```
import pandas as pd
import numpy as np
from statsmodels.formula.api import ols
from sklearn.preprocessing import StandardScaler, OneHotEncoder, StandardScaler,
MinMaxScaler, LabelEncoder


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

#remove rows with null values.. there are 100 of null values as we saw in exploratory
analysis
df = df.dropna()
print("Viewing all columns")
print('\n')
#emove rows with null values.. there are 100 of null values as we saw in exploratory
analysis
df = df.dropna()
#remove trailing spaces & delete columns not needed
df.columns = df.columns.str.strip() #For column names
df.columns = [col.strip() for col in df.columns] #For data in each column
print('\n')
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"]
print(df.info()) #to confirm its deleted for null values
print('\n')


###########################################MULTIPLE REGRESSION MCI
CATEGORY##########################


#transform categories as int
df['mci_category'] =df['mci_category'].astype('category')
df['mci_category'] =df['mci_category'].cat.codes
df['occurrencemonth'] =df['occurrencemonth'].astype('category')
```

```
df['occurrencemonth'] =df['occurrencemonth'].cat.codes
df['occurrencedayofweek'] =df['occurrencedayofweek'].astype('category')
df['occurrencedayofweek'] =df['occurrencedayofweek'].cat.codes
df['premises_type'] =df['premises_type'].astype('category')
df['premises_type'] =df['premises_type'].cat.codes
print("checking transformed-category")
print(df.info())
print(df)
print(df.isnull().sum()) #check null nums

print('\n')
print("label encoder")
#label encoder
df = df.apply(LabelEncoder().fit_transform)
print(df.head())
print('\n')


print('\n')
#use multiple categories for multi regression to predict what time 'occurencetime'
based on categories
#simple regression  y = mx + c
#multiple linear regression x1,x2,x3...xn  & m1,m2,m3...mn
            # y = m1x1 + m2x2 + m3x3 + m4x4 ...+mnXn + c



print('\n')
print("#####################################Multiple regression based on Occurance
Hour")
########################################Multiple regression based on Occurance HOUR
X = df.drop(columns = 'occurrencehour')
 #dropping occurence hour as we will compare this to others
print(X)
y = df['occurrencehour']
from sklearn.model_selection import train_test_split
#next step is to split the dataset to keep portion of data for training and portion
for testing
#keeps 30% for  testing and 70% for training
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3,
random_state=0)
#random state creates same test train if necesary
from sklearn.linear_model import LinearRegression
lr = LinearRegression()
#lets fit our training data into our linear regression
lr.fit(X_train, y_train) #the model should be trained
#lets view the parametere
print('\n')
print("the y intercept:")
c = lr.intercept_ #this is the y intercept
print(c)
print('\n')
print("The coefficients for each column in training features:")
m = lr.coef_ #the coefficient
print(m)
print('\n')
#time to test the model training
y_pred_train = lr.predict(X_train)
print(y_pred_train)
import matplotlib.pyplot as plt
plt.scatter(y_train, y_pred_train)
plt.xlabel("Actual MCI Occurence Time")
plt.ylabel("Predicted MCI Occurence Time")
plt.show()
print('\n')
#now to predict accuracy .. use r2 score
print("The accuracy of r2_score:")
from sklearn.metrics import r2_score
print(r2_score(y_train, y_pred_train))
print('\n')
print('df.info')
print(df.info())


print('\n')
print("#####################################Multiple regression based on Occurance Day of Week")
########################################Multiple regression based on Occurance Day of Week
X = df.drop(columns = 'occurrencedayofweek')
print(X)
y = df['occurrencedayofweek']
from sklearn.model_selection import train_test_split
#next step is to split the dataset to keep portion of data for training and portion for testing
#keeps 30% for  testing and 70% for training
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=0)
#random state creates same test train if necesary
from sklearn.linear_model import LinearRegression
lr = LinearRegression()
#lets fit our training data into our linear regression
lr.fit(X_train, y_train) #the model should be trained
#lets view the parametere
print('\n')
print("the y intercept:")
```

```
c = lr.intercept_ #this is the y intercept
print(c)
print('\n')
print("The coefficients for each column:")
m = lr.coef_ #the coefficient
print(m)
print('\n')
#time to test the model training
y_pred_train = lr.predict(X_train)
print(y_pred_train)
import matplotlib.pyplot as plt
plt.scatter(y_train, y_pred_train)
plt.xlabel("Actual MCI Occurence Time")
plt.ylabel("Predicted MCI Occurence Time")
plt.show()
print('\n')
#now to predict accuracy .. use r2 score
print("The accuracy of r2_score:")
from sklearn.metrics import r2_score
print(r2_score(y_train, y_pred_train))
print('\n')
print('df.info')
print(df.info())
```

```
print('\n')
print("#####################################Multiple regression based on Occurance
MONTH")
########################################Multiple regression based on Occurance MONTH
X = df.drop(columns = 'occurrencemonth')
print(X)
y = df['occurrencemonth']
from sklearn.model_selection import train_test_split
#next step is to split the dataset to keep portion of data for training and portion
for testing
#keeps 30% for  testing and 70% for training
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3,
random_state=0)
#random state creates same test train if necesary
from sklearn.linear_model import LinearRegression
lr = LinearRegression()
#lets fit our training data into our linear regression
lr.fit(X_train, y_train) #the model should be trained
#lets view the parametere
```

```
print('\n')
print("the y intercept:")
c = lr.intercept_ #this is the y intercept
print(c)
print('\n')
print("The coefficients for each column:")
m = lr.coef_ #the coefficient
print(m)
print('\n')
#time to test the model training
y_pred_train = lr.predict(X_train)
print(y_pred_train)
import matplotlib.pyplot as plt
plt.scatter(y_train, y_pred_train)
plt.xlabel("Actual MCI Occurence Time")
plt.ylabel("Predicted MCI Occurence Time")
plt.show()
print('\n')
#now to predict accuracy .. use r2 score
print("The accuracy of r2_score:")
from sklearn.metrics import r2_score
print(r2_score(y_train, y_pred_train))
print('\n')
print('df.info')
print(df.info())
```

```
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.
ndasprojectok:pandasproject royasalehzai$ cd /Users/royasalehzai/studysession/pa
/usr/local/bin/python3 /Users/royasalehzai/studysession/pandasproject/Multipleregression.py
Royas-MacBook:pandasproject royasalehzai$ /usr/local/bin/python3
/Users/royasalehzai/studysession/pandasproject/Multipleregression.py
Viewing all columns




<class 'pandas.core.frame.DataFrame'>
Int64Index: 299828 entries, 0 to 299827
Data columns (total 15 columns):
 #   Column              Non-Null Count   Dtype 
---  ------              --------------   ----- 
 0   location_type       299828 non-null  object
 1   premises_type       299828 non-null  object
 2   offence             299828 non-null  object
 3   reportedyear        299828 non-null  int64
 4   reportedmonth       299828 non-null  object
 5   reportedday         299828 non-null  int64
 6   reporteddayofweek   299828 non-null  object
 7   reportedhour        299828 non-null  int64
 8   occurrenceyear      299828 non-null  float64
 9   occurrencemonth     299828 non-null  object
 10  occurrenceday       299828 non-null  float64
 11  occurrencedayofweek  299828 non-null  object
 12  occurrencehour      299828 non-null  int64
 13  mci_category        299828 non-null  object
 14  Neighbourhood       299828 non-null  object
dtypes: float64(2), int64(4), object(9)
memory usage: 36.6+ MB
None


checking transformed-category
<class 'pandas.core.frame.DataFrame'>
Int64Index: 299828 entries, 0 to 299827
Data columns (total 15 columns):
 #   Column              Non-Null Count   Dtype
---  ------              --------------   -----
 0   location_type       299828 non-null  object
 1   premises_type       299828 non-null  int8
 2   offence             299828 non-null  object
 3   reportedyear        299828 non-null  int64
 4   reportedmonth       299828 non-null  object
 5   reportedday         299828 non-null  int64
 6   reporteddayofweek   299828 non-null  object
 7   reportedhour        299828 non-null  int64
 8   occurrenceyear      299828 non-null  float64
 9   occurrencemonth     299828 non-null  int8
 10  occurrenceday       299828 non-null  float64
 11  occurrencedayofweek  299828 non-null  int8
 12  occurrencehour      299828 non-null  int64
 13  mci_category        299828 non-null  int8
 14  Neighbourhood       299828 non-null  object
dtypes: float64(2), int64(4), int8(4), object(5)
memory usage: 28.6+ MB
None
                              location_type  premises_type              offence  ...  occurrencehour
mci_category             Neighbourhood
```

```
0                  Apartment (Rooming House, Condo)           0              Assault  ...           11
0    York University Heights
1      Single Home, House (Attach Garage, Cottage, Mo...            3                 B&E  ...           14
2                 Malvern
2                  Open Areas (Lakes, Parks, Rivers)           5              Assault  ...           13          0
Long Branch
3      Other Commercial / Corporate Places (For Profi...            1           Theft Over  ...           12
4         Thorncliffe Park
4                            Convenience Stores           1     Robbery - Business  ...           14          3
Islington-City Centre West
...                                    ...          ...                  ...  ...           ...         ...                    ...
299823  Single Home, House (Attach Garage, Cottage, Mo...            3  Theft Of Motor Vehicle  ...
20          1        Westminster-Branson
299824  Parking Lots (Apt., Commercial Or Non-Commercial)           5  Theft Of Motor Vehicle
...             21          1                   Woburn
299825  Other Commercial / Corporate Places (For Profi...           1  Theft Of Motor Vehicle  ...
12          1              Dorset Park
299826  Parking Lots (Apt., Commercial Or Non-Commercial)           5  Theft Of Motor Vehicle
...              0          1                      NSA
299827  Parking Lots (Apt., Commercial Or Non-Commercial)           5  Theft Of Motor Vehicle
...             16          1               Humbermede

[299828 rows x 15 columns]
location_type         0
premises_type         0
offence               0
reportedyear          0
reportedmonth         0
reportedday           0
reporteddayofweek     0
reportedhour          0
occurrenceyear        0
occurrencemonth       0
occurrenceday         0
occurrencedayofweek   0
occurrencehour        0
mci_category          0
Neighbourhood         0
dtype: int64


label encoder
   location_type  premises_type  offence  reportedyear  reportedmonth  ...  occurrenceday
occurrencedayofweek  occurrencehour  mci_category  Neighbourhood
```

```
0          0          0     5         0          4  ...          2              0          11          0
139
1         36          3     12         0          4  ...          2              0          14          2
73
2         19          5      5         0          4  ...          2              0          13          0
72
3         20          1     43         0          4  ...          2              0          12          4
119
4          7          1     25         0          4  ...          2              0          14          3
58

[5 rows x 15 columns]
```

```
######################################Multiple regression based on Occurance Hour
       location_type  premises_type  offence  reportedyear  ...  occurrenceday
occurrencedayofweek  mci_category  Neighbourhood
0               0          0      5         0 ...          2              0          0         139
1              36          3      12         0 ...          2              0          2          73
2              19          5      5         0 ...          2              0          0         72
3              20          1      43         0 ...          2              0          4         119
4               7          1      25         0 ...          2              0          3         58
...           ...          ...    ...        ...  ...          ...              ...        ...         ...
299823         36          3      41         8 ...          27              5          1         126
299824         28          5      41         8 ...          27              5          1         133
299825         20          1      41         8 ...          19              0          1          31
299826         28          5      41         8 ...          28              6          1          84
299827         28          5      41         8 ...          28              6          1          55

[299828 rows x 14 columns]


the y intercept:
3.3255473086911884


The coefficients for each column in training features:
[ 2.06605079e-02  1.21568656e-02  8.13821765e-02 -1.99058896e+00
 -1.87772116e-03 -4.36389129e-02 -6.69849246e-03  6.37570358e-01
  1.99749493e+00  1.13301130e-02  4.96942075e-02  1.75591201e-02
 -7.52336395e-01 -7.81599613e-04]


[ 8.47147269 12.45449335 15.11376674 ...  3.92857599 16.38010845
 16.22375506]


The accuracy of r2_score:
0.3502956551301798


df.info
<class 'pandas.core.frame.DataFrame'>
Int64Index: 299828 entries, 0 to 299827
Data columns (total 15 columns):
 #   Column               Non-Null Count   Dtype
---  ------               --------------   -----
 0   location_type        299828 non-null  int64
 1   premises_type        299828 non-null  int64
 2   offence              299828 non-null  int64
 3   reportedyear         299828 non-null  int64
 4   reportedmonth        299828 non-null  int64
 5   reportedday          299828 non-null  int64
 6   reporteddayofweek    299828 non-null  int64
 7   reportedhour         299828 non-null  int64
 8   occurrenceyear       299828 non-null  int64
 9   occurrencemonth      299828 non-null  int64
 10  occurrenceday        299828 non-null  int64
 11  occurrencedayofweek  299828 non-null  int64
 12  occurrencehour       299828 non-null  int64
 13  mci_category         299828 non-null  int64
 14  Neighbourhood        299828 non-null  int64
dtypes: int64(15)
memory usage: 36.6 MB
None


######################################Multiple regression based on Occurance Day of
Week
    location_type  premises_type  offence  reportedyear  reportedmonth  ... occurrencemonth
occurrenceday  occurrencehour  mci_category  Neighbourhood
0              0              0       5            0              4  ...              4           2           11           0
139
```

```
1              36          3     12         0          4  ...          4          2          14          2
73
2              19          5      5         0          4  ...          4          2          13          0
72
3              20          1     43         0          4  ...          4          2          12          4
119
4               7          1     25         0          4  ...          4          2          14          3
58
...           ...        ...    ...       ...        ...  ...        ...        ...         ...        ...        ...
299823         36          3     41         8          6  ...          6         27          20          1
126
299824         28          5     41         8          6  ...          6         27          21          1
133
299825         20          1     41         8          6  ...          8         19          12          1
31
299826         28          5     41         8          6  ...          6         28           0          1
84
299827         28          5     41         8          6  ...          6         28          16          1
55

[299828 rows x 14 columns]
```

```
the y intercept:
0.88538424586342
```

```
The coefficients for each column:
[ 1.68347285e-03 -1.02059820e-02  1.46083431e-03  6.12307348e-03
 -8.64037243e-04 -1.73305080e-04  6.89679066e-01 -8.40940493e-04
 -4.94973958e-03  4.61306966e-04  7.62564672e-05  1.07846217e-03
 -2.20307665e-02 -8.75804874e-05]
```

```
[5.00632754 4.40786301 1.61274746 ... 1.55828323 0.9259268  4.30441155]
```

```
The accuracy of r2_score:
0.4801737048288207
```

```
df.info
<class 'pandas.core.frame.DataFrame'>
Int64Index: 299828 entries, 0 to 299827
```

```
Data columns (total 15 columns):
 #   Column              Non-Null Count   Dtype
---  ------              --------------   -----
 0   location_type       299828 non-null  int64
 1   premises_type       299828 non-null  int64
 2   offence             299828 non-null  int64
 3   reportedyear        299828 non-null  int64
 4   reportedmonth       299828 non-null  int64
 5   reportedday         299828 non-null  int64
 6   reporteddayofweek   299828 non-null  int64
 7   reportedhour        299828 non-null  int64
 8   occurrenceyear      299828 non-null  int64
 9   occurrencemonth     299828 non-null  int64
 10  occurrenceday       299828 non-null  int64
 11  occurrencedayofweek  299828 non-null  int64
 12  occurrencehour      299828 non-null  int64
 13  mci_category        299828 non-null  int64
 14  Neighbourhood       299828 non-null  int64
dtypes: int64(15)
memory usage: 36.6 MB
None


######################################Multiple regression based on Occurance MONTH
        location_type  premises_type  offence  reportedyear  ...  occurrencedayofweek  occurrencehour  mci_category  Neighbourhood
0                   0              0        5             0  ...                    0              11             0            139
1                  36              3       12             0  ...                    0              14             2             73
2                  19              5        5             0  ...                    0              13             0             72
3                  20              1       43             0  ...                    0              12             4            119
4                   7              1       25             0  ...                    0              14             3             58
...               ...            ...      ...           ...  ...                  ...             ...           ...            ...
299823             36              3       41             8  ...                    5              20             1            126
299824             28              5       41             8  ...                    5              21             1            133
299825             20              1       41             8  ...                    0              12             1             31
299826             28              5       41             8  ...                    6               0             1             84
299827             28              5       41             8  ...                    6              16             1             55

[299828 rows x 14 columns]


the y intercept:
0.40653640169361704
```

```
The coefficients for each column:
[ 3.26224601e-04  5.07914723e-03 -1.68038304e-04  8.91230125e-02
  9.24328018e-01 -2.42394247e-03 -4.40774319e-04 -7.24399578e-04
 -9.29520496e-02  3.01585858e-03  3.74949834e-04  5.65613229e-04
  4.99032719e-05 -6.82781762e-05]


[ 7.78661714 10.58013059  7.83639105 ... 10.56682295  5.97920246
  5.01020169]


The accuracy of r2_score:
0.8553384791891867


df.info
<class 'pandas.core.frame.DataFrame'>
Int64Index: 299828 entries, 0 to 299827
Data columns (total 15 columns):
 #   Column              Non-Null Count   Dtype
---  ------              --------------   -----
 0   location_type       299828 non-null  int64
 1   premises_type       299828 non-null  int64
 2   offence             299828 non-null  int64
 3   reportedyear        299828 non-null  int64
 4   reportedmonth       299828 non-null  int64
 5   reportedday         299828 non-null  int64
 6   reporteddayofweek   299828 non-null  int64
 7   reportedhour        299828 non-null  int64
 8   occurrenceyear      299828 non-null  int64
 9   occurrencemonth     299828 non-null  int64
 10  occurrenceday       299828 non-null  int64
 11  occurrencedayofweek  299828 non-null  int64
 12  occurrencehour      299828 non-null  int64
 13  mci_category        299828 non-null  int64
 14  Neighbourhood       299828 non-null  int64
dtypes: int64(15)
memory usage: 36.6 MB
None
Royas-MacBook:pandasproject royasalehzai$
```