```
# Import important library
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
import warnings
from sklearn.model_selection import train_test_split,cross_val_score # Import train_test_split function
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import classification_report,confusion_matrix
warnings.filterwarnings('ignore')
```

Read the input file and check the data dimension

1. List item
2. List item

```
from google.colab import drive
drive.mount('/content/drive')
file = '/content/drive/My Drive/PGML/unsupervised/lab/german_credit.csv'
```

```
Go to this URL in a browser: https://accounts.google.com/o/oauth2/auth?client_id=947318989803-6bn6qk8qdgf4n4g

Enter your authorization code:
··········
Mounted at /content/drive
```

```
df = pd.read_csv(file)
df.head()
```

.ncome_perc	personal_status_sex	other_debtors	present_res_since	property	age	other_installment_plans	hou:
4	male : single	none	4	real estate	67	none	
2	female : divorced/separated/married	none	2	real estate	22	none	
2	male : single	none	3	real estate	49	none	
2	male : single	guarantor	4	if not A121 : building society savings agreeme...	45	none	fo
3	male : single	none	4	unknown / no property	53	none	fo

```
df.shape
```

```
(1000, 21)
```

```
# You can access from https://www.kaggle.com/uciml/german-credit
#Read input file and understand the data
# "default" is my dependent variable
```

Q1 Randomly select 50% data for this use case(1 Marks)

Hint: Use train_test_split

```
new_df = df.sample(frac = 0.5)
new_df.shape
```

```
(500, 21)
```

```
# Lets build a Ensemble model but need to modify the dataset first
dmColumn = ['account_check_status','credit_history','purpose','savings','present_emp_since','personal_status_sex','other_de
```

```
account_check_status
credit_history
purpose
savings
present_emp_since
personal_status_sex
other_debtors
property
other_installment_plans
housing
job
telephone
foreign_worker
```

Q2.Prepare the model data by converting non-numeric to dummy (1 Marks)

Hint: Use get_dummies

```
new_df = pd.get_dummies(new_df,dmColumn)
```

```
new_df.head()
```

	default	duration_in_month	credit_amount	installment_as_income_perc	present_res_since	age	credits_thi
263	0	12	2748	2	4	57	
761	1	18	2124	4	4	24	
530	0	36	2273	3	1	32	
29	1	60	6836	3	4	63	
644	0	18	1880	4	1	32	

```
# Print Shape of model data
new_df.shape
```

```
(500, 62)
```

Drop the original variables which are converted to dummy

```
# after running the get_dummy, not able to get the original varaible, which can delete
new_df.columns
```

```
Index(['default', 'duration_in_month', 'credit_amount',
       'installment_as_income_perc', 'present_res_since', 'age',
       'credits_this_bank', 'people_under_maintenance',
       'account_check_status_0 <= ... < 200 DM', 'account_check_status_< 0 DM',
       'account_check_status_>= 200 DM / salary assignments for at least 1 year',
       'account_check_status_no checking account',
       'credit_history_all credits at this bank paid back duly',
       'credit_history_critical account/ other credits existing (not at this bank)',
       'credit_history_delay in paying off in the past',
       'credit_history_existing credits paid back duly till now',
       'credit_history_no credits taken/ all credits paid back duly',
       'purpose_(vacation - does not exist?)', 'purpose_business',
       'purpose_car (new)', 'purpose_car (used)',
       'purpose_domestic appliances', 'purpose_education',
       'purpose_furniture/equipment', 'purpose_radio/television',
       'purpose_repairs', 'purpose_retraining', 'savings_.. >= 1000 DM ',
       'savings_... < 100 DM', 'savings_100 <= ... < 500 DM',
       'savings_500 <= ... < 1000 DM ', 'savings_unknown/ no savings account',
       'present_emp_since_.. >= 7 years', 'present_emp_since_... < 1 year ',
       'present_emp_since_1 <= ... < 4 years',
       'present_emp_since_4 <= ... < 7 years', 'present_emp_since_unemployed',
       'personal_status_sex_female : divorced/separated/married',
       'personal_status_sex_male : divorced/separated',
       'personal_status_sex_male : married/widowed',
       'personal_status_sex_male : single', 'other_debtors_co-applicant',
       'other_debtors_guarantor', 'other_debtors_none',
       'property_if not A121 : building society savings agreement/ life insurance',
       'property_if not A121/A122 : car or other, not in attribute 6',
       'property_real estate', 'property_unknown / no property',
       'other_installment_plans_bank', 'other_installment_plans_none',
       'other_installment_plans_stores', 'housing_for free', 'housing_own',
       'housing_rent',
       'job_management/ self-employed/ highly qualified employee/ officer',
       'job_skilled employee / official',
       'job_unemployed/ unskilled - non-resident', 'job_unskilled - resident',
       'telephone_none', 'telephone_yes, registered under the customers name ',
       'foreign_worker_no', 'foreign_worker_yes'],
      dtype='object')
```

Check for highly correlated variables but don't required any treatment for this use case

```
new_df.corr()
```

	default	duration_in_month	credit_amount	installment_as_income_perc	present_re
default	1.000000	0.203763	0.139345	0.107035	
duration_in_month	0.203763	1.000000	0.590847	0.036163	
credit_amount	0.139345	0.590847	1.000000	-0.311270	
installment_as_income_perc	0.107035	0.036163	-0.311270	1.000000	
present_res_since	0.024055	0.049656	0.030403	0.066241	
age	-0.069179	-0.048150	-0.013888	0.107738	
credits_this_bank	-0.045812	-0.042464	-0.003339	0.027093	
people_under_maintenance	0.089978	0.039022	0.030336	-0.043010	
account_check_status_0 <= ... < 200 DM	0.118656	0.066569	0.108813	-0.034931	-
account_check_status_< 0 DM	0.269169	0.082690	-0.030678	0.056154	
account_check_status_>= 200 DM / salary assignments for at least 1 year	-0.085136	-0.074219	-0.085958	-0.037409	-
account_check_status_no checking account	-0.307143	-0.096610	-0.024500	-0.001516	
credit_history_all credits at this bank paid back duly	0.140081	0.056605	-0.020065	0.025318	
credit_history_critical account/ other credits existing (not at this bank)	-0.174769	-0.060674	-0.040141	0.052313	
credit_history_delay in paying off in the past	0.029142	0.117562	0.128376	-0.052343	-

off in the past					
credit_history_existing credits paid back duly till now	0.022285	-0.086952	-0.080193	-0.009344	-
credit_history_no credits taken/ all credits paid back duly	0.131416	0.114675	0.118343	-0.042482	-
purpose_(vacation - does not exist?)	0.046664	0.001888	0.042433	0.019889	-
purpose_business	0.029878	0.207498	0.108297	-0.006873	-
purpose_car (new)	0.073250	-0.081040	-0.037926	0.042704	
purpose_car (used)	-0.096507	0.111668	0.243329	-0.149525	
purpose_domestic appliances	-0.076402	-0.070911	-0.180452	0.112199	-
purpose_education	0.063092	0.021706	-0.012413	0.052121	-
purpose_furniture/equipment	0.088746	0.088358	0.376043	-0.103067	-
purpose_radio/television	0.013118	-0.067735	-0.076596	-0.078836	-
purpose_repairs	-0.009424	-0.030504	-0.057107	0.061577	-
purpose_retraining	-0.040419	-0.087327	-0.107328	0.030991	
savings_.. >= 1000 DM	-0.080447	-0.064761	-0.065404	0.043357	
savings_... < 100 DM	0.121391	0.002882	-0.002382	0.005239	-
savings_100 <= ... < 500 DM	0.045923	0.018466	-0.032849	-0.035532	
...	...	...	...	...	
present_emp_since_.. >= 7 years	-0.090910	-0.005838	-0.050035	0.139293	
present_emp_since_... < 1 year	0.077805	-0.040470	-0.058305	0.018410	-
present_emp_since_1 <= ... < 4 years	0.013303	0.001938	0.050637	-0.118613	-
present_emp_since_4 <= ... < 7 years	-0.056818	0.048589	0.021428	-0.025022	
present_emp_since_unemployed	0.115191	-0.005863	0.055559	-0.016662	

personal_status_sex_female : divorced/separated/married	0.046159	-0.096802	-0.070419	-0.090061	
personal_status_sex_male : divorced/separated	-0.005624	0.001565	0.094541	-0.155765	-
personal_status_sex_male : married/widowed	-0.035754	-0.099760	-0.138210	-0.042667	-
personal_status_sex_male : single	-0.017953	0.149356	0.111274	0.171684	
other_debtors_co-applicant	0.044795	0.037937	0.135211	-0.026658	
other_debtors_guarantor	0.003048	-0.019830	-0.057725	-0.035410	-
other_debtors_none	-0.034676	-0.013124	-0.056162	0.044987	
property_if not A121 : building society savings agreement/ life insurance	-0.011462	-0.071393	-0.062726	-0.013678	
property_if not A121/A122 : car or other, not in attribute 6	0.016053	0.113495	0.140710	-0.016765	-
property_real estate	-0.143527	-0.230488	-0.265003	-0.031001	-
property_unknown / no property	0.161343	0.208255	0.202970	0.074605	
other_installment_plans_bank	0.119658	0.014060	0.032954	0.004582	-
other_installment_plans_none	-0.130725	-0.054279	-0.057846	-0.003861	
other_installment_plans_stores	0.044795	0.077252	0.052742	-0.000403	-
housing_for free	0.107492	0.151314	0.123321	0.083460	
housing_own	-0.181326	-0.060555	-0.080006	0.001138	-
housing_rent	0.127730	-0.050277	-0.004752	-0.068546	
job_management/ self-employed/ highly qualified employee/ officer	0.081416	0.114979	0.329662	0.034581	
job_skilled employee / official	-0.053164	0.059884	-0.090458	0.035582	-

job_unemployed/ unskilled - non-resident	0.000625	-0.035920	-0.048710	-0.091932	-
job_unskilled - resident	-0.010019	-0.165906	-0.174814	-0.042628	
telephone_none	0.015267	-0.123649	-0.255544	-0.021034	-
telephone_yes, registered under the customers name	-0.015267	0.123649	0.255544	0.021034	
foreign_worker_no	-0.066051	-0.124231	-0.026569	-0.037798	-
foreign_worker_yes	0.066051	0.124231	0.026569	0.037798	

62 rows × 62 columns

```
(500, 61)
```

Q3 Split Train/Test data 70:30 ratio(1 Marks)

Hint:from sklearn.model_selection import train_test_split

```
y = new_df['default']
#droping default from new_df
new_df = new_df.drop('default',1)
new_df.columns
```

```
Index(['duration_in_month', 'credit_amount', 'installment_as_income_perc',
       'present_res_since', 'age', 'credits_this_bank',
       'people_under_maintenance', 'account_check_status_0 <= ... < 200 DM',
       'account_check_status_< 0 DM',
       'account_check_status_>= 200 DM / salary assignments for at least 1 year',
       'account_check_status_no checking account',
       'credit_history_all credits at this bank paid back duly',
       'credit_history_critical account/ other credits existing (not at this bank)',
       'credit_history_delay in paying off in the past',
       'credit_history_existing credits paid back duly till now',
       'credit_history_no credits taken/ all credits paid back duly',
       'purpose_(vacation - does not exist?)', 'purpose_business',
       'purpose_car (new)', 'purpose_car (used)',
       'purpose_domestic appliances', 'purpose_education',
       'purpose_furniture/equipment', 'purpose_radio/television',
       'purpose_repairs', 'purpose_retraining', 'savings_.. >= 1000 DM ',
       'savings_... < 100 DM', 'savings_100 <= ... < 500 DM',
       'savings_500 <= ... < 1000 DM ', 'savings_unknown/ no savings account',
       'present_emp_since_.. >= 7 years', 'present_emp_since_... < 1 year ',
       'present_emp_since_1 <= ... < 4 years',
       'present_emp_since_4 <= ... < 7 years', 'present_emp_since_unemployed',
       'personal_status_sex_female : divorced/separated/married',
       'personal_status_sex_male : divorced/separated',
       'personal_status_sex_male : married/widowed',
       'personal_status_sex_male : single', 'other_debtors_co-applicant',
       'other_debtors_guarantor', 'other_debtors_none',
       'property_if not A121 : building society savings agreement/ life insurance',
       'property_if not A121/A122 : car or other, not in attribute 6',
       'property_real estate', 'property_unknown / no property',
       'other_installment_plans_bank', 'other_installment_plans_none',
       'other_installment_plans_stores', 'housing_for free', 'housing_own',
       'housing_rent',
       'job_management/ self-employed/ highly qualified employee/ officer',
       'job_skilled employee / official',
       'job_unemployed/ unskilled - non-resident', 'job_unskilled - resident',
       'telephone_none', 'telephone_yes, registered under the customers name ',
       'foreign_worker_no', 'foreign_worker_yes'],
      dtype='object')
```

```
X = new_df
```

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=1) # 70% training and 30% test
```

Q4 Build Random Forest Model(1 Marks)

Hint:from sklearn.ensemble import RandomForestClassifier using n_jobs=2,n_estimators=500,criterion="entropy",random_state=9999

```
print("RandomForestClassifier")
randomforest = RandomForestClassifier(n_jobs=2,n_estimators=500,criterion="entropy",random_state=9999).fit(X_train, y_trair

# Predict target variables y for test data
y_pred = randomforest.predict(X_test)
```

```
RandomForestClassifier
```

Q5 Calculate Confusion Matrix and Accuracy score (1 Marks)

Hint: Use confusion_matrix and accuracy_score

```
scores1 = cross_val_score(randomforest, X_train, y_train, cv=10, scoring='accuracy')
scores2 = cross_val_score(randomforest, X_train, y_train, cv=10, scoring='precision')
scores3 = cross_val_score(randomforest, X_train, y_train, cv=10, scoring='roc_auc')
# The mean score and standard deviation of the score estimate
print("Cross Validation Accuracy: %0.2f (+/- %0.2f)" % (scores1.mean(), scores1.std()))
print("Cross Validation Precision: %0.2f (+/- %0.2f)" % (scores2.mean(), scores2.std()))
print("Cross Validation roc_auc: %0.2f (+/- %0.2f)" % (scores3.mean(), scores3.std()))
```

```
Cross Validation Accuracy: 0.75 (+/- 0.07)
Cross Validation Precision: 0.71 (+/- 0.26)
Cross Validation roc_auc: 0.75 (+/- 0.10)
```

```
print("Confusing Metrix")
cm = confusion_matrix(y_test,y_pred)
print(cm)
```

```
Confusing Metrix
[[97  8]
 [26 19]]
```

Q6 Show the list of the features importance(1 Marks)

```
feature_importances = pd.DataFrame(randomforest.feature_importances_,index = X_train.columns,columns=['importance']).sort_v
with pd.option_context('display.max_rows', None, 'display.max_columns', None):  # more options can be specified also
    print(feature_importances)
```

```
                                                    importance
credit_amount                                         0.094215
duration_in_month                                     0.076102
age                                                   0.071744
account_check_status_no checking account              0.057527
installment_as_income_perc                            0.035257
present_res_since                                     0.033691
account_check_status_< 0 DM                           0.030234
credit_history_critical account/ other credits ...    0.022855
housing_own                                           0.021480
account_check_status_0 <= ... < 200 DM                0.018460
savings_... < 100 DM                                  0.017782
other_installment_plans_none                          0.017601
credits_this_bank                                     0.017159
savings_unknown/ no savings account                   0.016308
property_unknown / no property                        0.016087
present_emp_since_.. >= 7 years                       0.015882
personal_status_sex_female : divorced/separated...    0.015880
property_real estate                                  0.015779
job_skilled employee / official                       0.015359
people_under_maintenance                              0.015276
present_emp_since_1 <= ... < 4 years                  0.015153
purpose_business                                      0.014942
purpose_domestic appliances                           0.014876
personal_status_sex_male : single                     0.014745
credit_history_existing credits paid back duly ...    0.014528
purpose_radio/television                              0.013207
property_if not A121 : building society savings...    0.013130
property_if not A121/A122 : car or other, not i...    0.013085
job_management/ self-employed/ highly qualified...    0.013017
purpose_car (new)                                     0.012924
telephone_none                                        0.012869
other_installment_plans_bank                          0.012858
job_unskilled - resident                              0.012475
telephone_yes, registered under the customers n...    0.012165
credit_history_no credits taken/ all credits pa...    0.011768
housing_rent                                          0.011752
present_emp_since_4 <= ... < 7 years                  0.011518
present_emp_since_... < 1 year                        0.011494
credit_history_delay in paying off in the past        0.011382
present_emp_since_unemployed                          0.011189
credit_history_all credits at this bank paid ba...    0.010120
housing_for free                                      0.009722
```

```
purpose_car (used)                                        0.009361
personal_status_sex_male : married/widowed                0.008446
savings_100 <= ... < 500 DM                               0.008242
other_debtors_none                                        0.008055
other_installment_plans_stores                            0.007682
purpose_(vacation - does not exist?)                      0.006877
account_check_status_>= 200 DM / salary assignm...        0.005659
savings_500 <= ... < 1000 DM                              0.005089
savings_.. >= 1000 DM                                     0.004947
other_debtors_co-applicant                                0.004896
personal_status_sex_male : divorced/separated             0.004076
foreign_worker_no                                         0.004034
foreign_worker_yes                                        0.003915
other_debtors_guarantor                                   0.003698
purpose_education                                         0.003347
purpose_furniture/equipment                               0.002594
job_unemployed/ unskilled - non-resident                  0.002519
purpose_retraining                                        0.001901
purpose_repairs                                           0.001063
```

Q7 K-fold cross-validation(2 Marks)

k-fold cross validation(without stratification)

Usually k is set as 10-20 in practical settings, depends on data set size

```
from sklearn.model_selection import KFold
from sklearn.model_selection import cross_val_score
```

```
# Use below values
num_folds = 10
seed = 77
```

```
#Validate the Random Forest model build above using k fold
```

```
kfold = KFold(n_splits=num_folds, random_state=seed)
num_trees = 100
max_features = 3
model = RandomForestClassifier(n_estimators=num_trees, max_features=max_features)
results = cross_val_score(model, X, y, cv=kfold)
```

```
print(results)
```

```
[0.72 0.88 0.78 0.72 0.6  0.76 0.72 0.76 0.68 0.76]
```

```
#Calculate Mean score
```

```
print(results.mean())
```

```
0.7379999999999999
```

```
# Calculate score standard deviation using std()
```

```
print(results.std())
```

```
0.06838128398911504
```

Q8 Print the confusion matrix(1 Marks)

```
# there is no clearity for what you are asking the confusing matrix, is it for confusing matrix. Trying the bellow but it i

cm = confusion_matrix(y_test,results)
print(cm)
```

Q9.Classification accuracy:

percentage of correct predictions and Calculate sensitivity (or True Positive Rate or Recall) and Precision. (1 Marks)

```
from sklearn import metrics #Import scikit-learn metrics module for accuracy calculation
print("Accuracy:",metrics.accuracy_score(y_test, y_pred))
```

```
Accuracy: 0.7733333333333333
```

Q10.Plot Receiver Operating Characteristic (ROC) Curves(1 Marks)

```
#Hint: Use roc_curve
```

```
# not able to relate
```

ROC curve can help you to choose a threshold that balances sensitivity and specificity in a way that makes sense for your particular context

Q11. Calculate AUC(the percentage of the ROC plot that is underneath the cu - optional

Bootstrapping (Bonus)

Given a dataset of size n, a bootstrap sample is created by sampling n instances uniformly from the data (with/without replacement)

Create a model with each bootstrap sample and validate it with the test set

Final result is calculated by averaging the accuracy of models

↳ *4 cells hidden*