000	31.000000	4918.000000	63.000000	871.000000	275.000000

From **description** and **boxplot**, we can see there are outliers in numerical input variables like age, balance, campaign, pdays and previous. Pdays have most outliers comparatively.

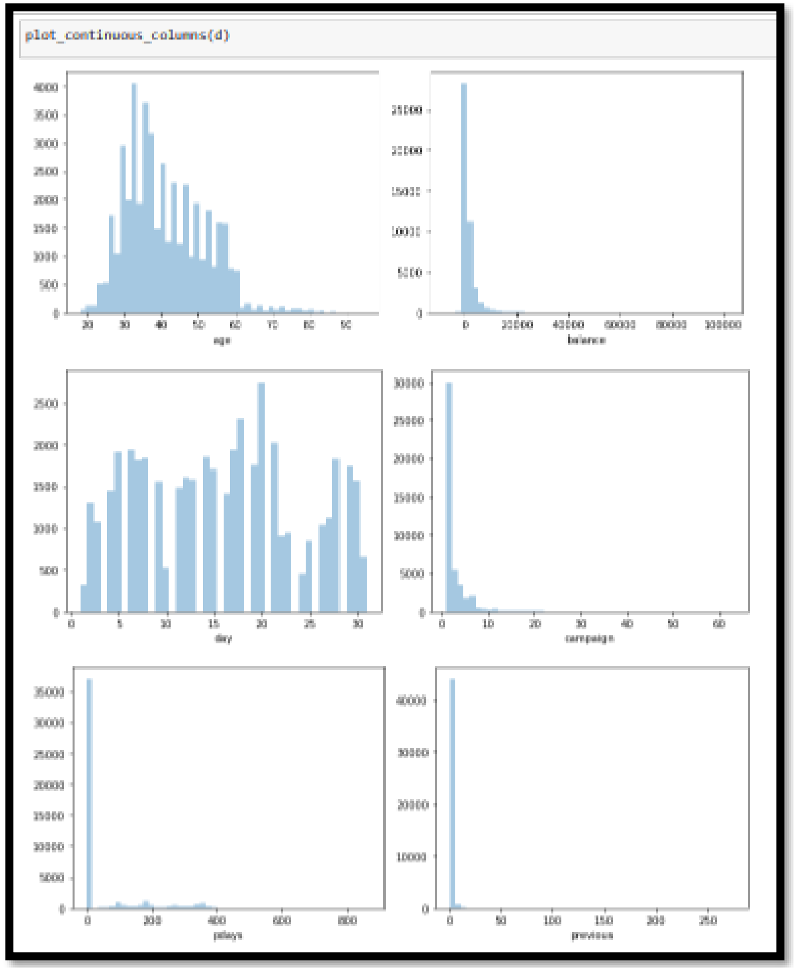
Visualization (boxplot) of Numerical Attributes



Hira Fahim

Univariate Analysis

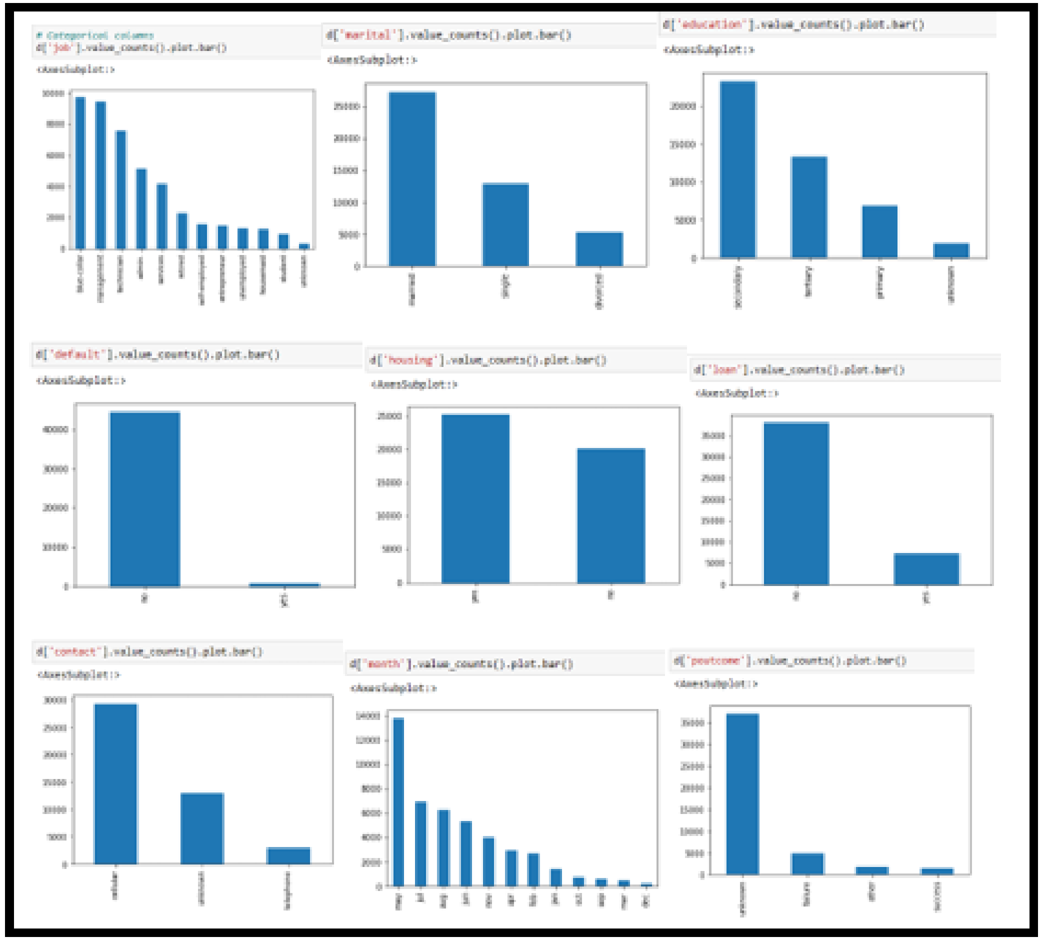
Histogram for Numerical Attributes



In Histogram, we can see input variables like age, balance, campaign, pdays and previous are **positively skewed**, and we can also see uneven distribution of data in day column.

Univariate Analysis

Visualization of Categorical Attributes



In **Bar chart** of categorical columns, we see uneven distribution of data in all the input categorical columns.

Correlation Analysis

Correlation between numerical input and Output variables

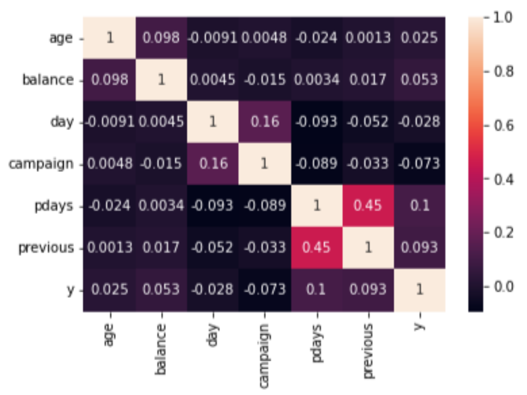
```
d.corr()
```

	age	balance	day	campaign	pdays	previous	y
age	1.000000	0.097783	-0.009120	0.004760	-0.023758	0.001288	0.025155
balance	0.097783	1.000000	0.004503	-0.014578	0.003435	0.016674	0.052838
day	-0.009120	0.004503	1.000000	0.162490	-0.093044	-0.051710	-0.028348
campaign	0.004760	-0.014578	0.162490	1.000000	-0.088628	-0.032855	-0.073172
pdays	-0.023758	0.003435	-0.093044	-0.088628	1.000000	0.454820	0.103621
previous	0.001288	0.016674	-0.051710	-0.032855	0.454820	1.000000	0.093236
y	0.025155	0.052838	-0.028348	-0.073172	0.103621	0.093236	1.000000

Heat Map

```
sns.heatmap(d.corr(),annot=True)
```

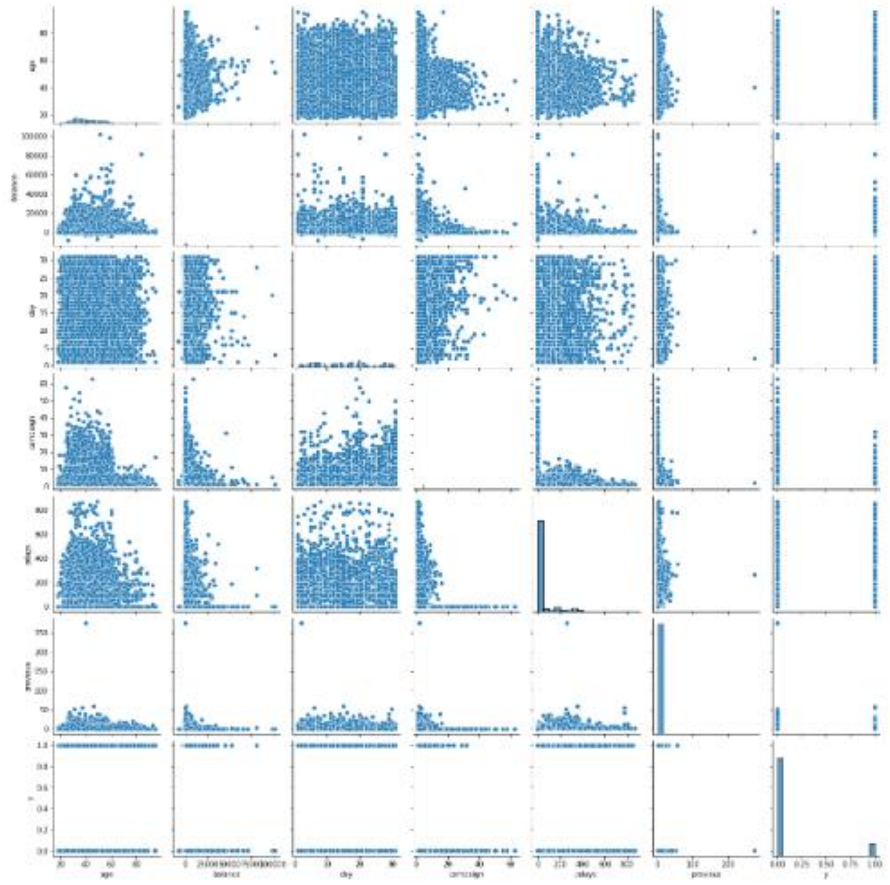
<AxesSubplot:>



Pairplot

```
sns.pairplot(d) # checking the relationship
```

<seaborn.axisgrid.PairGrid at 0x1d2e45934c8>



From Correlation Analysis, we see there is less correlation between numerical attribute and output variable

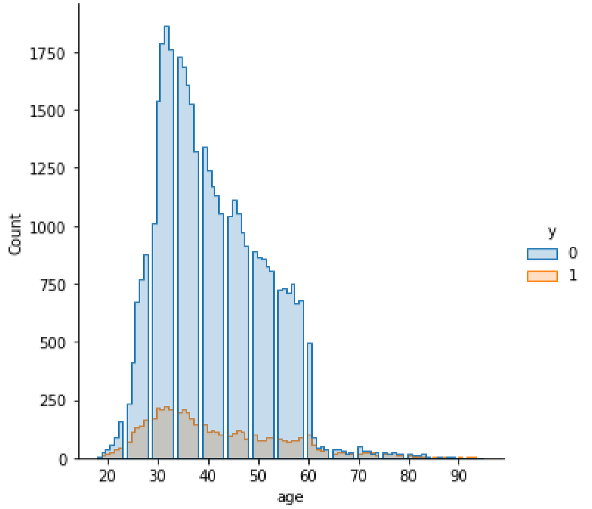
Hira Fahim

Bivariate Analysis

Term Deposit based on Age

```
sns.displot(d, x="age", hue="y", element="step")
```

<seaborn.axisgrid.FacetGrid at 0x1d2f0f0cfd0>

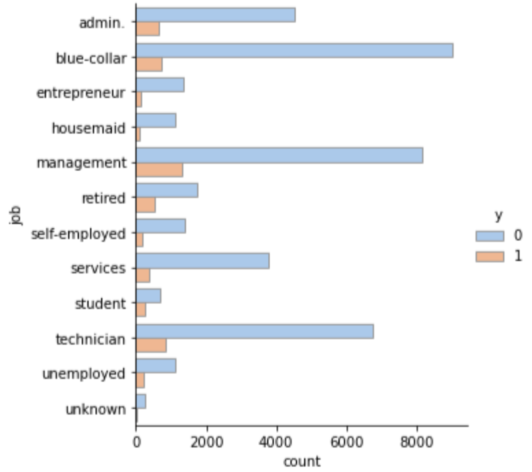


Clients between age 30-40 are more responsive towards term deposit

Term Deposit based on Job

```
sns.catplot(
    data=d, y="job", hue="y", kind="count",
    palette="pastel", edgecolor=".6",)
```

<seaborn.axisgrid.FacetGrid at 0x1d2f108e670>

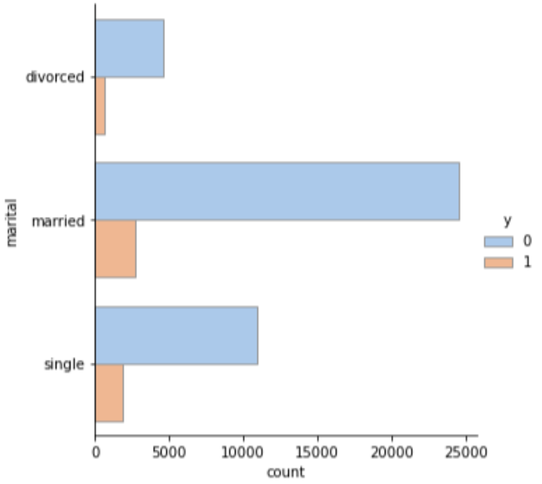


Clients with job related to 'management, blue-collar and technician' have subscribed for deposit.

Term Deposit based on Marital status

```
sns.catplot(
    data=d, y="marital", hue="y", kind="count",
    palette="pastel", edgecolor=".6",)
```

<seaborn.axisgrid.FacetGrid at 0x1d2f0edd9a0>

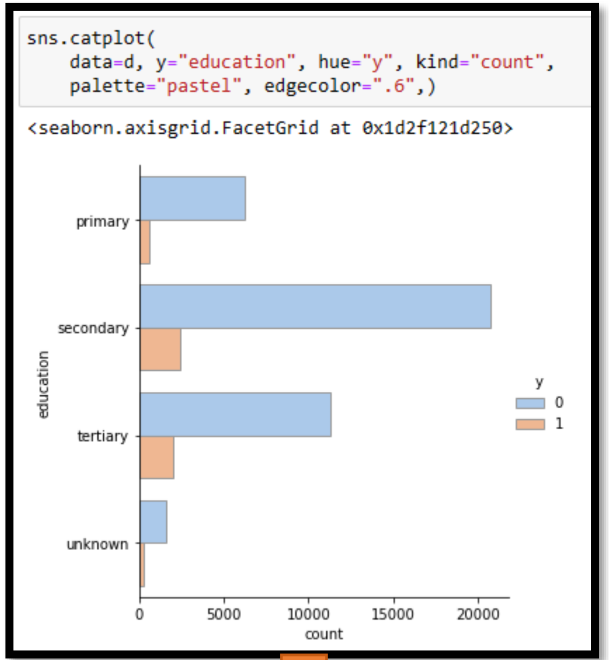


Married Clients are main contributor of deposit scheme.

Hira Fahim

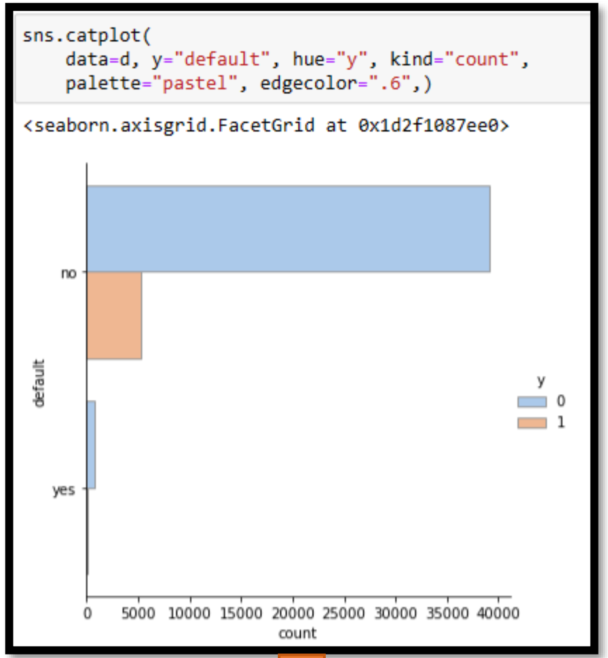
Bivariate Analysis

Term Deposit based on Education



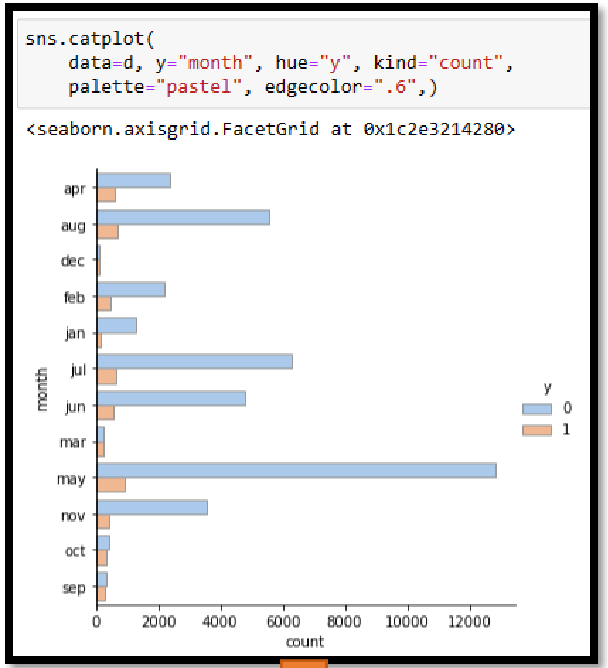
Clients with secondary and tertiary educational background are main contributors.

Term Deposit based on Credit default



Clients with good credit history are more interested in term deposit

Term Deposit based on Month

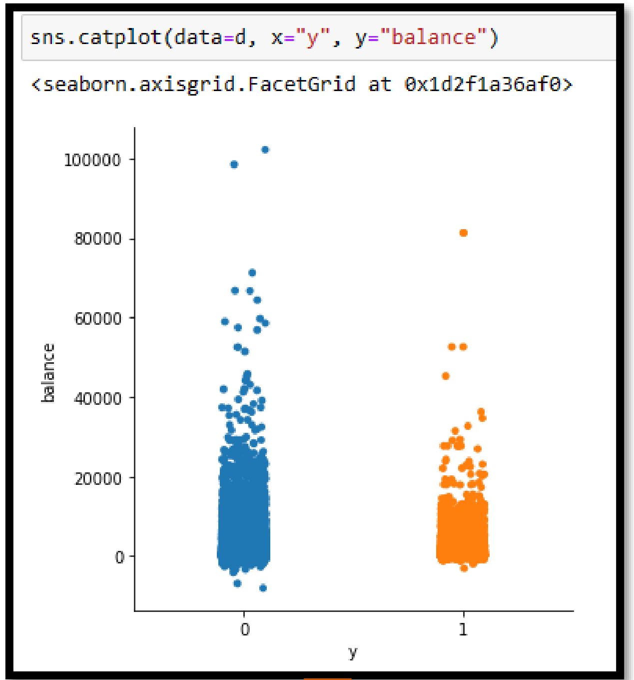


Investment in term deposit is fluctuating throughout the year but in the month of 'May' we have highest success rate.

Hira Fahim

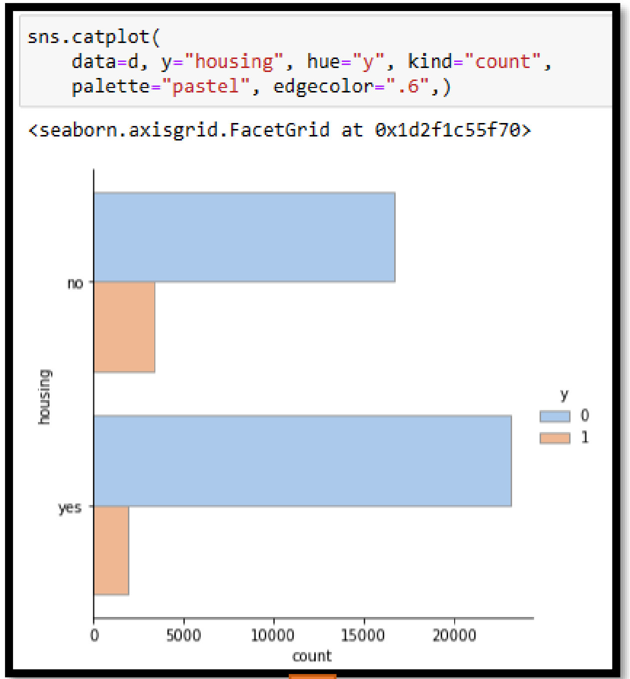
Bivariate Analysis

Term Deposit based on Balance



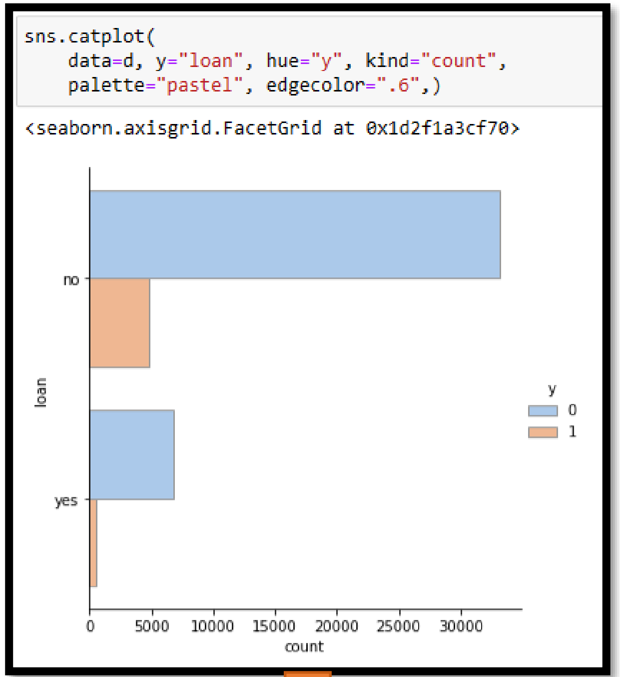
Client with balance up to 20,000 are more interested in term deposit

Term Deposit based on Housing Loan



Client with no housing loan have subscribed for the term deposit

Term Deposit based on Personal Loan



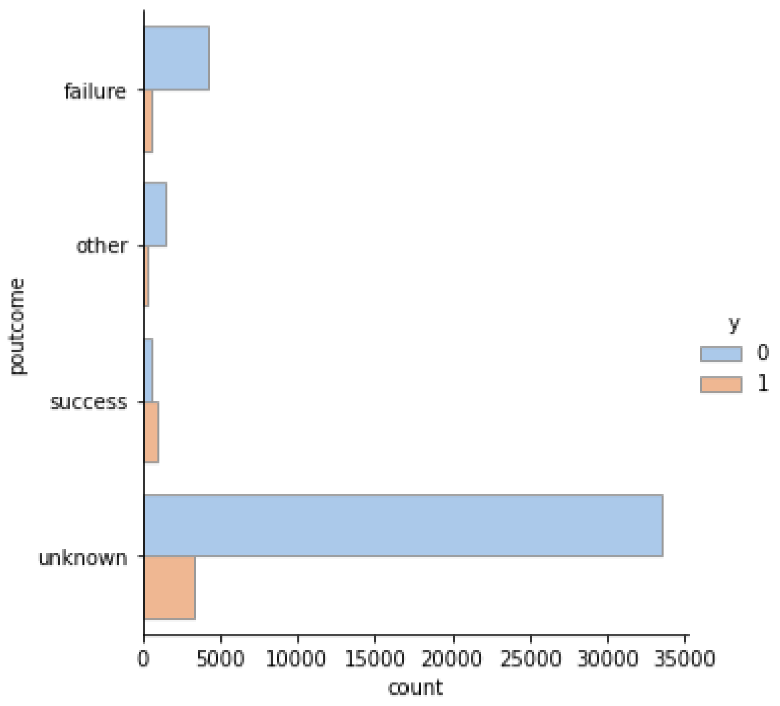
Client with no personal loan are more interested in term deposit.

Bivariate Analysis

Term Deposit based on Outcome of Previous Campaign

```
sns.catplot(
    data=d, y="poutcome", hue="y", kind="count",
    palette="pastel", edgecolor=".6",)
```

```
<seaborn.axisgrid.FacetGrid at 0x1d9864430a0>
```



From the Outcome of previous Campaign, if the outcome is Failure, then there is a less chance that client will subscribe to the term deposit. whereas if the outcome of previous Campaign is Success, then it is more likely that Client will subscribe to the term deposit.

Hira Fahim

Proposed Modelling Technique

In this section, we choose the type of machine learning prediction that is suitable to our problem. We want to determine if this is a regression problem or a classification problem. In this project, we want to predict *weather the clients will subscribe for term deposit or not*. The output variable we want to predict is a discrete value; it can be yes(1) or no (0). This can be seen by looking at the target variable in our dataset “y”:

```
# Number of counts of Target variable
d['y'].value_counts()

no     39922
yes     5289
Name: y, dtype: int64
```

That means that the prediction type that is appropriate to our problem is **classification.**

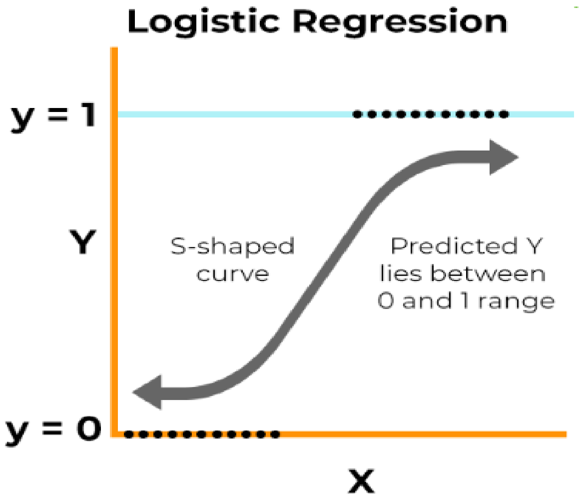
Now we move to choose the modelling techniques we want to use. There are a lot of techniques available for classification problems like Logistic Regression, Decision Tree , Random Forest ,SVC, etc.

Proposed Modelling Technique

In this project, we will test many modelling techniques, and then choose the technique(s) that yield the best results. The techniques that we will try are:

1. Logistic Regression

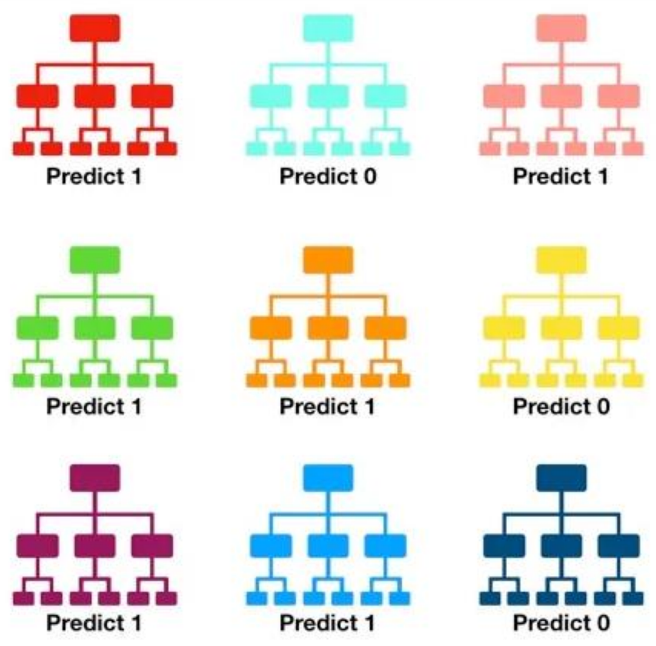
Logistic regression is a statistical method that is used for building machine learning models where the dependent variable is dichotomous: i.e. binary. Logistic regression is used to describe data and the relationship between one dependent variable and one or more independent variables.



Proposed Modelling Technique

2. Random Forest Classifier

Random forest consists of a large number of individual decision trees that operate as an ensemble. Each individual tree in the random forest spits out a class prediction and the class with the most votes becomes our model's prediction. It uses averaging to improve the predictive accuracy and control over-fitting.

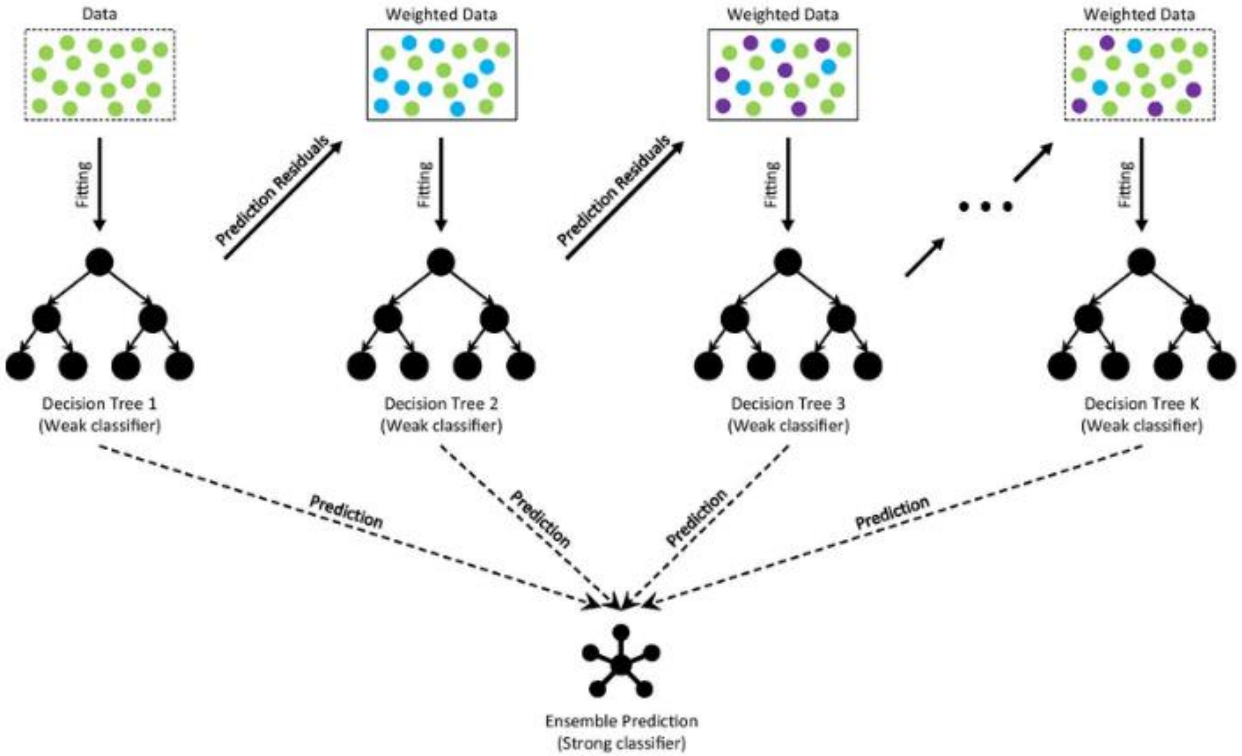


Tally: Six 1s and three 0s
Prediction: 1

Proposed Modelling Technique

3. Gradient Boosting Classifier

Gradient boosting classifiers are a group of machine learning algorithms that combine many weak learning models together to create a strong predictive model. Decision trees are usually used when doing gradient boosting.



Proposed Modelling Technique

Gradient Boosting consists of three essential parts:

Loss Function

The loss function's purpose is to calculate how well the model predicts, given the available data. Depending on the particular issue at hand, this may change.

Weak Learner

A weak learner classifies the data, but it makes a lot of mistakes in doing so. Usually, these are decision trees.

Additive Model

This is how the trees are added incrementally, iteratively, and sequentially. You should be getting closer to your final model with each iteration.

Thank You

Data Glacier
Your Deep Learning Partner

Hira Fahim