**FRAUD LOAN PREDICTION DATASET**

**PROBLEM STATEMENT:**

For banks, it is a very big problem that whom they should give the loan. Banks are not able to identify the behaviour of the customer, they can only assume the behaviour of customer about the repayment of loan amount. Banks have all the past data of their loan customer and these data can be used in predicting the behaviour of customers if they will pay or not the loan amount. In recent era of data science banks are using machine learning models for predicting the recovery of loan from its customers.

This data set includes the information about the previous loan customers of the bank. These information are used for the predicting the approval or rejection of the loan. In this dataset 13 columns includes the 614 rows. Following are the variables of this data set:

* **Loan\_id:** Unique loan id of each customer
* **Gender:** Male / Female
* **Dependents:** No of dependents
* **Education:** Applicant education (Graduate / Under-Graduate)
* **Self\_Employes:** Self employed (yes / no)
* **Applicantincome:** Income of applicant
* **CoapplicantIncome:** Income of coapplicant
* **loanAmount:** Amount of loan in thousands
* **Loan\_Amount\_term:** Term of loan amount in months
* **Credit\_History:** Credit history meets the guideline
* **Property\_Area:** Urban / Semiurban / Rural
* **Loan\_Status:** Loan Approved (yes / no)

There are 13 columns in this dataset. The target or y variable in this dataset is Loan\_Status and others are x variable. Y variable is dependent variable and x variable is independent variable which means the value of y is always dependent on the value of x.

This a classification problem as the target value is a categorical variable. Machine learning model of categorical target variable can be prepare by classification process. Loan status is the target variable and it is a categorical variable as it includes only two values Y / N so we use classification process in this dataset.

**Data Analysis:**

First we read the csv file into our jupyter notebook using pd.read\_csv method and check the 5 rows of the data using df.head() function. After checking the data we used the df.info() function for getting the information about the datatype of this dataset. We found that some data are object type some are float type and some are integer type. Then we check the unique values in the dataset by using df.unique() function.

* In Loan Id column we can see that each person has unique loan id
* In gender column there is two values one for Male and second for Female
* In married column Y represents that the applicant is married and N represents that the applicant is not married
* Dependent column shows the no or person dependent on the applicant. In this dataset there is maximum 3 person dependent on the applicant and the minimum is 0 person.
* Education column shows whether the applicant is Graduate or Non-Graduate. According to assumption of this dataset the graduate person gets the higher chances for approving the loan.
* Self Employed column shows whether the applicant is self employed or employee.
* Applicant Income shows the income of the applicant. Higher income increases the chances of approval the loan
* Coapplicant Income shows the income of coapplinant. Higher income of coapplicant increases the chances of loan approval.
* Loan Amount shows the amount of loan demanded by the applicant. If applicant demand for a big amount then there are chances of rejection of loan application.
* Loan Amount term shows the no month in which the applicant pays the total amount of loan with interest.
* Credit History shows the past credit history of the applicant. In this dataset 1 stands for good credit history and 0 stands for bad credit history.
* Property Area shows the area of property it can be urban, semiurban or rural.
* Loan status shows the approval and rejection of the loan. In this dataset Y stands for approval of the loan and N stands for rejection of the loan.

**EDA:**

EDA is known as the exploratory data analysis. In this analysis it is shown that which variable has the most impact on the target variable and how a column is performing in predicting the target variable. In this dataset the maximum income of applicant is 8100 and the mean of the income is 5403. Maximum loan amount requested by the applicant is 700 thousand and minimum is 9 thousand. Maximum no of month for repayment of loan id 480 months and minimum is 12 months.

Maximum term is used in this dataset in 360 month which means most of the loan are requested for 360 months.

In this dataset we can find that no of graduate applicant is higher than non-graduate applicant.

Most of the applicant has 0 dependents. Less no of dependents increases the chances of loan approval.

Self Employed applicant person are less educated in this dataset.

Loan Status is directly dependent on credit history of the applicant if the credit history of applicant is 1 then there is high chances of loan approval.

Applicant income is also has positive effect on loan status if income of applicant Is high then the chances of loan approval will also high.

If the amount of loan is less then there is more chances for loan approval and high amount of loan has less chances for approval.

Here we check the correlation of the variables. It shows the relation between the variables. We check the correlation by using df.corr() function. In this dataset credit history affect the loan status very much.

We also check the null values by using df.isnull().sum() function and do proper treatment for null values. We will use df.fillna() function for filling the null values. This code suggests that there are 13 missing values in Gender, 3 in Married, 15 in Dependents, 32 in Self\_Employed, 22 in Loan Amount, 14 in Loan\_Amount\_Term and 50 in Credit History. In this dataset null values are filled by the mode and mean values of particular null column and make null values 0 for good model prediction. Mode is used for the Gender, Married, Dependents and Credit History as these are categorical variables. Mean is used for Loan Amount and Loan Amount Term.

**Pre – Processing Pipeline:**

After getting the proper knowledge of dataset now we forward to the data cleaning and data preprocessing.

First we drop the Loan Id column as it a unique id for every applicant, it is not playing a important role in predicting the Loan status. We drop the loan id column by using df.drop() function.

Now we have to deal with the categorical variable of the dataset. For predicting the target variable using machine learning model data should be in numeric form. We use the LabelEncoder for converting categorical variable into numeric form. We can import labelencoder from siktlearn and can be use as df(‘col’)=labelencoder.fit\_transform(df(‘col’)). In this dataset Gender, Married, Dependents, Education, self\_Employed and Loan\_Status are categorical variables. We convert them into numeric form by using labelencoder. Now Loan approval is converted in 1 and rejection is converted in 0.

There are some outliers in this dataset. These outliers can be removed by using zscore. Zscore can be import from the scipy.stats. The threshold value is 3 in zscore. We have applied the zscore function and create a new dataframe as df\_new. Now some values are removed from the dataset and the df\_new has the 577 rows in it.

High skewness is also a problem for a good model. We can check the skewness of DataFrame by using df.skew() function. For predicting a good model the skewness of a column is less then 0.55. There are many methods for removing the skewness like log method, square root method, boxcox method. In this dataset we used the log method for removing the skewness.

Now we will separate the data into x and y variable. X variable includes all the independent variable of the dataset and y variable includes only dependent variable that is LoanStatus in this dataset.

We use the standardscaler function for making the independent value scaler. StandardScaler can be imported from sklearn.preprocessing. we use standardscaler by using this code :- x=StandardScaler.fit\_transform(x).

**Building Machine Learning Model:**

We are using the classification model for this dataset as target variable of this dataset is a categorical variable.First we import all the necessary library for the machine learning model from the sikitlearn. After importing all the libraries we first split the data into train test split.

We will create a function for getting the maximum accuracy score from the provided classification model. We can easily get the maximum accuracy score at random state from 42 to 101. The following code we have to pass for getting the max accuracy score:-

def max\_acc\_score(clf,x,y):

max\_score=0

for r in range(42,101):

x\_train,x\_test,y\_train,y\_test=train\_test\_split(x,y,test\_size=.20,random\_state=r)

clf.fit(x\_train,y\_train)

pred=clf.predict(x\_test)

a\_sc=accuracy\_score(y\_test,pred)

if a\_sc>max\_score:

max\_score=a\_sc

a\_state=r

print('max score at random\_state of',a\_state,'is',max\_score)

return a\_state

**Logistic Regression:**

Logistic regression is a classification model of machine learning for preparing a machine learning model of a classification problem. First we import the logistic regression from sikitlean and create a variable lg=LogisticRegression(). We pass this variable into the max\_acc\_score and we get the maximum accuracy score at suitable random state. We get the 89.65 accuracy score at the random state of 91. We will use this random state for the train test split. After splitting the data into xtrain, ytrain, xtest and ytest we will fit the the xtrain and ytrain into the model predict the model by using xtest. After predicting the model we will check the accuracy score of the model, classification report of the model and confusion matrix of the model with help of ytest and prediction and also check the cross validation score of the model for checking the over fitting and under fitting of the model. The mean cross validation score of this model is 81.80. Here we can see that there in no overfitting and underfitting in this model. Before saving the model we also check the r2\_score of the model. The r2\_score of this model is 78.57.

**Decision Tree Classifier:**

DecisionTreeClassifier is a classification model of machine learning for preparing a machine learning model of a classification problem. First we import the DecisionTreeClassifier from sikitlean.tree and create a variable dt=DecisionTreeClassifier(). We pass this variable into the max\_acc\_score and we get the maximum accuracy score at suitable random state. We get the 77.85 accuracy score at the random state of 50. We will use this random state for the train test split. After splitting the data into xtrain, ytrain, xtest and ytest we will fit the the xtrain and ytrain into the model predict the model by using xtest. After predicting the model we will check the accuracy score of the model, classification report of the model and confusion matrix of the model with help of ytest and prediction and also check the cross validation score of the model for checking the over fitting and under fitting of the model. The mean cross validation score of this model is 71.40. Here we can see that there is no overfitting and underfitting in this model. Before saving the model we also check the r2\_score of the model. The r2\_score of this model is 74.93.

**KNeighborsClassifier:**

KNeighborsClassifier is a classification model of machine learning for preparing a machine learning model of a classification problem. First we import the KNeighborsClassifier from sikitlean.neighbors and create a variable knn= KNeighborsClassifier (). We pass this variable into the max\_acc\_score and we get the maximum accuracy score at suitable random state. We get the 87.06 accuracy score at the random state of 91. We will use this random state for the train test split. After splitting the data into xtrain, ytrain, xtest and ytest we will fit the the xtrain and ytrain into the model predict the model by using xtest. After predicting the model we will check the accuracy score of the model, classification report of the model and confusion matrix of the model with help of ytest and prediction and also check the cross validation score of the model for checking the over fitting and under fitting of the model. The mean cross validation score of this model is 81.11. Here we can see that there is no overfitting and underfitting in this model. Before saving the model we also check the r2\_score of the model. The r2\_score of this model is 73.21.

**AdaBoostClassifier:**

AdaBoostClassifier is a classification model of machine learning for preparing a machine learning model of a classification problem. First we import the AdaBoostClassifier from sikitlean.ensemble and create a variable ada= AdaBoostClassifier (). We pass this variable into the max\_acc\_score and we get the maximum accuracy score at suitable random state. We get the 89.65 accuracy score at the random state of 91. We will use this random state for the train test split. After splitting the data into xtrain, ytrain, xtest and ytest we will fit the the xtrain and ytrain into the model predict the model by using xtest. After predicting the model we will check the accuracy score of the model, classification report of the model and confusion matrix of the model with help of ytest and prediction and also check the cross validation score of the model for checking the over fitting and under fitting of the model. The mean cross validation score of this model is 81.80. Here we can see that there is no overfitting and underfitting in this model. Before saving the model we also check the r2\_score of the model. The r2\_score of this model is 78.57.

**Model Selection:**

After performing all the classification model we have to select the model for the dataset. In our dataset we can see that adaboostclassifier is performing very well and giving the output very close so we select the adaboost classifier for this dataset. We can save this model by using sklearn.joblib and use this model for future use in predicting the Loanstatus of the customer.

**Concluding Remarks:**

In simple words we can say that the dataset we are using for this model is a banking dataset. We prepare this model for checking the loan status of a customer weather he will pay the loan or not. This model is a classification model as its target variable is a categorical variable. By giving the necessary information into this model a bank can easily get the loan status of the applicant.