Karanjot Singh

In [3074]:

```
import pandas as pd
import numpy as np
import seaborn as sns
from pandas import DataFrame
import warnings
warnings.filterwarnings("ignore")
```

In [3075]:

```
df = pd.read_csv(r'loan.tsv',sep='\t')
```

In [3076]:

```
print(df.head())
```

```
   Age     Sex  Job Housing Saving accounts Checking account  Credit amoun
t  \
0   67    male    2     own             NaN           little           116
9
1   22  female    2     own          little         moderate           595
1
2   49    male    1     own          little              NaN           209
6
3   45    male    2    free          little           little           788
2
4   53    male    2    free          little           little           487
0

   Duration              Purpose  Risk
0         6             radio/TV  good
1        48             radio/TV   bad
2        12            education  good
3        42  furniture/equipment  good
4        24                  car   bad
```

In [3077]:

```
df.isnull().sum()
```

Out[3077]:

```
Age                 0
Sex                 0
Job                 0
Housing             0
Saving accounts     180
Checking account    382
Credit amount       0
Duration            0
Purpose             0
Risk                0
dtype: int64
```

In [3078]:

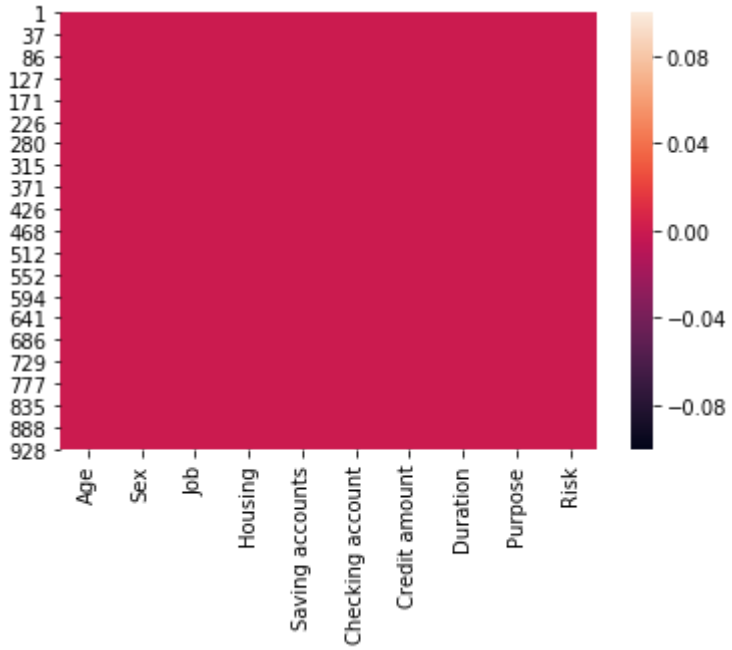
```
df.dropna(inplace=True)
```

In [3079]:

```
df.shape
sns.heatmap(df.isnull())
```

Out[3079]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x16dc0de15c8>
```



In [3080]:

```
df.groupby('Checking account').count()
```

Out[3080]:

Checking account	Age	Sex	Job	Housing	Saving accounts	Credit amount	Duration	Purpose	Risk
little	237	237	237	237	237	237	237	237	237
moderate	210	210	210	210	210	210	210	210	210
rich	53	53	53	53	53	53	53	53	53

In [3081]:

```
from sklearn.preprocessing import LabelEncoder
from sklearn.preprocessing import OneHotEncoder
```

In [3082]:

```
df['Checking account'] = df['Checking account'].replace('little','little_ca')
df['Checking account'] = df['Checking account'].replace('moderate','moderate_ca')
df['Checking account'] = df['Checking account'].replace('rich','rich_ca')
df['Sex']=LabelEncoder().fit_transform(df['Sex'])
housing_data=pd.get_dummies(df['Housing'])
df = df.merge(housing_data, left_index=True, right_index=True, how='inner')
purpose_data=pd.get_dummies(df['Purpose'])
df = df.merge(purpose_data, left_index=True, right_index=True, how='inner')
checking_account_data=pd.get_dummies(df['Checking account'])
df = df.merge(checking_account_data, left_index=True, right_index=True, how='inner')
saving_accounts_data=pd.get_dummies(df['Saving accounts'])
df = df.merge(saving_accounts_data, left_index=True, right_index=True, how='inner')
df['Risk']=LabelEncoder().fit_transform(df['Risk'])
df.drop(['Housing','Purpose'],axis=1,inplace=True)
df_sa = df.reindex(columns = ['Age','Sex','Job','free','own','rent','Saving accounts',
'little_ca','moderate_ca','rich_ca','Credit amount','Duration','business','car','domest
ic appliances','education','furniture/equipment','radio/TV','repairs','vacation/others'
,'Risk'])
df_ca = df.reindex(columns = ['Age','Sex','Job','free','own','rent','little','quite ric
h','moderate','rich','Checking account','Credit amount','Duration','business','car','do
mestic appliances','education','furniture/equipment','radio/TV','repairs','vacation/oth
ers','Risk'])
df = df.reindex(columns = ['Age','Sex','Job','free','own','rent','little','quite rich',
'moderate','rich','little_ca','moderate_ca','rich_ca','Credit amount','Duration','busin
ess','car','domestic appliances','education','furniture/equipment','radio/TV','repairs'
,'vacation/others','Risk'])

print(df_sa.info())
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 500 entries, 1 to 963
Data columns (total 21 columns):
Age                     500 non-null int64
Sex                     500 non-null int32
Job                     500 non-null int64
free                    500 non-null uint8
own                     500 non-null uint8
rent                    500 non-null uint8
Saving accounts         500 non-null object
little_ca               500 non-null uint8
moderate_ca             500 non-null uint8
rich_ca                 500 non-null uint8
Credit amount           500 non-null int64
Duration                500 non-null int64
business                500 non-null uint8
car                     500 non-null uint8
domestic appliances     500 non-null uint8
education               500 non-null uint8
furniture/equipment     500 non-null uint8
radio/TV                500 non-null uint8
repairs                 500 non-null uint8
vacation/others         500 non-null uint8
Risk                    500 non-null int32
dtypes: int32(2), int64(4), object(1), uint8(14)
memory usage: 54.2+ KB
None
```

In [3083]:

```
print(df.info())
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 500 entries, 1 to 963
Data columns (total 24 columns):
Age                     500 non-null int64
Sex                     500 non-null int32
Job                     500 non-null int64
free                    500 non-null uint8
own                     500 non-null uint8
rent                    500 non-null uint8
little                  500 non-null uint8
quite rich              500 non-null uint8
moderate                500 non-null uint8
rich                    500 non-null uint8
little_ca               500 non-null uint8
moderate_ca             500 non-null uint8
rich_ca                 500 non-null uint8
Credit amount           500 non-null int64
Duration                500 non-null int64
business                500 non-null uint8
car                     500 non-null uint8
domestic appliances     500 non-null uint8
education               500 non-null uint8
furniture/equipment     500 non-null uint8
radio/TV                500 non-null uint8
repairs                 500 non-null uint8
vacation/others         500 non-null uint8
Risk                    500 non-null int32
dtypes: int32(2), int64(4), uint8(18)
memory usage: 52.2 KB
None
```

In [3084]:

```
from sklearn.metrics import classification_report,confusion_matrix
from sklearn.model_selection import KFold
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score
#print(df_sa)
X=df_sa.drop(['Saving accounts','repairs'],axis=1)
y=df_sa['Saving accounts']
kf = KFold(n_splits=60)
kf.get_n_splits(X)
for train_index, test_index in kf.split(X):
    X_train, X_test = X.iloc[train_index], X.iloc[test_index]
    y_train, y_test = y.iloc[train_index], y.iloc[test_index]

forest = RandomForestClassifier(n_estimators=1000)
rf = forest.fit(X_train,y_train)
predicted=forest.predict(X_test)
print(accuracy_score(y_test,predicted)*100)
print(confusion_matrix(y_test,predicted))
```

```
87.5
[[7 0]
 [1 0]]
```

In [3085]:

```
raw_data = pd.read_csv(r'loan.tsv',sep='\t')
cols = ['Saving accounts','Checking account']
j = raw_data[raw_data[cols].isnull().all(axis = 1)].index
raw_data = raw_data.drop(j,axis = 0)
sa_nan = raw_data[raw_data['Saving accounts'].isnull()]

sa_nan['Checking account'] = sa_nan['Checking account'].replace('little','little_ca')
sa_nan['Checking account'] = sa_nan['Checking account'].replace('moderate','moderate_c
a')
sa_nan['Checking account'] = sa_nan['Checking account'].replace('rich','rich_ca')
sa_nan['Sex']=LabelEncoder().fit_transform(sa_nan['Sex'])
housing_data=pd.get_dummies(sa_nan['Housing'])
sa_nan = sa_nan.merge(housing_data, left_index=True, right_index=True, how='inner')
purpose_data=pd.get_dummies(sa_nan['Purpose'])
sa_nan = sa_nan.merge(purpose_data, left_index=True, right_index=True, how='inner')
checking_account_data=pd.get_dummies(sa_nan['Checking account'])
sa_nan = sa_nan.merge(checking_account_data, left_index=True, right_index=True, how='in
ner')
sa_nan['Risk']=LabelEncoder().fit_transform(sa_nan['Risk'])
sa_nan.drop(['Housing','Purpose'],axis=1,inplace=True)

sa_nan = sa_nan.reindex(columns = ['Age','Sex','Job','free','own','rent','little_ca','m
oderate_ca','rich_ca','Credit amount','Duration','business','car','domestic appliances'
,'education','furniture/equipment','radio/TV','vacation/others','Risk'])
#print(sa_nan)
```

In [3086]:

```
pred=forest.predict(sa_nan)
#print(pred)

sa_nan['Saving accounts'] = pred
saving_accounts_data=pd.get_dummies(sa_nan['Saving accounts'])
sa_nan = sa_nan.merge(saving_accounts_data, left_index=True, right_index=True, how='inn
er')
sa_nan.drop(['Saving accounts'],axis=1,inplace=True)

sa_nan = sa_nan.reindex(columns = ['Age','Sex','Job','free','own','rent','little','mode
rate','rich','quite rich','little_ca','moderate_ca','rich_ca','Credit amount','Duratio
n','business','car','domestic appliances','education','furniture/equipment','radio/TV',
'vacation/others','repairs','Risk'])

#print(sa_nan)
```

In [3087]:

```
raw_data = pd.read_csv(r'loan.tsv',sep='\t')
cols = ['Saving accounts','Checking account']
j = raw_data[raw_data[cols].isnull().all(axis = 1)].index
raw_data = raw_data.drop(j,axis = 0)
ca_nan = raw_data[raw_data['Checking account'].isnull()]

ca_nan['Sex']=LabelEncoder().fit_transform(ca_nan['Sex'])
housing_data=pd.get_dummies(ca_nan['Housing'])
ca_nan = ca_nan.merge(housing_data, left_index=True, right_index=True, how='inner')
purpose_data=pd.get_dummies(ca_nan['Purpose'])
ca_nan = ca_nan.merge(purpose_data, left_index=True, right_index=True, how='inner')
saving_accounts_data=pd.get_dummies(ca_nan['Saving accounts'])
ca_nan = ca_nan.merge(saving_accounts_data, left_index=True, right_index=True, how='inn
er')
ca_nan['Risk']=LabelEncoder().fit_transform(ca_nan['Risk'])
ca_nan.drop(['Housing','Purpose'],axis=1,inplace=True)

ca_nan = ca_nan.reindex(columns = ['Age','Sex','Job','free','own','rent','little','mode
rate','rich','quite rich','Credit amount','Duration','business','car','domestic applian
ces','education','furniture/equipment','radio/TV','vacation/others','repairs','Risk'])
#print(ca_nan)
```

In [3088]:

```
from sklearn.metrics import classification_report,confusion_matrix
from sklearn.model_selection import KFold
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score

X=df_ca.drop(['Checking account'],axis=1)
y=df_ca['Checking account']
kf = KFold(n_splits=60)

for train_index, test_index in kf.split(X):
    X_train, X_test = X.iloc[train_index], X.iloc[test_index]
    y_train, y_test = y.iloc[train_index], y.iloc[test_index]

forest = RandomForestClassifier(n_estimators=1000)
rf = forest.fit(X_train,y_train)
predicted=forest.predict(X_test)
print(accuracy_score(y_test,predicted)*100)
print(confusion_matrix(y_test,predicted))
```

```
75.0
[[5 0 0]
 [1 1 0]
 [0 1 0]]
```

In [3089]:

```
pred = forest.predict(ca_nan)
#print(pred)
ca_nan['Checking account'] = pred

checking_account_data=pd.get_dummies(ca_nan['Checking account'])
ca_nan = ca_nan.merge(checking_account_data, left_index=True, right_index=True, how='in
ner')
ca_nan.drop(['Checking account'],axis=1,inplace=True)

ca_nan = ca_nan.reindex(columns = ['Age','Sex','Job','free','own','rent','little','mode
rate','rich','quite rich','little_ca','moderate_ca','rich_ca','Credit amount','Duratio
n','business','car','domestic appliances','education','furniture/equipment','radio/TV',
'vacation/others','repairs','Risk'])
#print(ca_nan)
```

In [3090]:

```
df1 = pd.concat([sa_nan,ca_nan], axis=0)
df = pd.concat([df,df1],axis = 0)
df = df.fillna(0)

#print(df)
```

In [3091]:

```
from sklearn.model_selection import KFold
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score
X=df.drop(['Risk'],axis=1)
y=df['Risk']
kf = KFold(n_splits=25)

for train_index, test_index in kf.split(X):
    X_train, X_test = X.iloc[train_index], X.iloc[test_index]
    y_train, y_test = y.iloc[train_index], y.iloc[test_index]


forest = RandomForestClassifier(n_estimators=1000)
rf = forest.fit(X_train,y_train)
predicted=forest.predict(X_test)
print(accuracy_score(y_test,predicted)*100)
print(confusion_matrix(y_test,predicted))
```

```
85.29411764705883
[[ 2  4]
 [ 1 27]]
```