```
import pandas as pd
from sklearn.preprocessing import LabelEncoder
le = LabelEncoder()
from sklearn.preprocessing import MinMaxScaler
scaler = MinMaxScaler()

from sklearn.model_selection import train_test_split
from sklearn.ensemble import AdaBoostClassifier
from sklearn.metrics import accuracy_score
from sklearn.ensemble import GradientBoostingClassifier
from sklearn.ensemble import RandomForestClassifier

data =
pd.read_csv('C:\\Users\\PandaBas\\Desktop\\Personal\\Teaching\\5thSem\\PartB
\\loanapproval.csv')
data.head()

# Checking for Missing values
data.isnull().sum()

# convert string variable to One Hot Encoding
dummyfied_data = data.apply(le.fit_transform)
dummyfied_data.head()

# Scalling the numeric column
col_to_scale =['age']
dummyfied_data[col_to_scale] = scaler.fit_transform(dummyfied_data[col_to_scale])
dummyfied_data.head()

X = dummyfied_data.drop('Loan_approved',axis = 1)
Y = dummyfied_data[['Loan_approved']]


X_train, X_test, y_train, y_test = train_test_split(X,Y, test_size = 0.20,
random_state = 42)

test_df = pd.concat ([X_test,y_test],axis =1 )

# Adaboost classifier
abc = AdaBoostClassifier(random_state=0)
abc_model = abc.fit(X_train, y_train)
abc_pred = abc_model.predict(X_test)
test_df['abc_pred'] = abc_pred
# test_df.to_csv('classification_results.csv')
score = accuracy_score(y_test, abc_pred)
```

```
score

# creating a RF classifier, training a random Forest model
clf = RandomForestClassifier(random_state = 42)
clf.fit(X_train, y_train)
# Predicting on Test data and checking the accuracy. This can be extended to
train data as well
rf_pred = clf.predict(X_test)
test_df['rf_pred'] = rf_pred
score = accuracy_score(y_test, rf_pred)
score

# Gradient boosting classifier

gbcl = GradientBoostingClassifier(random_state = 42)
gbcl.fit(X_train, y_train)
gbcl_pred = gbcl.predict(X_test)
test_df['gbcl_pred'] = gbcl_pred
# test_df.to_csv('classification_results.csv')
score = accuracy_score(y_test, gbcl_pred)
score

# Ensemble the ensembles

column_names = ['abc_pred', 'rf_pred', 'gbcl_pred']
test_df['sum'] = test_df[column_names].sum(axis=1)
test_df['ensembeled_pred'] = [1 if x > 1 else 0 for x in test_df['sum'] ]
score = accuracy_score(y_test, test_df['ensembeled_pred'])
score
```