```
from sklearn.linear_model import LogisticRegression
from sklearn.naive_bayes import GaussianNB
from sklearn.ensemble import RandomForestClassifier, VotingClassifier
from sklearn.tree import DecisionTreeClassifier
import pandas as pd

df=pd.read_csv("./dataset/train_clean_data.csv",index_col=0)
df.head()
```

```
    Loan_ID Dependents  ApplicantIncome  CoapplicantIncome  LoanAmount
\
1  LP001003          1             4583             1508.0       128.0

2  LP001005          0             3000                0.0        66.0

3  LP001006          0             2583             2358.0       120.0

4  LP001008          0             6000                0.0       141.0

5  LP001011          2             5417             4196.0       267.0


   Loan_Amount_Term  Credit_History  gender_Male  married_Yes  \
1             360.0             1.0            1            1
2             360.0             1.0            1            1
3             360.0             1.0            1            1
4             360.0             1.0            1            0
5             360.0             1.0            1            1

   education_Not Graduate  property_area_Semiurban
property_area_Urban  \
1                       0                        0
0
2                       0                        0
1
3                       1                        0
1
4                       0                        0
1
5                       0                        0
1

   self_employed_Yes  Loan_status_Y
1                  0              0
2                  1              1
3                  0              1
4                  0              1
5                  1              1
```

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 517 entries, 1 to 613
Data columns (total 14 columns):
 #   Column                   Non-Null Count  Dtype  
---  ------                   --------------  -----  
 0   Loan_ID                  517 non-null    object 
 1   Dependents               517 non-null    object 
 2   ApplicantIncome          517 non-null    int64  
 3   CoapplicantIncome        517 non-null    float64
 4   LoanAmount               517 non-null    float64
 5   Loan_Amount_Term         517 non-null    float64
 6   Credit_History           517 non-null    float64
 7   gender_Male              517 non-null    int64  
 8   married_Yes              517 non-null    int64  
 9   education_Not Graduate   517 non-null    int64  
 10  property_area_Semiurban  517 non-null    int64  
 11  property_area_Urban      517 non-null    int64  
 12  self_employed_Yes        517 non-null    int64  
 13  Loan_status_Y            517 non-null    int64  
dtypes: float64(4), int64(8), object(2)
memory usage: 60.6+ KB
```

```
df.shape
```

```
(517, 14)
```

```
df['Loan_status_Y'].value_counts()
```

```
Loan_status_Y
1    360
0    157
Name: count, dtype: int64
```

```
''' Train Test Split'''
X=df.drop(['Loan_ID','Dependents','Loan_status_Y'],axis=1)
y=df['Loan_status_Y']

from sklearn.model_selection import train_test_split
X_train,X_test, y_train,y_test=train_test_split(X,y,
test_size=0.33,stratify=y,random_state=42)

!pip install --upgrade scikit-learn
!pip install --upgrade imbalanced-learn
```

```
Requirement already satisfied: scikit-learn in c:\users\hp\anaconda3\
lib\site-packages (1.6.1)
Requirement already satisfied: numpy>=1.19.5 in c:\users\hp\anaconda3\
lib\site-packages (from scikit-learn) (1.26.4)
Requirement already satisfied: scipy>=1.6.0 in c:\users\hp\anaconda3\
lib\site-packages (from scikit-learn) (1.13.1)
Requirement already satisfied: joblib>=1.2.0 in c:\users\hp\anaconda3\
```

```
lib\site-packages (from scikit-learn) (1.2.0)
Requirement already satisfied: threadpoolctl>=3.1.0 in c:\users\hp\
anaconda3\lib\site-packages (from scikit-learn) (3.5.0)
Requirement already satisfied: imbalanced-learn in c:\users\hp\
anaconda3\lib\site-packages (0.13.0)
Requirement already satisfied: numpy<3,>=1.24.3 in c:\users\hp\
anaconda3\lib\site-packages (from imbalanced-learn) (1.26.4)
Requirement already satisfied: scipy<2,>=1.10.1 in c:\users\hp\
anaconda3\lib\site-packages (from imbalanced-learn) (1.13.1)
Requirement already satisfied: scikit-learn<2,>=1.3.2 in c:\users\hp\
anaconda3\lib\site-packages (from imbalanced-learn) (1.6.1)
Requirement already satisfied: sklearn-compat<1,>=0.1 in c:\users\hp\
anaconda3\lib\site-packages (from imbalanced-learn) (0.1.3)
Requirement already satisfied: joblib<2,>=1.1.1 in c:\users\hp\
anaconda3\lib\site-packages (from imbalanced-learn) (1.2.0)
Requirement already satisfied: threadpoolctl<4,>=2.0.0 in c:\users\hp\
anaconda3\lib\site-packages (from imbalanced-learn) (3.5.0)
```

```
from imblearn.over_sampling import SMOTE
smote = SMOTE(random_state=42)
X_train_res, y_train_res = smote.fit_resample(X_train, y_train)

model1=DecisionTreeClassifier(class_weight='balanced')
model2=GaussianNB()
model3=LogisticRegression(solver='liblinear',max_iter=200,class_weight
='balanced')

model1.fit(X_train_res,y_train_res)
model2.fit(X_train_res,y_train_res)
model3.fit(X_train_res,y_train_res)
```

```
LogisticRegression(class_weight='balanced', max_iter=200,
solver='liblinear')
```

```
pred1=model1.predict_proba(X_test)
pred2=model2.predict_proba(X_test)
pred3=model3.predict_proba(X_test)

final_pred=(pred1+pred2+pred3)/3

final_pred
```

```
array([[0.39223731, 0.60776269],
       [0.40659715, 0.59340285],
       [0.02848186, 0.97151814],
       [0.04511311, 0.95488689],
       [0.35097035, 0.64902965],
       [0.23817243, 0.76182757],
       [0.29479918, 0.70520082],
       [0.10438856, 0.89561144],
       [0.47694483, 0.52305517],
```

```
       [0.09476933, 0.90523067],
       [0.13907493, 0.86092507],
       [0.14761552, 0.85238448],
       [0.06334219, 0.93665781],
       [0.11690038, 0.88309962],
       [0.10704854, 0.89295146],
       [0.52204028, 0.47795972],
       [0.05678688, 0.94321312],
       [0.04764203, 0.95235797],
       [0.97920344, 0.02079656],
       [0.07782672, 0.92217328],
       [0.77405891, 0.22594109],
       [0.59550243, 0.40449757],
       [0.03753762, 0.96246238],
       [0.0702189 , 0.9297811 ],
       [0.62788621, 0.37211379],
       [0.40903588, 0.59096412],
       [0.02358319, 0.97641681],
       [0.16701355, 0.83298645],
       [0.09439583, 0.90560417],
       [0.61470569, 0.38529431],
       [0.19094428, 0.80905572],
       [0.19994528, 0.80005472],
       [0.03651946, 0.96348054],
       [0.94172623, 0.05827377],
       [0.34386   , 0.65614   ],
       [0.17219798, 0.82780202],
       [0.64870456, 0.35129544],
       [0.24293707, 0.75706293],
       [0.18219011, 0.81780989],
       [0.98506099, 0.01493901],
       [0.60467578, 0.39532422],
       [0.06690004, 0.93309996],
       [0.85849533, 0.14150467],
       [0.05396406, 0.94603594],
       [0.30974308, 0.69025692],
       [0.36173541, 0.63826459],
       [0.37874488, 0.62125512],
       [0.56636961, 0.43363039],
       [0.3878724 , 0.6121276 ],
       [0.10266172, 0.89733828],
       [0.8235694 , 0.1764306 ],
       [0.24950517, 0.75049483],
       [0.48954473, 0.51045527],
       [0.23865234, 0.76134766],
       [0.57781499, 0.42218501],
       [0.36480475, 0.63519525],
       [0.10416488, 0.89583512],
       [0.24594933, 0.75405067],
```

```
       [0.06270525, 0.93729475],
       [0.37853547, 0.62146453],
       [0.08039977, 0.91960023],
       [0.12339853, 0.87660147],
       [0.24639923, 0.75360077],
       [0.16321457, 0.83678543],
       [0.14226174, 0.85773826],
       [0.55648344, 0.44351656],
       [0.2484255 , 0.7515745 ],
       [0.10489714, 0.89510286],
       [0.98770409, 0.01229591],
       [0.97427746, 0.02572254],
       [0.93081756, 0.06918244],
       [0.98136976, 0.01863024],
       [0.13181808, 0.86818192],
       [0.05257288, 0.94742712],
       [0.14986857, 0.85013143],
       [0.11732665, 0.88267335],
       [0.93969747, 0.06030253],
       [0.15023051, 0.84976949],
       [0.11413457, 0.88586543],
       [0.42017344, 0.57982656],
       [0.14599222, 0.85400778],
       [0.96365373, 0.03634627],
       [0.37458732, 0.62541268],
       [0.77455218, 0.22544782],
       [0.23732365, 0.76267635],
       [0.96825252, 0.03174748],
       [0.13666245, 0.86333755],
       [0.12635167, 0.87364833],
       [0.99196877, 0.00803123],
       [0.97274661, 0.02725339],
       [0.04683254, 0.95316746],
       [0.94161224, 0.05838776],
       [0.06478955, 0.93521045],
       [0.05562148, 0.94437852],
       [0.14495654, 0.85504346],
       [0.26132345, 0.73867655],
       [0.18902338, 0.81097662],
       [0.12724161, 0.87275839],
       [0.34600846, 0.65399154],
       [0.09062174, 0.90937826],
       [0.13865821, 0.86134179],
       [0.39696446, 0.60303554],
       [0.3783675 , 0.6216325 ],
       [0.05767018, 0.94232982],
       [0.13440323, 0.86559677],
       [0.54936718, 0.45063282],
       [0.16338474, 0.83661526],
```

```
       [0.24888678, 0.75111322],
       [0.04649542, 0.95350458],
       [0.05810475, 0.94189525],
       [0.06726684, 0.93273316],
       [0.0347083 , 0.9652917 ],
       [0.07277309, 0.92722691],
       [0.04736222, 0.95263778],
       [0.26392414, 0.73607586],
       [0.13004317, 0.86995683],
       [0.0616286 , 0.9383714 ],
       [0.057139  , 0.942861  ],
       [0.24471376, 0.75528624],
       [0.28269427, 0.71730573],
       [0.12476753, 0.87523247],
       [0.24762113, 0.75237887],
       [0.17149273, 0.82850727],
       [0.08219659, 0.91780341],
       [0.27276099, 0.72723901],
       [0.07110289, 0.92889711],
       [0.46447266, 0.53552734],
       [0.13523122, 0.86476878],
       [0.06061216, 0.93938784],
       [0.18129636, 0.81870364],
       [0.95369512, 0.04630488],
       [0.98792005, 0.01207995],
       [0.061814  , 0.938186  ],
       [0.14926299, 0.85073701],
       [0.48353623, 0.51646377],
       [0.30743803, 0.69256197],
       [0.65056053, 0.34943947],
       [0.98192255, 0.01807745],
       [0.24479796, 0.75520204],
       [0.98449143, 0.01550857],
       [0.14107747, 0.85892253],
       [0.06430535, 0.93569465],
       [0.98477633, 0.01522367],
       [0.92251243, 0.07748757],
       [0.99508196, 0.00491804],
       [0.11150255, 0.88849745],
       [0.03620421, 0.96379579],
       [0.14602632, 0.85397368],
       [0.07531721, 0.92468279],
       [0.43218168, 0.56781832],
       [0.20282846, 0.79717154],
       [0.54620235, 0.45379765],
       [0.3055709 , 0.6944291 ],
       [0.96577597, 0.03422403],
       [0.30581131, 0.69418869],
       [0.98497405, 0.01502595],
```

```
       [0.97204098, 0.02795902],
       [0.21292266, 0.78707734],
       [0.28882854, 0.71117146],
       [0.96167384, 0.03832616],
       [0.2462701 , 0.7537299 ],
       [0.15338226, 0.84661774],
       [0.27216106, 0.72783894],
       [0.40335195, 0.59664805],
       [0.23020923, 0.76979077],
       [0.96751815, 0.03248185],
       [0.97383931, 0.02616069],
       [0.18876619, 0.81123381],
       [0.47191153, 0.52808847],
       [0.24653702, 0.75346298],
       [0.9771367 , 0.0228633 ]])

model1.classes_

array([0, 1], dtype=int64)
```

Voting Classifier

```
clf1=DecisionTreeClassifier(class_weight='balanced')
clf2=GaussianNB()
clf3=LogisticRegression(solver='liblinear',max_iter=200,class_weight='
balanced')

eclf1=VotingClassifier(estimators=[('DT',clf1),('GNB',clf2),
('LR',clf3)], voting='hard')
eclf1=eclf1.fit(X_train_res,y_train_res)

predictions=eclf1.predict(X_test)

predictions

array([0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 0,
1,
       1, 1, 0, 0, 1, 1, 1, 0, 1, 1, 1, 0, 1, 1, 0, 1, 1, 0, 0, 1, 0,
1,
       1, 1, 1, 0, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
1,
       0, 1, 0, 0, 0, 0, 1, 1, 1, 1, 0, 1, 1, 0, 1, 0, 1, 0, 1, 0, 1,
1,
       0, 0, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1,
1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 0,
0,
       1, 1, 0, 1, 0, 0, 0, 0, 1, 1, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1,
0,
```

```
       1, 0, 0, 1, 1, 0, 1, 1, 1, 1, 1, 0, 0, 1, 1, 1, 0],
dtype=int64)
```

```
from sklearn.metrics import classification_report , confusion_matrix

print(classification_report(y_test,predictions))
print(confusion_matrix(y_test,predictions))
```

```
              precision    recall  f1-score   support

           0       0.70      0.62      0.65        52
           1       0.84      0.88      0.86       119

    accuracy                           0.80       171
   macro avg       0.77      0.75      0.76       171
weighted avg       0.80      0.80      0.80       171

[[ 32  20]
 [ 14 105]]
```

```
eclf2=VotingClassifier(estimators=[('DT',clf1),('GNB',clf2),
('LR',clf3)], voting='soft')
eclf2=eclf2.fit(X_train,y_train)

prediction=eclf2.predict(X_test)

from sklearn.metrics import classification_report , confusion_matrix

print(classification_report(y_test,prediction))
print(confusion_matrix(y_test,prediction))
```

```
              precision    recall  f1-score   support

           0       0.67      0.60      0.63        52
           1       0.83      0.87      0.85       119

    accuracy                           0.79       171
   macro avg       0.75      0.74      0.74       171
weighted avg       0.78      0.79      0.79       171

[[ 31  21]
 [ 15 104]]
```

```
from sklearn.ensemble import BaggingClassifier
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import accuracy_score
base_model1=DecisionTreeClassifier()
bagging = BaggingClassifier(estimator=base_model1, n_estimators=200,
random_state=0)
bagging.fit(X_train_res, y_train_res)
y_pred_bagging = bagging.predict(X_test)
print("Bagging Accuracy:", accuracy_score(y_test, y_pred_bagging))
```

```
Bagging Accuracy: 0.7660818713450293
```

```
from sklearn.ensemble import RandomForestClassifier

rf = RandomForestClassifier(n_estimators=100, random_state=0)
rf.fit(X_train_res, y_train_res)
y_pred_rf = rf.predict(X_test)
print("Random Forest Accuracy:", accuracy_score(y_test, y_pred_rf))
```

```
Random Forest Accuracy: 0.7953216374269005
```

```
from sklearn.ensemble import AdaBoostClassifier

boosting = AdaBoostClassifier(n_estimators=200, random_state=0)
boosting.fit(X_train_res, y_train_res)
y_pred_boost = boosting.predict(X_test)
print("Boosting (AdaBoost) Accuracy:", accuracy_score(y_test,
y_pred_boost))
```

```
Boosting (AdaBoost) Accuracy: 0.7777777777777778
```