Credit Card Lead Prediction Approach Document

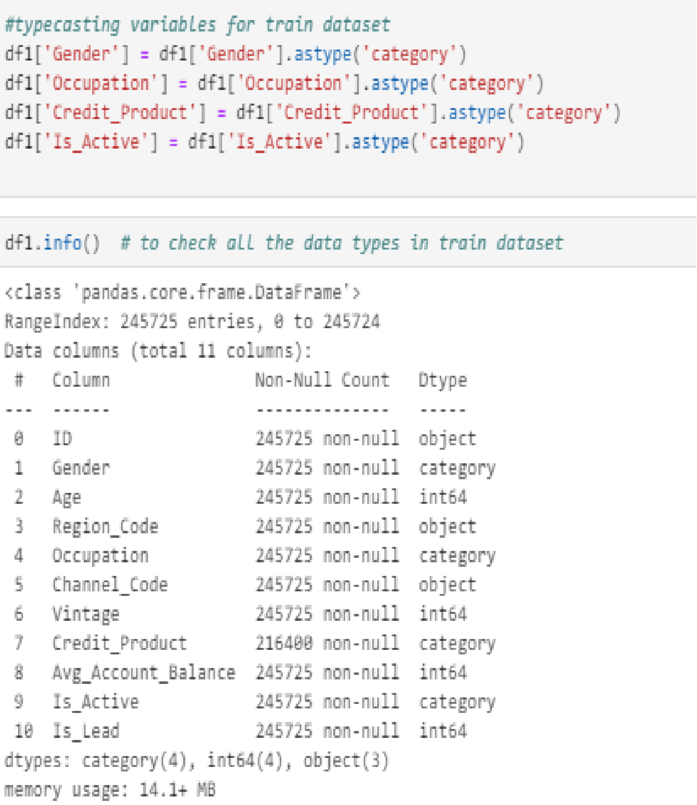
Created by
Shubhasree Sarkar

Contents

Exploratory Data Analysis

Typecasting – *Change of one datatype to another*

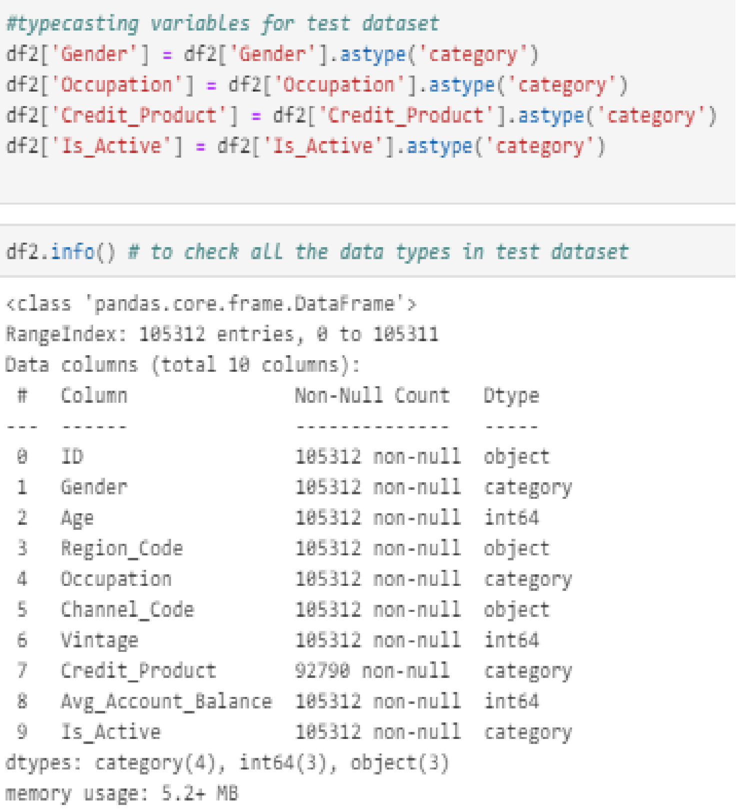
In this step, we have converted the object datatypes to character datatypes in both the train and test dataset.



```
#typecasting variables for train dataset
df1['Gender'] = df1['Gender'].astype('category')
df1['Occupation'] = df1['Occupation'].astype('category')
df1['Credit_Product'] = df1['Credit_Product'].astype('category')
df1['Is_Active'] = df1['Is_Active'].astype('category')
```

```
df1.info()  # to check all the data types in train dataset
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 245725 entries, 0 to 245724
Data columns (total 11 columns):
 #   Column               Non-Null Count   Dtype
---  ------               --------------   -----
 0   ID                   245725 non-null  object
 1   Gender               245725 non-null  category
 2   Age                  245725 non-null  int64
 3   Region_Code          245725 non-null  object
 4   Occupation           245725 non-null  category
 5   Channel_Code         245725 non-null  object
 6   Vintage              245725 non-null  int64
 7   Credit_Product       216400 non-null  category
 8   Avg_Account_Balance  245725 non-null  int64
 9   Is_Active            245725 non-null  category
 10  Is_Lead              245725 non-null  int64
dtypes: category(4), int64(4), object(3)
memory usage: 14.1+ MB
```



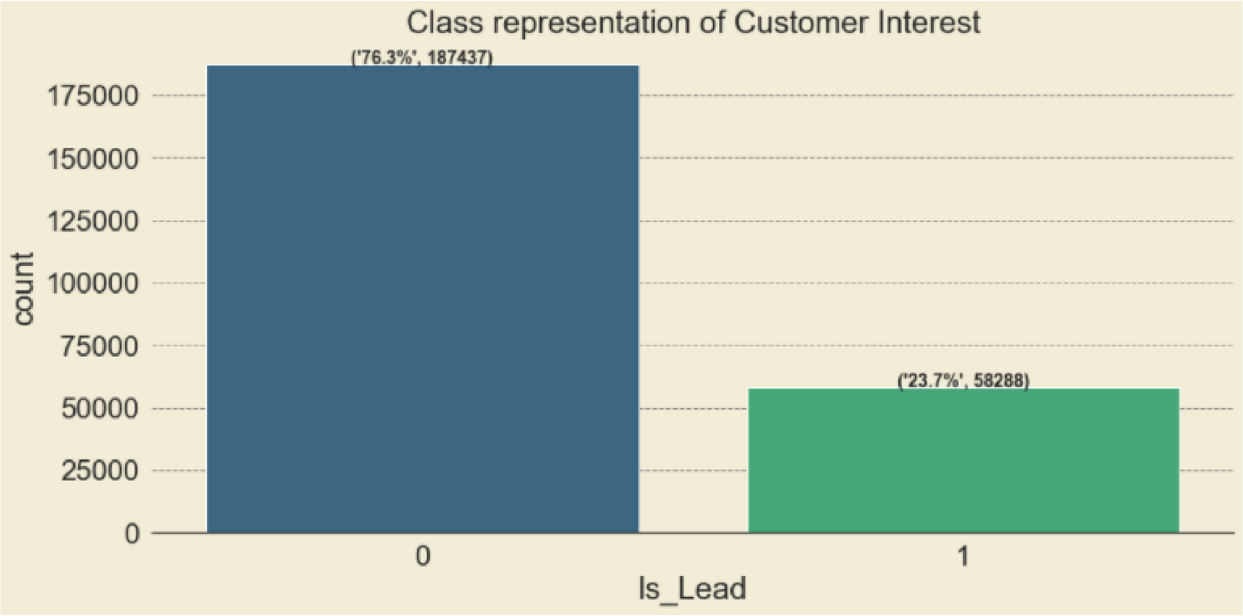
```
#typecasting variables for test dataset
df2['Gender'] = df2['Gender'].astype('category')
df2['Occupation'] = df2['Occupation'].astype('category')
df2['Credit_Product'] = df2['Credit_Product'].astype('category')
df2['Is_Active'] = df2['Is_Active'].astype('category')
```

```
df2.info() # to check all the data types in test dataset
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 105312 entries, 0 to 105311
Data columns (total 10 columns):
 #   Column               Non-Null Count   Dtype
---  ------               --------------   -----
 0   ID                   105312 non-null  object
 1   Gender               105312 non-null  category
 2   Age                  105312 non-null  int64
 3   Region_Code          105312 non-null  object
 4   Occupation           105312 non-null  category
 5   Channel_Code         105312 non-null  object
 6   Vintage              105312 non-null  int64
 7   Credit_Product       92790 non-null   category
 8   Avg_Account_Balance  105312 non-null  int64
 9   Is_Active            105312 non-null  category
dtypes: category(4), int64(3), object(3)
memory usage: 5.2+ MB
```

Class distribution of Target Variable

In this step, we can observe that the target variable can be distributed into two groups based on whether he shows intent towards a recommended credit card-



As we can see from the above image that 76.3% of the customers show no intent, as compared to those showing interest (23.7%), which is why it has led to a rise in class imbalance.

The issue is later solved where we have used SMOTE -oversampling technique to balance the majority and minority class.

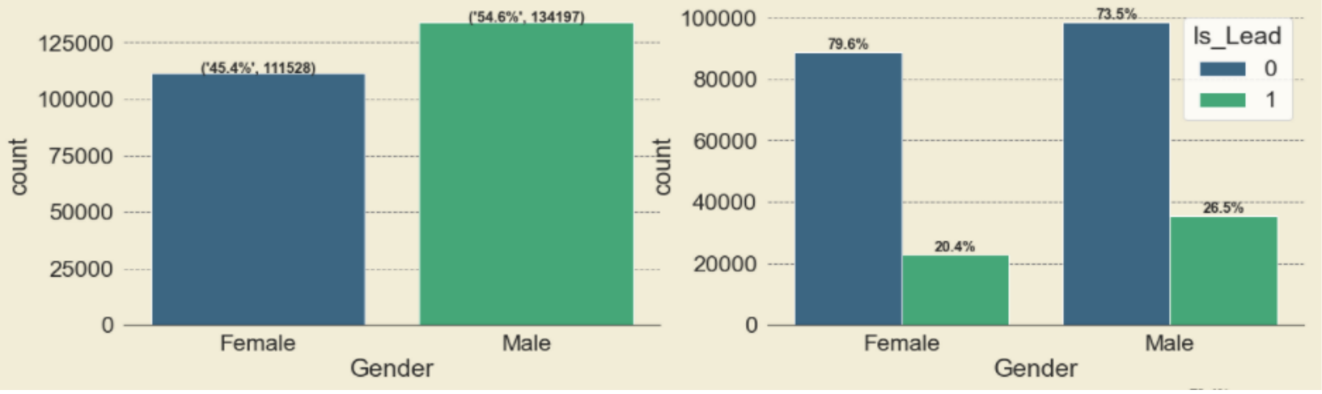
Categorical Values

Class distribution of Independent Variable (Gender)

In this step, we can observe that the Gender attribute can be distributed into two classes – Male (54.6%) and Female (45.4%)

If we consider the Target variable, we can see from the below chart that the less intended customers are more in Male as compared to Female.

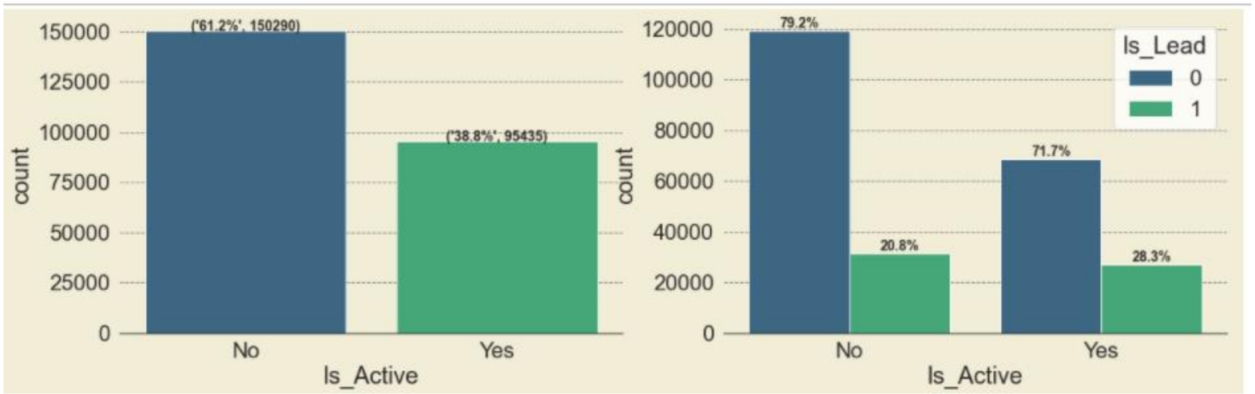
Similarly, for high intended customers, the Male category is more than Female



Class distribution of Independent Variable (Is_Active)

In this step, we can observe that the Active attribute can be distributed into two classes – No (61.2%) and Yes (38.8%)

If we consider the Target variable, we can see from the below chart that the less intended customers are more in No active category as compared to active one.

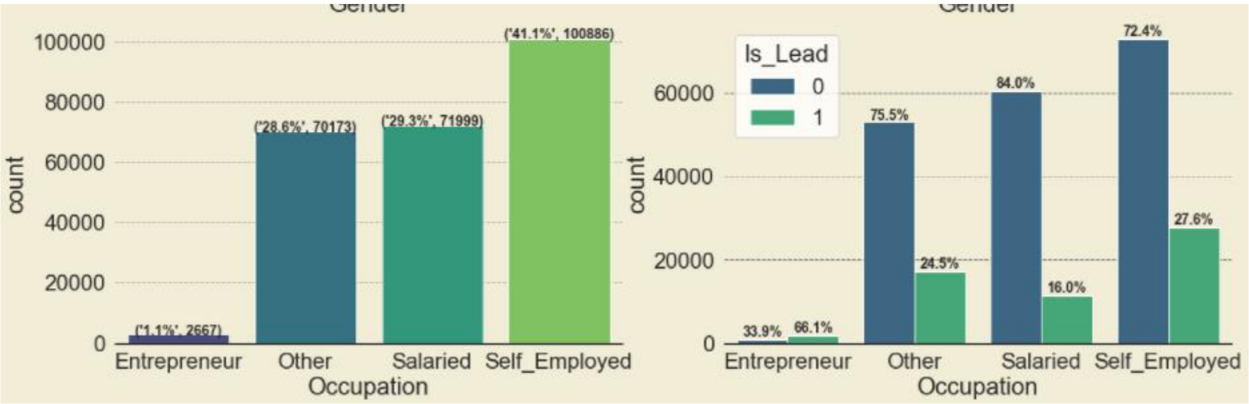


Class distribution of Independent Variable (Occupation)

In this step, we can observe that the Occupation attribute can be distributed into four classes – Self Employed (41.1%), Salaried(29.3%), Other(28.6%) and Entrepreneur(1.1%)

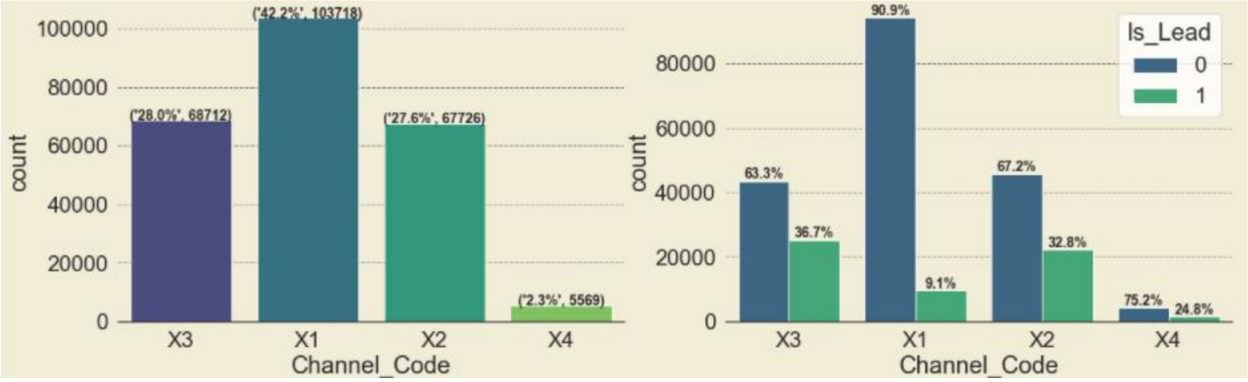
If we consider the Target variable, we can see from the below chart that the less intended customers are more in Self Employed category as compared to intended ones.

While for the Entrepreneurs, the intended customers are higher, even though they are less as compared to other groups.



Class distribution of Independent Variable (Channel)

In this step, we can observe that the Channel attribute can be distributed into four classes – X1 (42.2%), X3 (28.0%), X2 (27.6%) and X4 (2.3%)



If we consider the Target variable, we can see from the below chart that the less intended customers are more in X1 category as compared to intended ones.

While for the X3, the intended customers are higher as compared to the other groups.

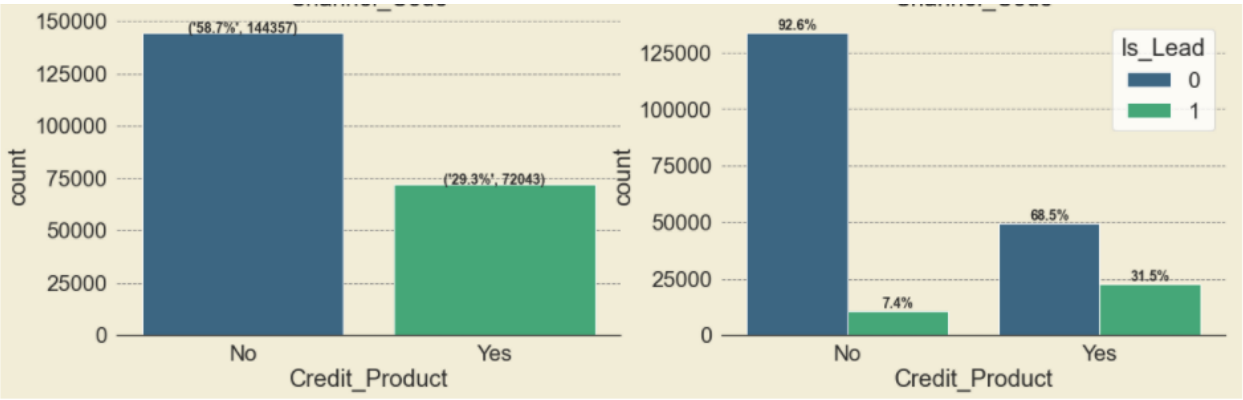
Class distribution of Independent Variable (Credit_Product)

In this step, we can observe that the Credit Product attribute can be distributed into two classes – Yes (29.3%) and No (58.7%)

The rest of the data is missing.

If we consider the Target variable, we can see from the below chart that the less intended customers are more in No credit product category as compared to intended ones.

While for the credit product with Yes, the intended customers are higher as compared to the other .



Numerical Values

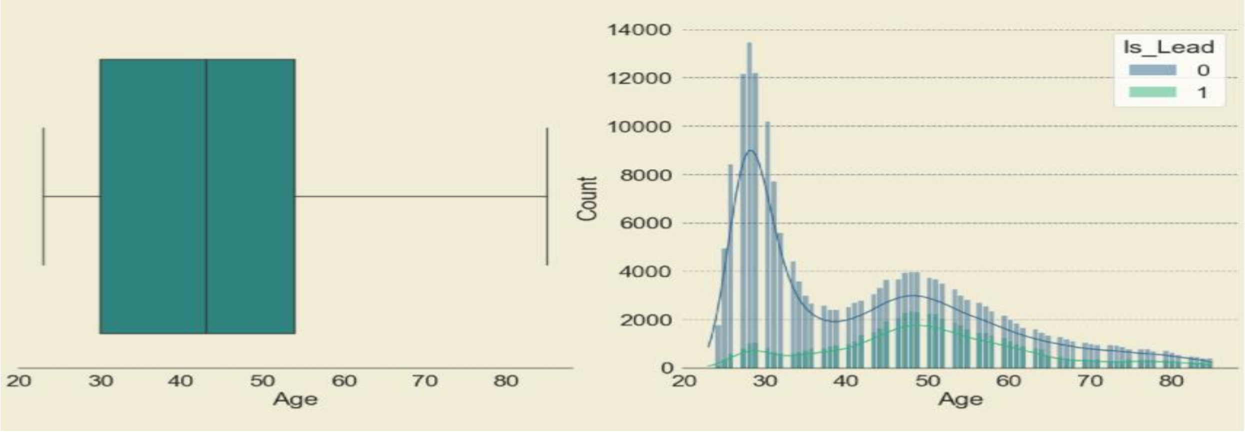
In this step, the distribution of the numerical attributes is observed, and several insights are being drawn from the charts

Description of Independent Variable (Age)

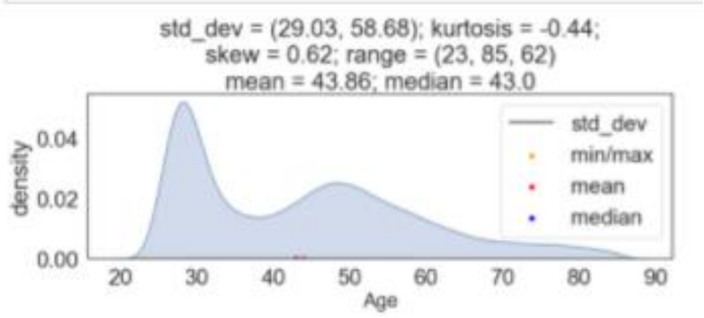
From the train and the test set, we can say that the overall distribution of the Age attribute is similar

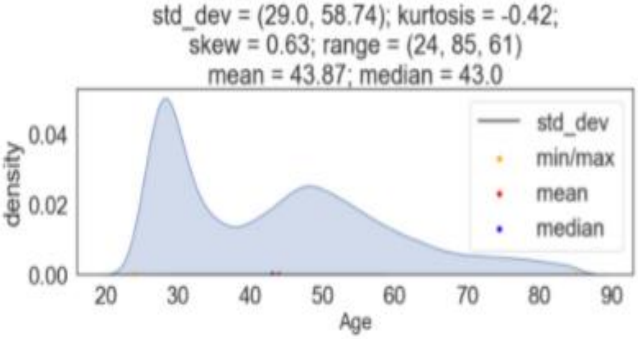
Customers aged between 40-60 are more intended to buy credit cards.

Customers in their 20s and 30s are comparatively less intended



Age feature has a skewness of 0.62 and kurtosis of -0.44 in train set and a skewness of 0.63 and kurtosis of -0.42 in test set



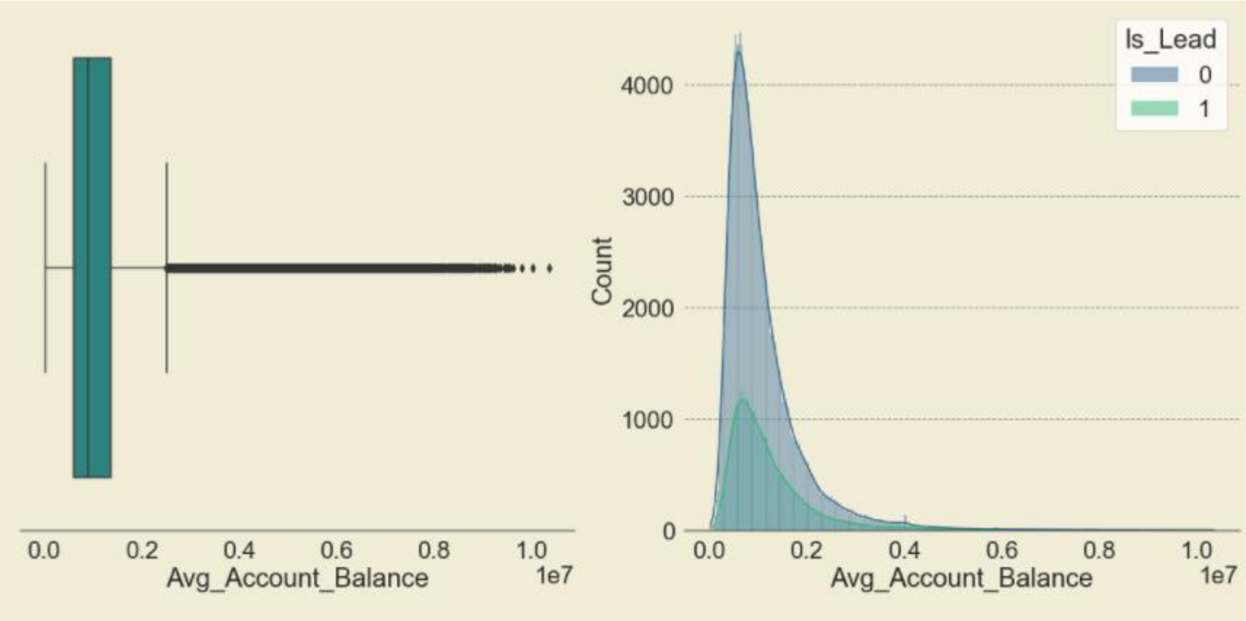


Description of Independent Variable (Average Account balance)

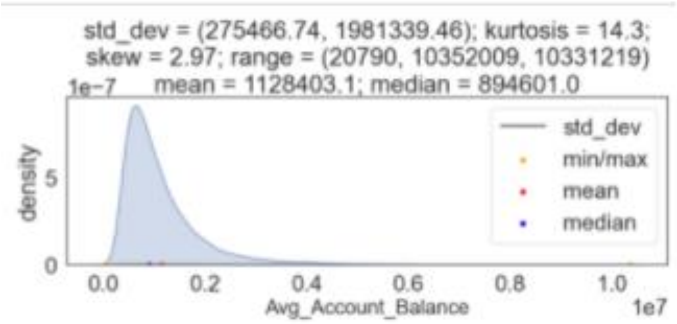
From the train and the test set, we can say that the overall distribution of the Average account balance attribute is similar

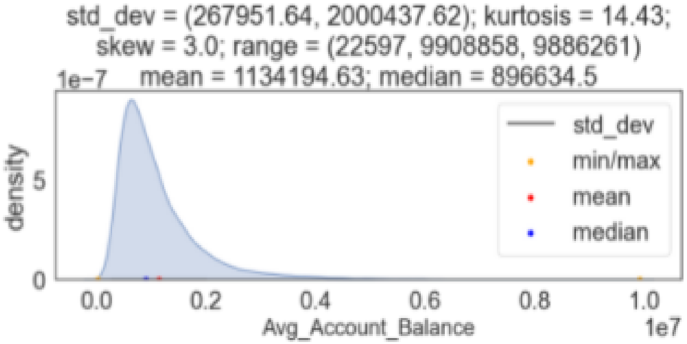
Customers who are less intended to buy credit cards have higher average account balance.

Outliers are present in this category



Average Account Balance feature has a skewness of 2.97 and kurtosis of 14.3 in train set and a skewness of 3.0 and kurtosis of 14.43 in test set



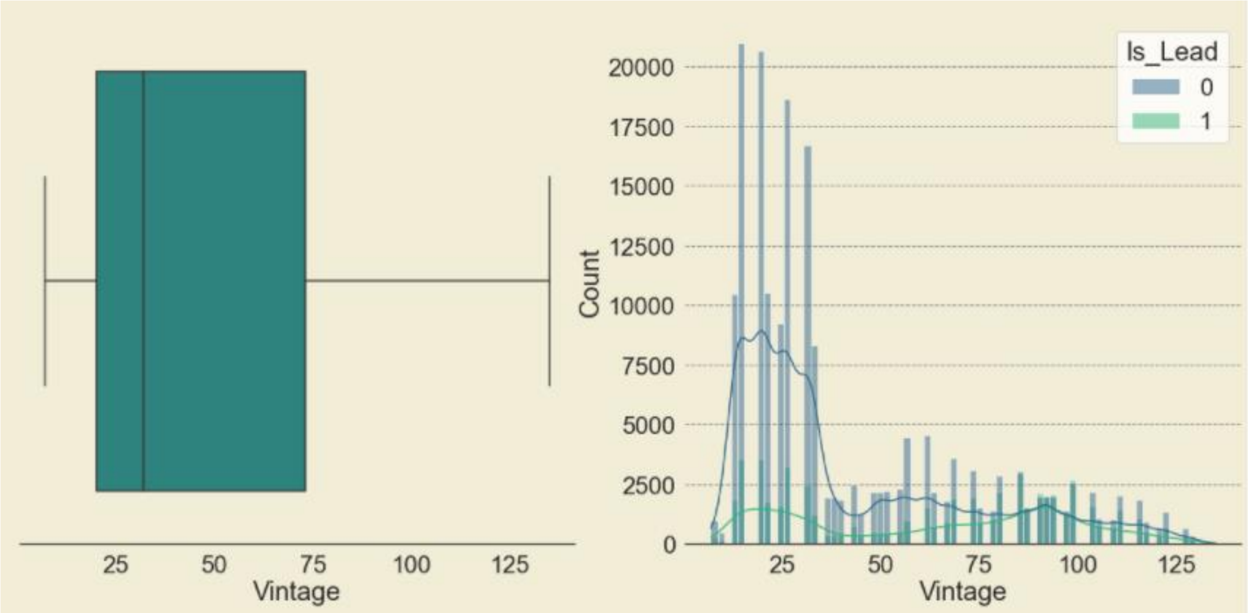


Description of Independent Variable (Vintage)

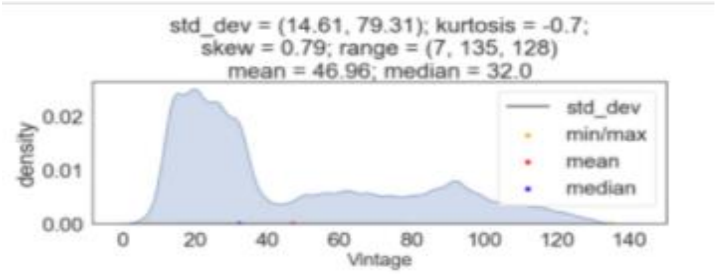
From the train and the test set, we can say that the overall distribution of the Vintage attribute is similar

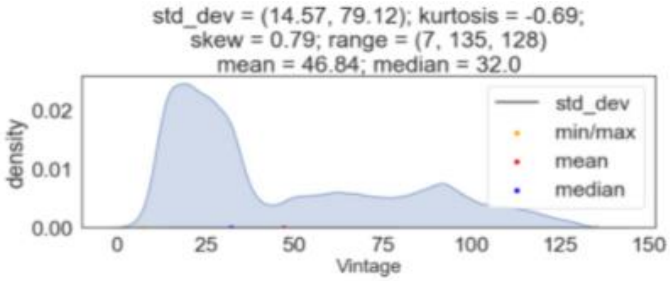
In the customer segment, those who have maintained a longer vintage period (75-100 months) are more intended to take up Credit Cards.

In the category of customers with lower Vintage period (0-40 months) the nos. of customers not intended in taking up Credit Cards is high.



Vintage feature has a skewness of 0.79 and kurtosis of -0.7 in train set and a skewness of 0.79 and kurtosis of -0.69 in test set

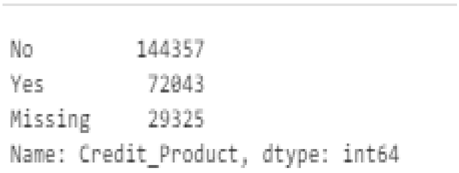




Data Cleaning

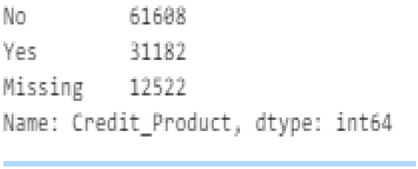
Missing value treatment

In both the train and test datasets, the missing values from the Credit product columns are replaced with a new category named 'Missing'



```
No         144357
Yes         72043
Missing     29325
Name: Credit_Product, dtype: int64
```

Train



```
No         61608
Yes        31182
Missing    12522
Name: Credit_Product, dtype: int64
```

Test

Feature Engineering

For Categorical Variables

Here two approaches are being used in both the train and test datasets-

- Label Encoding
  For the categorical values – Gender, Is_Active and Region_code , Label encoding is being used
- One hot Encoding
  This approach is being used for other categorical features like- Occupation, Channel_Code and Credit_Product.

For Continuous Variables

Here we have used the log transformation method which is a part of feature transformation process

- **Log transformation**

The Avg_Account_Balance attribute has a huge amount of data distribution, which is very high as compared to other independent variables.

For this we chose to apply log transformation on Avg_Account_Balance

We have also used the feature scaling method, ie., using standard scaler on the continuous variables to scale them on a similar range.

- **Standard Scalar**

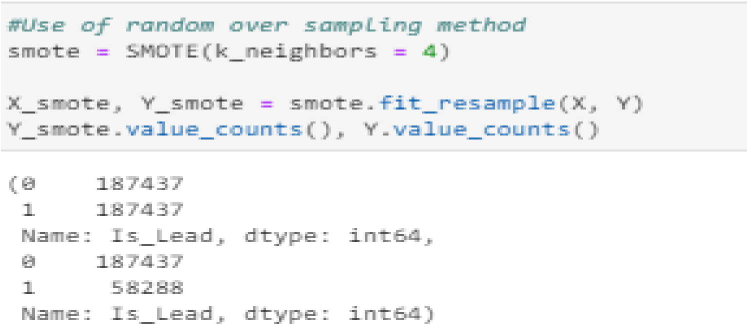
  All the three numerical attributes, Average_Account_Balance, Vintage and Age are being scaled, using the Standard scaler

Oversampling Techniques

As previously mentioned, that the class in Target variable is imbalanced, hence we have opted for the Oversampling technique using SMOTE on the target Column.

As observed, 76.3% customers are not interested in credit card, and about 23.7% are interested in credit card.

In order to solve this issue Oversampling techniques like SMOTE is opted to provide balance in class in the target variable. We have used SMOTE from imblearn module using which the class imbalanced was removed through oversampling technique.



```
#Use of random over sampling method
smote = SMOTE(k_neighbors = 4)

X_smote, Y_smote = smote.fit_resample(X, Y)
Y_smote.value_counts(), Y.value_counts()
```

```
(0    187437
 1    187437
 Name: Is_Lead, dtype: int64,
 0    187437
 1     58288
 Name: Is_Lead, dtype: int64)
```

Model building techniques

For this we have used the Logistic Regression without the doing the Hyper parameter Tuning.

The Value of ROC_AUC_Score is 0.8633