```python
import pandas as pd
import numpy as np
from statsmodels.formula.api import ols
from scipy import stats
import statsmodels.api as sm
from datetime import datetime
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import StratifiedKFold
from sklearn.preprocessing import StandardScaler, OneHotEncoder, StandardScaler,
MinMaxScaler, LabelEncoder
from sklearn.linear_model import LogisticRegression
from sklearn.ensemble import RandomForestRegressor, RandomForestClassifier
from sklearn.metrics import accuracy_score, recall_score, precision_score, f1_score,
confusion_matrix, classification_report, roc_auc_score, roc_curve

df = pd.read_csv('MCI.csv', encoding='utf-8-sig')

print('\n')
#remove rows with null values.. there are 100 of null values as we saw in exploratory
analysis
df = df.dropna() #remove trailing spaces
df.columns = df.columns.str.strip() #For column names
df.columns = [col.strip() for col in df.columns] #For data in each column
del df["X"]
del df["Y"]
del df["Index_"]
del df["event_unique_id"]
del df["Division"]
del df["occurrencedate"]
del df["reporteddate"]
del df["ucr_code"]
del df["ucr_ext"]
del df["reporteddayofyear"]
del df["occurrencedayofyear"]
del df["Hood_ID"]
del df["Longitude"]
del df["Latitude"]
del df["ObjectId"]
del df["Neighbourhood"]
del df["location_type"]
#convert the YEARS column such as 'reportedyear' and occurrenceyear column to int
format
df[ 'reportedyear']=df[ 'reportedyear'].astype(int)
df['occurrenceyear']=df['occurrenceyear'].astype(int)
print('\n')
# one hot encoding
print("################################## DF_PREMISE - encode premise type")
```

```python
df_lr = pd.get_dummies(df, drop_first=False) #logisic regression
df_premise=pd.get_dummies(df['premises_type']) #encoding prermise type
df_premise=pd.concat([df, df_premise], axis=1) #adding the df_premise to df
#change the df_premise to int
df_premise['Apartment']=df_premise['Apartment'].astype(int)
df_premise['Commercial']=df_premise['Commercial'].astype(int)
df_premise['Educational']=df_premise['Educational'].astype(int)
df_premise['House']=df_premise['House'].astype(int)
df_premise['Other']=df_premise['Other'].astype(int)
df_premise['Outside']=df_premise['Outside'].astype(int)
df_premise['Transit']=df_premise['Transit'].astype(int)
print("printing to confirm df_premise")
df_premise=pd.concat([df, df_premise], axis=1) #adding the df_premises to df
print(df_premise)
print('\n')

################################# MCI_CATEGORY - encode mci_category hot encoding
print("################################# MCI_CATEGORY - encode mci_category ")
df_dummy=pd.get_dummies(df['mci_category']) #encode mci in df_dummy df
df1=pd.concat([df_premise, df_dummy], axis=1) #adding the df_dummy to df_premise
df1 = df1.dropna()
#changing each mci type to INT for one hot encoding
df1['Assault']=df1['Assault'].astype(int)
df1['Auto Theft']=df1['Auto Theft'].astype(int)
df1['Break and Enter']=df1['Break and Enter'].astype(int)
df1['Robbery']=df1['Robbery'].astype(int)
df1['Theft Over']=df1['Theft Over'].astype(int)
print("printing to see df1 with INT encoding")
#creating df2 to remove all duplicates from df1
df2 = df1.loc[:,~df1.columns.duplicated()]

#deleting object columns since MCI & premises type are encoded
del df2["premises_type"]
del df2["offence"]
del df2["mci_category"]
del df2["reportedyear"]
del df2["reportedmonth"]
del df2["reportedday"]
del df2["reporteddayofweek"]
del df2["reportedhour"]
# display updated DataFrame
print(df2.columns)
print('\n')
print(df2.info())
print('\n')
#convert months to a num
print("convert months to a num for df2")
print("lets view the ORIGNAL FORMAT of occurrencemonth")
print(df2.occurrencemonth.unique())
mon = {'January':1, 'February':2, 'March':3, 'April':4, 'May':5, 'June':6, 'July':7,
'August':8, 'September':9, 'October':10, 'November':11, 'December':12 }
df2.occurrencemonth = df2.occurrencemonth.map(mon)
print('\n')
print("lets view the CHANGES of occurrencemonth...SHOULD SHOW AS INT from 1 to 12")
print(df2.occurrencemonth.unique())
print(df2.head())

print('\n')
#convert days of week to a num
print(df2.occurrencedayofweek.unique()) #To view unique
dow = {'Monday    ':1, 'Tuesday   ':2, 'Wednesday ':3, 'Thursday  ':4, 'Friday    ':5,
'Saturday  ':6, 'Sunday    ':7, }
df2.occurrencedayofweek = df2.occurrencedayofweek.map(dow)
print("convert day of week (dow) to a num")
#convert day of week to int
df2['occurrencedayofweek']=df2['occurrencedayofweek'].astype(int)
print(df2.occurrencedayofweek.unique())
print(df2.head())
print(df2.info())




#############################################RANDOM FOREST
print("DFLR")
df_lr = pd.get_dummies(df2, drop_first=False)
print(df_lr.shape)
print('\n')

print(df_lr.head())
print('\n')

print(df_lr.info())
print('\n')

df_tr = df2.apply(LabelEncoder().fit_transform)
print(df_tr.head())
print('\n')

print("##############################################FOR 'ASSAULT' MCI CATEGORY
###################################")
#setting 'assault' category as the target
target="Assault"
y=df2[target].values
# remove the target and independent variables
feature_col_tr=df_tr.columns.to_list()
```

```python
feature_col_tr.remove(target)

acc_RF=[]

# use a stratified 3 splits for the k-fold validation to check accuracy of model
kf=StratifiedKFold(n_splits=3)
for fold , (trn_,val_) in enumerate(kf.split(X=df_tr,y=y)):
    # next step is to split the dataset to keep portion of data for training and
portion for validation
    Xtr_train=df_tr.loc[trn_,feature_col_tr]
    ytr_train=df_tr.loc[trn_,target]
    Xtr_valid=df_tr.loc[val_,feature_col_tr]
    ytr_valid=df_tr.loc[val_,target]
    # fitting the random forest model
    clf_2=RandomForestClassifier(n_estimators=8,criterion="entropy")
    clf_2.fit(Xtr_train,ytr_train)
    # predict the classifier
    ytr_pred=clf_2.predict(Xtr_valid)
    # to print results for the classification and accuracy report
    print(f"FOLD: {fold+1} ")
    print(classification_report(ytr_valid,ytr_pred))
    acc=roc_auc_score(ytr_valid,ytr_pred)
    acc_RF.append(acc)
    print(f"The accuracy for fold is {fold+1} : {acc}\n")
print('\n')
```

```
The default interactive shell is now zsh.
To update your account to use zsh, please run `chsh -s /bin/zsh`.
For more details, please visit https://support.apple.com/kb/HT208050.
Royas-MacBook:pandasproject royasalehzai$ cd
/Users/royasalehzai/studysession/pandasproject
/usr/local/bin/python3 /Users/royasalehzai/studysession/pandasproject/RandomForest.py
Royas-MacBook:pandasproject royasalehzai$ /usr/local/bin/python3
/Users/royasalehzai/studysession/pandasproject/RandomForest.py
```

```
#################################### DF_PREMISE - encode premise type
printing to confirm df_premise
     premises_type          offence  reportedyear reportedmonth  reportedday  ... Educational
House  Other Outside  Transit
0       Apartment          Assault       2014      January         3  ...        0     0     0     0     0
1          House           B&E       2014      January         3  ...        0     1     0     0     0
```

```
2         Outside            Assault       2014     January        3  ...        0     0     0     1      0
3       Commercial          Theft Over       2014     January        3  ...        0     0     0     0
0
4       Commercial     Robbery - Business       2014     January        3  ...        0     0     0     0
0
...           ...                 ...         ...         ...        ...  ...       ...   ...   ...    ...    ...
299823       House  Theft Of Motor Vehicle       2022       June        29  ...        0     1     0
0      0
299824     Outside  Theft Of Motor Vehicle       2022       June        29  ...        0     0     0
1      0
299825   Commercial  Theft Of Motor Vehicle       2022       June        29  ...        0     0     0
0      0
299826     Outside  Theft Of Motor Vehicle       2022       June        29  ...        0     0     0
1      0
299827     Outside  Theft Of Motor Vehicle       2022       June        29  ...        0     0     0
1      0

[299828 rows x 33 columns]


################################## MCI_CATEGORY - encode mci_category
printing to see df1 with INT encoding
Index(['occurrenceyear', 'occurrencemonth', 'occurrenceday',
       'occurrencedayofweek', 'occurrencehour', 'Apartment', 'Commercial',
       'Educational', 'House', 'Other', 'Outside', 'Transit', 'Assault',
       'Auto Theft', 'Break and Enter', 'Robbery', 'Theft Over'],
      dtype='object')


<class 'pandas.core.frame.DataFrame'>
Int64Index: 299828 entries, 0 to 299827
Data columns (total 17 columns):
 #   Column               Non-Null Count   Dtype
---  ------               --------------   -----
 0   occurrenceyear       299828 non-null  int64
 1   occurrencemonth      299828 non-null  object
 2   occurrenceday        299828 non-null  float64
 3   occurrencedayofweek  299828 non-null  object
 4   occurrencehour       299828 non-null  int64
 5   Apartment            299828 non-null  int64
 6   Commercial           299828 non-null  int64
 7   Educational          299828 non-null  int64
 8   House                299828 non-null  int64
 9   Other                299828 non-null  int64
 10  Outside             299828 non-null  int64
 11  Transit             299828 non-null  int64
 12  Assault             299828 non-null  int64
 13  Auto Theft          299828 non-null  int64
 14  Break and Enter     299828 non-null  int64
 15  Robbery             299828 non-null  int64
 16  Theft Over          299828 non-null  int64
dtypes: float64(1), int64(14), object(2)
memory usage: 41.2+ MB
None


convert months to a num for df2
lets view the ORIGNAL FORMAT of occurrencemonth
['January' 'February' 'March' 'April' 'May' 'June' 'July' 'August'
 'September' 'October' 'November' 'December']
/Users/royasalehzai/studysession/pandasproject/RandomForest.py:95:
SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  df2.occurrencemonth = df2.occurrencemonth.map(mon)


lets view the CHANGES of occurrencemonth...SHOULD SHOW AS INT from 1 to 12
[ 1  2  3  4  5  6  7  8  9 10 11 12]
   occurrenceyear  occurrencemonth  occurrenceday occurrencedayofweek  occurrencehour  ...
Assault  Auto Theft  Break and Enter  Robbery  Theft Over
0          2014              1          3.0        Friday              11  ...       1         0             0       0
0
1          2014              1          3.0        Friday              14  ...       0         0             1       0
0
2          2014              1          3.0        Friday              13  ...       1         0             0       0
0
3          2014              1          3.0        Friday              12  ...       0         0             0       0
1
4          2014              1          3.0        Friday              14  ...       0         0             0       1
0

[5 rows x 17 columns]
```

```
['Friday    ' 'Thursday  ' 'Saturday  ' 'Wednesday ' 'Sunday    '
 'Monday    ' 'Tuesday   ']
/Users/royasalehzai/studysession/pandasproject/RandomForest.py:105:
SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  df2.occurrencedayofweek = df2.occurrencedayofweek.map(dow)
convert day of week (dow) to a num
/Users/royasalehzai/studysession/pandasproject/RandomForest.py:108:
SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  df2['occurrencedayofweek']=df2['occurrencedayofweek'].astype(int)
[5 4 6 3 7 1 2]
   occurrenceyear  occurrencemonth  occurrenceday  occurrencedayofweek  occurrencehour  ...
Assault  Auto Theft  Break and Enter  Robbery  Theft Over
0          2014                1            3.0                    5              11  ...        1           0                0        0           0
1          2014                1            3.0                    5              14  ...        0           0                1        0           0
2          2014                1            3.0                    5              13  ...        1           0                0        0           0
3          2014                1            3.0                    5              12  ...        0           0                0        0           1
4          2014                1            3.0                    5              14  ...        0           0                0        1           0

[5 rows x 17 columns]
<class 'pandas.core.frame.DataFrame'>
Int64Index: 299828 entries, 0 to 299827
Data columns (total 17 columns):
 #   Column               Non-Null Count   Dtype
---  ------               --------------   -----
 0   occurrenceyear       299828 non-null  int64
 1   occurrencemonth      299828 non-null  int64
 2   occurrenceday        299828 non-null  float64
 3   occurrencedayofweek  299828 non-null  int64
 4   occurrencehour       299828 non-null  int64
 5   Apartment            299828 non-null  int64
 6   Commercial           299828 non-null  int64
 7   Educational          299828 non-null  int64
 8   House                299828 non-null  int64
 9   Other                299828 non-null  int64
 10  Outside             299828 non-null  int64
 11  Transit             299828 non-null  int64
 12  Assault             299828 non-null  int64
 13  Auto Theft          299828 non-null  int64
 14  Break and Enter     299828 non-null  int64
 15  Robbery             299828 non-null  int64
 16  Theft Over          299828 non-null  int64
dtypes: float64(1), int64(16)
memory usage: 41.2 MB
None
DFLR
(299828, 17)


   occurrenceyear  occurrencemonth  occurrenceday  occurrencedayofweek  occurrencehour  ...  Assault  Auto Theft  Break and Enter  Robbery  Theft Over
0            2014                1            3.0                    5              11  ...        1           0                0        0           0
1            2014                1            3.0                    5              14  ...        0           0                1        0           0
2            2014                1            3.0                    5              13  ...        1           0                0        0           0
3            2014                1            3.0                    5              12  ...        0           0                0        0           1
4            2014                1            3.0                    5              14  ...        0           0                0        1           0

[5 rows x 17 columns]


<class 'pandas.core.frame.DataFrame'>
Int64Index: 299828 entries, 0 to 299827
Data columns (total 17 columns):
 #   Column               Non-Null Count   Dtype
---  ------               --------------   -----
 0   occurrenceyear       299828 non-null  int64
 1   occurrencemonth      299828 non-null  int64
 2   occurrenceday        299828 non-null  float64
 3   occurrencedayofweek  299828 non-null  int64
 4   occurrencehour       299828 non-null  int64
 5   Apartment            299828 non-null  int64
 6   Commercial           299828 non-null  int64
 7   Educational          299828 non-null  int64
 8   House                299828 non-null  int64
 9   Other                299828 non-null  int64
 10  Outside              299828 non-null  int64
 11  Transit              299828 non-null  int64
 12  Assault              299828 non-null  int64
 13  Auto Theft           299828 non-null  int64
 14  Break and Enter      299828 non-null  int64
 15  Robbery              299828 non-null  int64
 16  Theft Over           299828 non-null  int64
dtypes: float64(1), int64(16)
memory usage: 41.2 MB
None


   occurrenceyear  occurrencemonth  occurrenceday  occurrencedayofweek  occurrencehour  ...
Assault  Auto Theft  Break and Enter  Robbery  Theft Over
0             0              0             2               4            11  ...       1         0              0       0          0
1             0              0             2               4            14  ...       0         0              1       0          0
2             0              0             2               4            13  ...       1         0              0       0          0
3             0              0             2               4            12  ...       0         0              0       0          1
4             0              0             2               4            14  ...       0         0              0       1          0

[5 rows x 17 columns]


################################################FOR 'ASSAULT' MCI CATEGORY
####################################
FOLD: 1
              precision    recall  f1-score   support

           0       1.00      1.00      1.00     46359
           1       1.00      1.00      1.00     53584

    accuracy                           1.00     99943
   macro avg       1.00      1.00      1.00     99943
weighted avg       1.00      1.00      1.00     99943

The accuracy for fold is 1 : 1.0

FOLD: 2
              precision    recall  f1-score   support

           0       1.00      1.00      1.00     46359
           1       1.00      1.00      1.00     53584

    accuracy                           1.00     99943
   macro avg       1.00      1.00      1.00     99943
weighted avg       1.00      1.00      1.00     99943

The accuracy for fold is 2 : 0.9999626754255002
```

```
FOLD: 3
           precision    recall  f1-score   support

           0       1.00      1.00      1.00     46359
           1       1.00      1.00      1.00     53583

    accuracy                           1.00     99942
   macro avg       1.00      1.00      1.00     99942
weighted avg       1.00      1.00      1.00     99942

The accuracy for fold is 3 : 1.0



Royas-MacBook:pandasproject royasalehzai$
```