PPML MINIPROJECT REPORT

LOAN APPROVAL PREDICTION USING MACHINE LEARNING

-AMIRTA VARSHINI R

220901005

III-EEE-A

**Aim**

To analyze a dataset on loan approval prediction, preprocess the data, and evaluate the performance of various machine learning classifiers (K-Nearest Neighbors, Random Forest, Support Vector Machine, and Logistic Regression) for predicting loan approval status.

**Abstract**

Loan approval prediction is crucial for financial institutions to minimize risks and streamline decision-making processes. This project involves the exploration and preprocessing of a dataset containing loan application details. The data is analyzed to understand the distribution of categorical variables and correlations between features. Missing values are handled using imputation, and categorical variables are encoded. The dataset is then split into training and testing sets to evaluate the performance of four machine learning models. Performance metrics, including accuracy scores, are used to compare the classifiers.

**Program Code:**

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

import seaborn as sns

data = pd.read\_csv("LoanApprovalPrediction.csv")

data.head(5)

obj = (data.dtypes == 'object')

print("Categorical variables:",len(list(obj[obj].index)))

# Dropping Loan\_ID column

data.drop(['Loan\_ID'],axis=1,inplace=True)

obj = (data.dtypes == 'object')

object\_cols = list(obj[obj].index)

plt.figure(figsize=(18,36))

index = 1

for col in object\_cols:

y = data[col].value\_counts()

plt.subplot(11,4,index)

plt.xticks(rotation=90)

sns.barplot(x=list(y.index), y=y)

index +=1

# Import label encoder

from sklearn import preprocessing

# label\_encoder object knows how

# to understand word labels.

label\_encoder = preprocessing.LabelEncoder()

obj = (data.dtypes == 'object')

for col in list(obj[obj].index):

data[col] = label\_encoder.fit\_transform(data[col])

# To find the number of columns with

# datatype==object

obj = (data.dtypes == 'object')

print("Categorical variables:",len(list(obj[obj].index)))

plt.figure(figsize=(12,6))

sns.heatmap(data.corr(),cmap='BrBG',fmt='.2f', linewidths=2,annot=True)

sns.catplot(x="Gender", y="Married", hue="Loan\_Status", kind="bar", data=data)

for col in data.columns:

data[col] = data[col].fillna(data[col].mean())

data.isna().sum()

from sklearn.model\_selection import train\_test\_split

X = data.drop(['Loan\_Status'],axis=1)

Y = data['Loan\_Status']

X.shape,Y.shape

X\_train, X\_test, Y\_train, Y\_test = train\_test\_split(X, Y, test\_size=0.4, random\_state=1)

X\_train.shape, X\_test.shape, Y\_train.shape, Y\_test.shape

from sklearn.neighbors import KNeighborsClassifier

from sklearn.ensemble import RandomForestClassifier

from sklearn.svm import SVC

from sklearn.linear\_model import LogisticRegression

from sklearn import metrics

knn = KNeighborsClassifier(n\_neighbors=3)

rfc = RandomForestClassifier(n\_estimators = 7,

criterion = 'entropy', random\_state =7)

svc = SVC()

lc = LogisticRegression()

# making predictions on the training set

for clf in (rfc, knn, svc,lc):

clf.fit(X\_train, Y\_train)

Y\_pred = clf.predict(X\_train)

print("Accuracy score of ", clf.\_\_class\_\_.\_\_name\_\_, "=",100\*metrics.accuracy\_score(Y\_train, Y\_pred))

# making predictions on the testing set

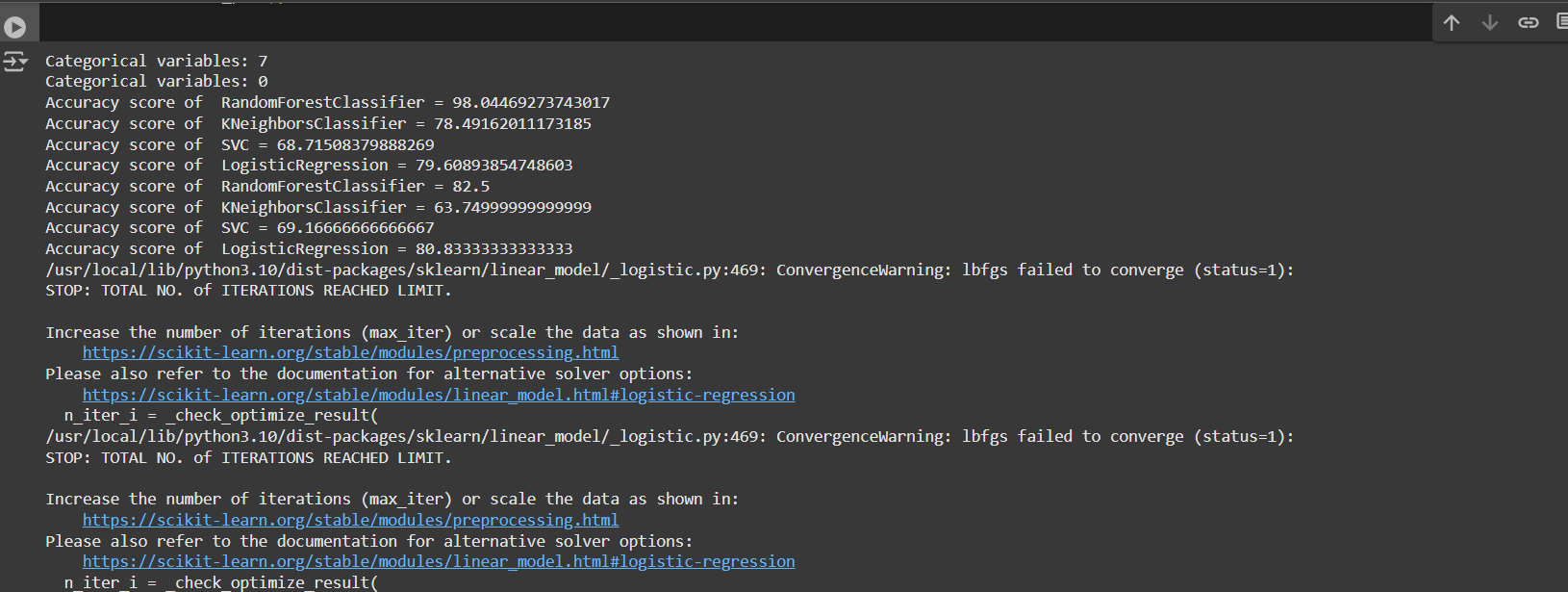
for clf in (rfc, knn, svc,lc):

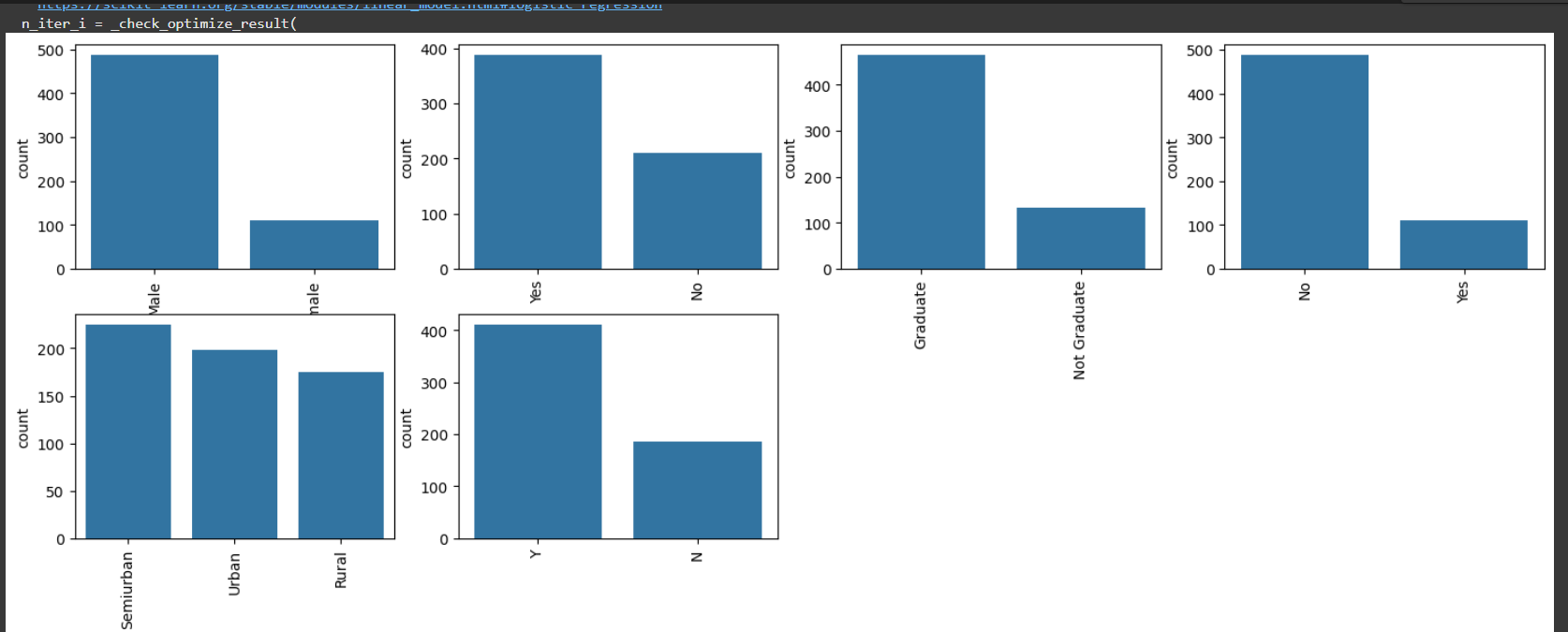
clf.fit(X\_train, Y\_train)

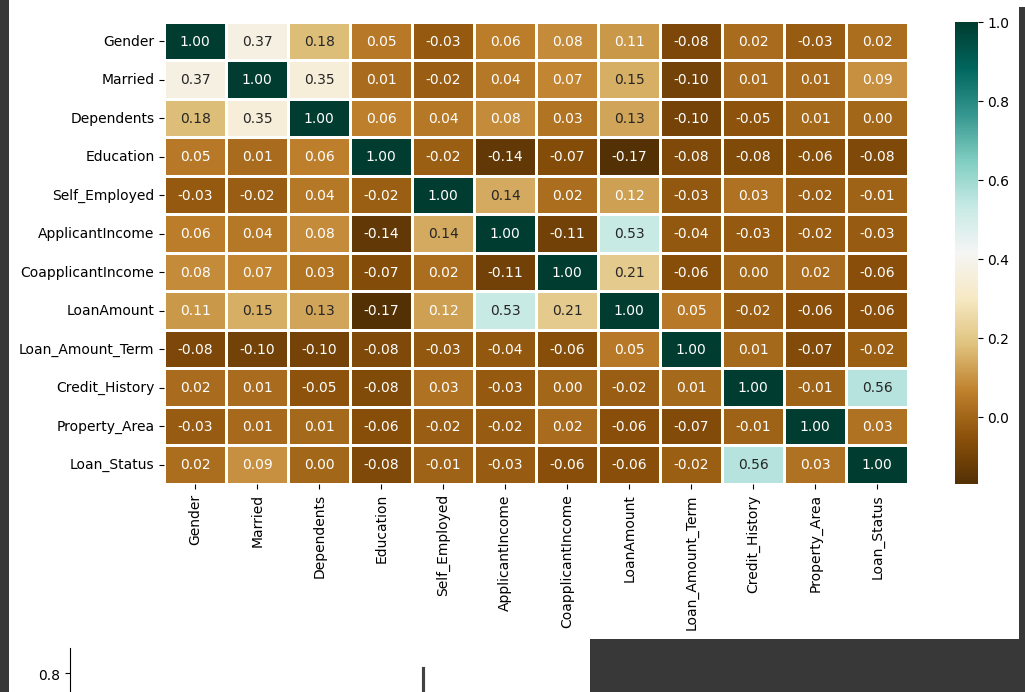
Y\_pred = clf.predict(X\_test)

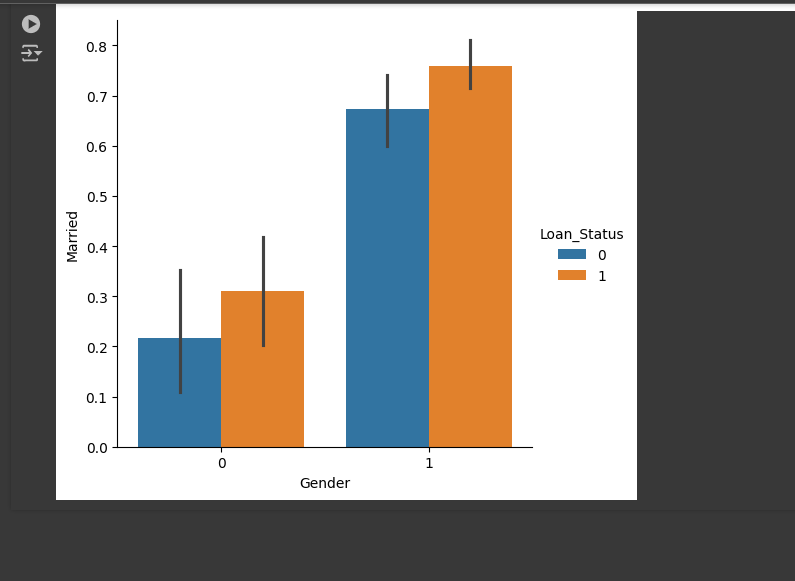
print("Accuracy score of ", clf.\_\_class\_\_.\_\_name\_\_,"=", 100\*metrics.accuracy\_score(Y\_test, Y\_pred))

**OUTPUT:**

****

****

****

****

**Colab link:**

[**https://colab.research.google.com/drive/1ht90qvpQtaHWPP4gVcAvOmbWpGSUK2Sf?usp=sharing**](https://colab.research.google.com/drive/1ht90qvpQtaHWPP4gVcAvOmbWpGSUK2Sf?usp=sharing)

**Result:**

Thus the loan approval prediction is done using machine learning algorithms.