Approach:

1. Loading both training and testing data. Concating both using pd.concat() function.
2. Identifying the type of problem. Its a classification type of problem with Is\_lead as target variable.
3. Exploratory data analysis:
4. Visualization for categorical columns using count plot:

**Categorical(Nominal data types):** Gender, Region\_Code, Occupation, Channel\_Code, Credit\_Product, Is\_Active, Is\_Lead.

**Numerical(Ordinal data type)**: Age,Vintage, Avg\_Account\_Balance.

**Observations from count plot for nominal type of data:**

1.More number of male customer bear credit card compared to female customers.

2.Most of the customers are self-employed and have very less customers who are Entrepreneurs.

3.Maximum number of customers have X1 as channel code and minimum customers are from channel code X4.

4.247812 customers have no credit product. Whereas, 103225 Customers have any active credit product (Home loan, Personal loan, Credit Card etc.)

5.136950 Customers are Active in last 3 Months whereas 214087 customers are inactive.

1. maximum no. of customers are not interested in buying credit card.

**Observations from bivariate analysis:**

1. More no. of male customers are interested for getting credit card compared to females.

2. Customers from channel\_code X2 and X3 are more interested to buy credit card.

3. customers holding Credit\_Product are much interested for credit card.

4. Customers who are entrepreneurs are highly interested to buy credit cards.

1. Data Cleaning:

1)Treating Missing Values:

'Credit\_Product': Using mode to replace null values.

'Is\_Lead' : Using mode to replace null values.

1. Checking for skewness and outliers:

Since, the data has skewness and outliers in categorical type of column. Hence, not removing outliers.

1. Data Preprocessing: Applying ordinalEncodeing is used to convert all datatypes into float type.
2. Splitting training and testing model.
3. Separating Independent and target variable.
4. Train Test Split.
5. from imblearn.over\_sampling importing SMOTE for treating imbalancing of target variable.
6. Finding the best random state.
7. Finding out the best algorithm.
8. Algorithms used:
   1. GaussianNB
   2. LinearDiscriminantAnalysis
   3. RandomForestClassifier
   4. AdaBoostClassifier
   5. GradientBoostingClassifier
   6. ExtraTreesClassifier
   7. SVC
   8. DecisionTreeClassifier
   9. KNeighborsClassifier
   10. SGDClassifier
9. Checking for accuracy\_score, confusion\_matrix, classification\_report, cross\_val\_score, roc\_auc\_score