Predicting Crimes in Colorado

•••

January 21, 2021

Overview

Crimes are intricate and interesting, and can be both predictable and unpredictable. Crimes have existed for a long time and will remain parts of society. Therefore, I am interested in predicting when crimes will take place.

I selected Burglary and Robbery as crimes indicators in the state of Colorado to discover data patterns and with the use Machine Learning Models to predict when crimes will take place based on the indicators.

Datasets were sourced from Colorado Crime Data Explorer site. While data for other crimes are available, in this project the scope was narrowed down to Burglary and Robbery crimes.

Understanding the problem

Question 1

Can we predict the when crimes will take place based on key indicators: historical crime data, incident date, type of crimes, location of crimes and number of crimes?

Question 2

Can we predict types of crimes with key indicators: historical crime data, incident date, type of crimes, location of crimes and number of crimes?

Question 3

What crimes are increasing or decreasing, at what rate and over what timeframes with a given regional?

Description of the source of data:

Datasets were available for download in comma separated values format files from the year 2016 to 2019 although the dataset from previous years were available. Datasets were pulled from a portal that collects data from the rest of the states that FBI maintains.

NIBRS:

https://crime-data-explorer.app.cloud.gov/downloads-and-docs

Dataset:

NIBRS_incidents_16_19
NIBRS_Offense_16_19

Features: DATA_YEAR INT, AGENCY_ID INT, INCIDENT_ID INT, NIBRS_MONTH_ID INT, CARGO_THEFT_FLAG VARCHAR, SUBMISSION_DATE DATE, INCIDENT_DATE INT, INCIDENT_Month VARCHAR, INCIDENT_DAY INT, REPORT_DATE_FLAG VARCHAR, INCIDENT_HOUR INT, DATA_HOME VARCHAR, ORIG_FORMAT VARCHAR, DID INT

Features: DATA_YEAR INT, OFFENSE_ID INT, INCIDENT_ID INT, OFFENSE_TYPE_ID VARCHAR, ATTEMPT_COMPLETE_FLAG VARCHAR, LOCATION_ID VARCHAR, METHOD_ENTRY_CODE VARCHAR

Description of the data exploration phase of the project:

Explored the dataset structure to uncover initial patterns, characteristics, creating a broad picture of important trends and major points to study in greater detail which took place in Tableau.

Data Exploration

Total Incidents: 54,930

Offence Breakdown:

Burglary: 53,133

Robbery: 1,797

Offence Location:

Commercial/Office Building: 2,927

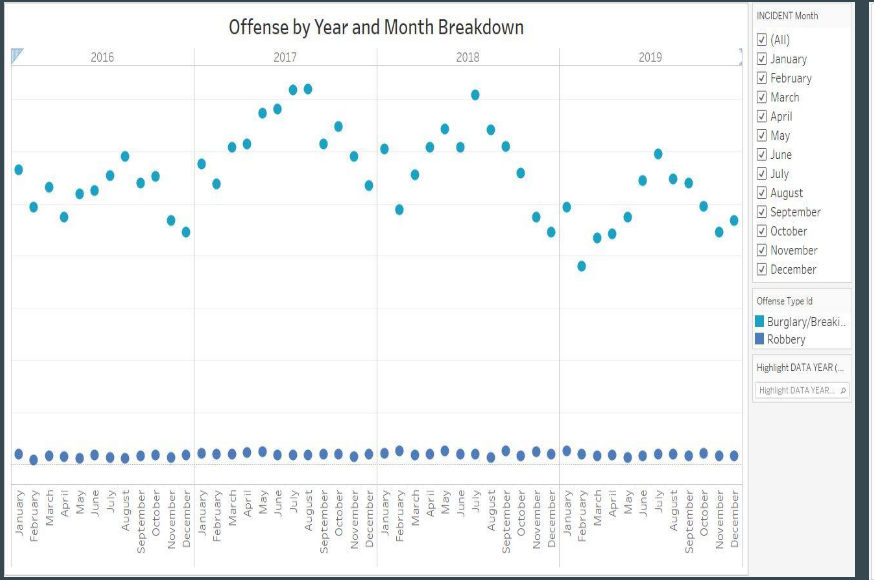
Government/Public Building: 384

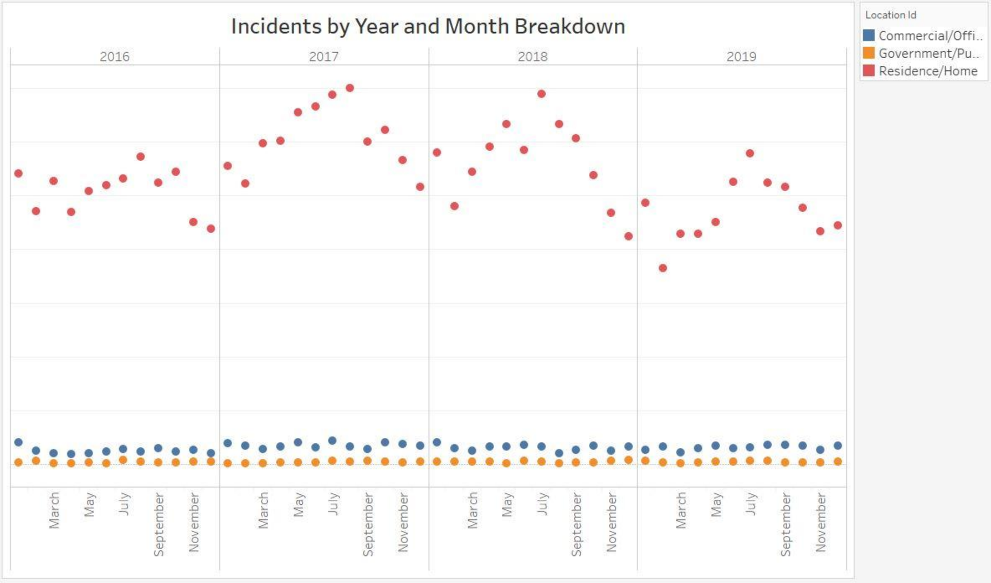
Residence/Home: 51,619

Dates:

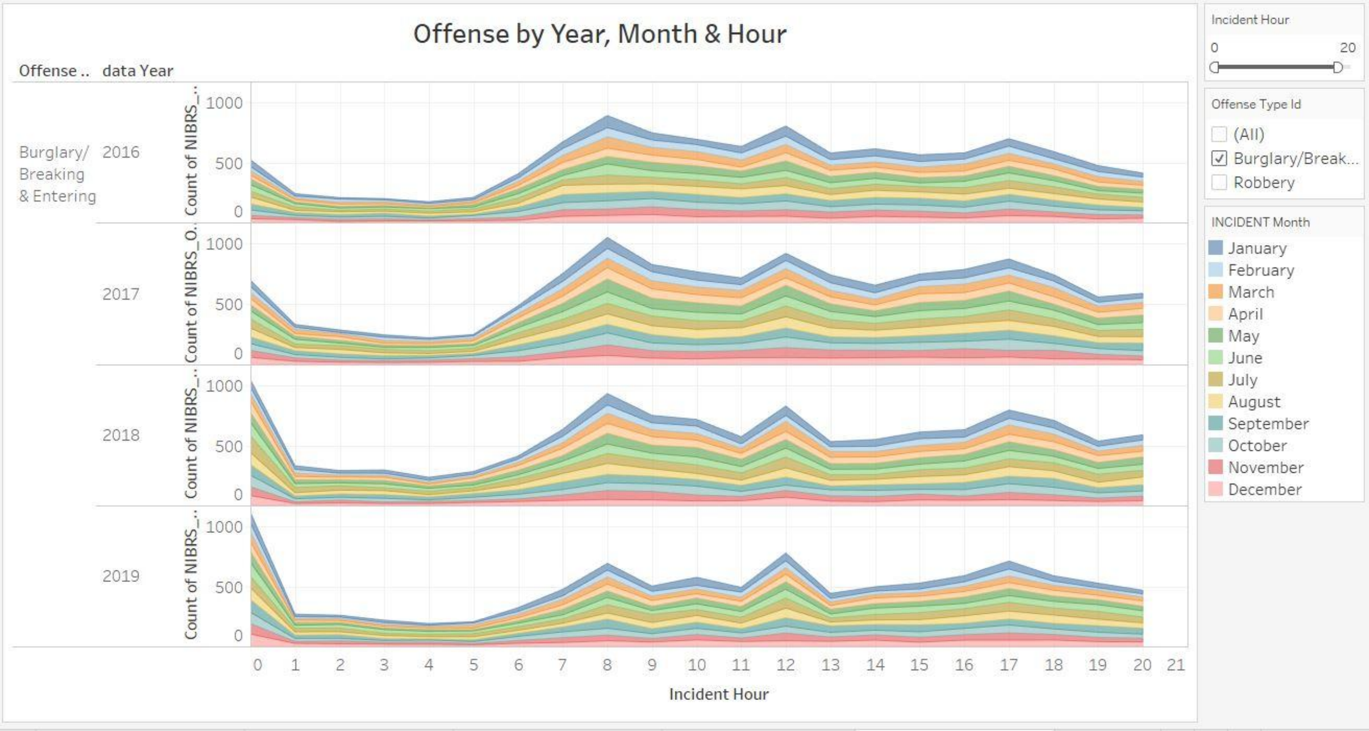
2016 - 2019

Understanding the data breakdown





Understanding the data cont...



Offense by Year, Month & Hour

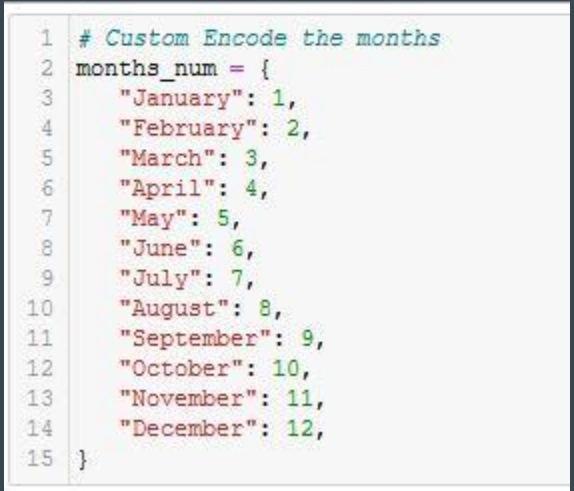
Incident Hour
0 20

Offense Type Id
□ (All)
☑ Burglary/Break...
□ Robbery

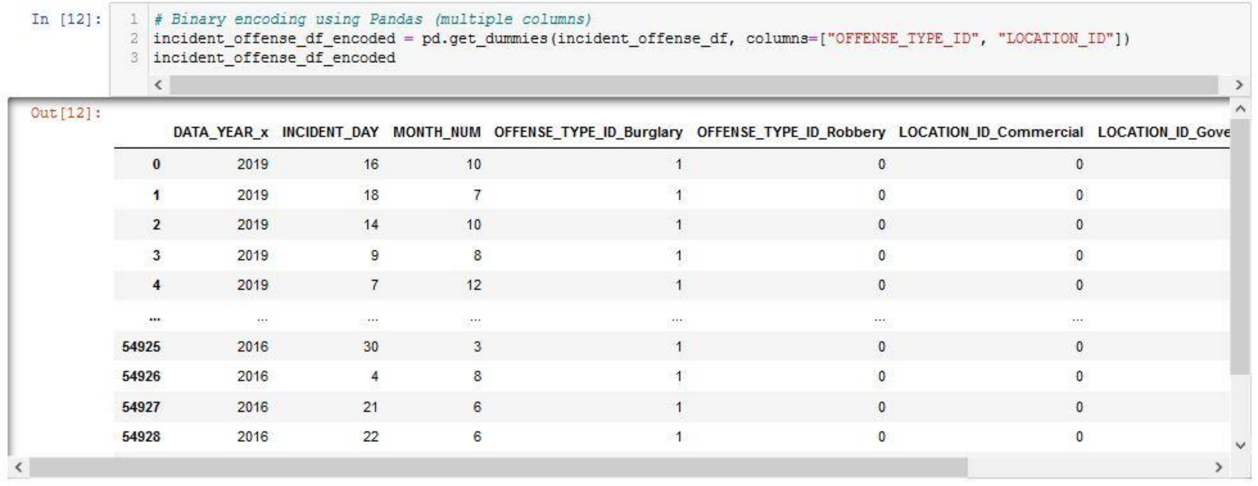
Description of the analysis phase of the project:

Analysis phase consist of preprocessing and predicting crimes with SciKitLearn Machine Learning (NL) library using to create a classifiers especifically Logistic Regression and Support Vector Machines (SVM) models.

Preprocessing



```python
# Custom Encode the months
months_num = {
    "January": 1,
    "February": 2,
    "March": 3,
    "April": 4,
    "May": 5,
    "June": 6,
    "July": 7,
    "August": 8,
    "September": 9,
    "October": 10,
    "November": 11,
    "December": 12,
}
```



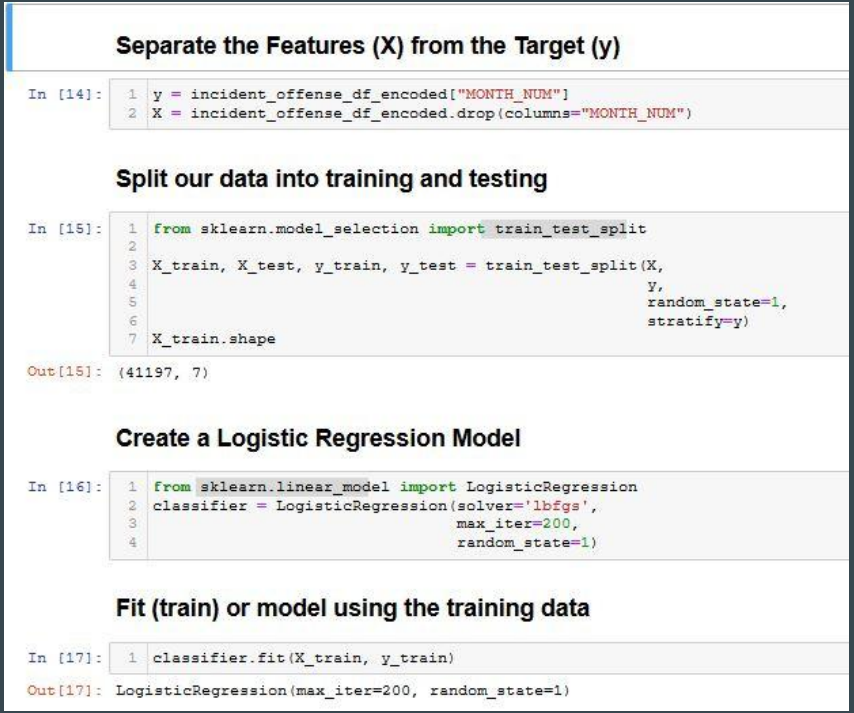
In [12]:

```python
# Binary encoding using Pandas (multiple columns)
incident_offense_df_encoded = pd.get_dummies(incident_offense_df, columns=["OFFENSE_TYPE_ID", "LOCATION_ID"])
incident_offense_df_encoded
```

Out[12]:

	DATA_YEAR_x	INCIDENT_DAY	MONTH_NUM	OFFENSE_TYPE_ID_Burglary	OFFENSE_TYPE_ID_Robbery	LOCATION_ID_Commercial	LOCATION_ID_Gove
0	2019	16	10	1	0	0	
1	2019	18	7	1	0	0	
2	2019	14	10	1	0	0	
3	2019	9	8	1	0	0	
4	2019	7	12	1	0	0	
...	...	...	...	...	...	...	
54925	2016	30	3	1	0	0	
54926	2016	4	8	1	0	0	
54927	2016	21	6	1	0	0	
54928	2016	22	6	1	0	0	

Predicting Crimes with Logistic Regression



Separate the Features (X) from the Target (y)

In [14]:

```
y = incident_offense_df_encoded["MONTH_NUM"]
X = incident_offense_df_encoded.drop(columns="MONTH_NUM")
```

Split our data into training and testing

In [15]:

```
from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(X,
                                                    y,
                                                    random_state=1,
                                                    stratify=y)
X_train.shape
```

Out[15]:

```
(41197, 7)
```

Create a Logistic Regression Model

In [16]:

```
from sklearn.linear_model import LogisticRegression
classifier = LogisticRegression(solver='lbfgs',
                                max_iter=200,
                                random_state=1)
```

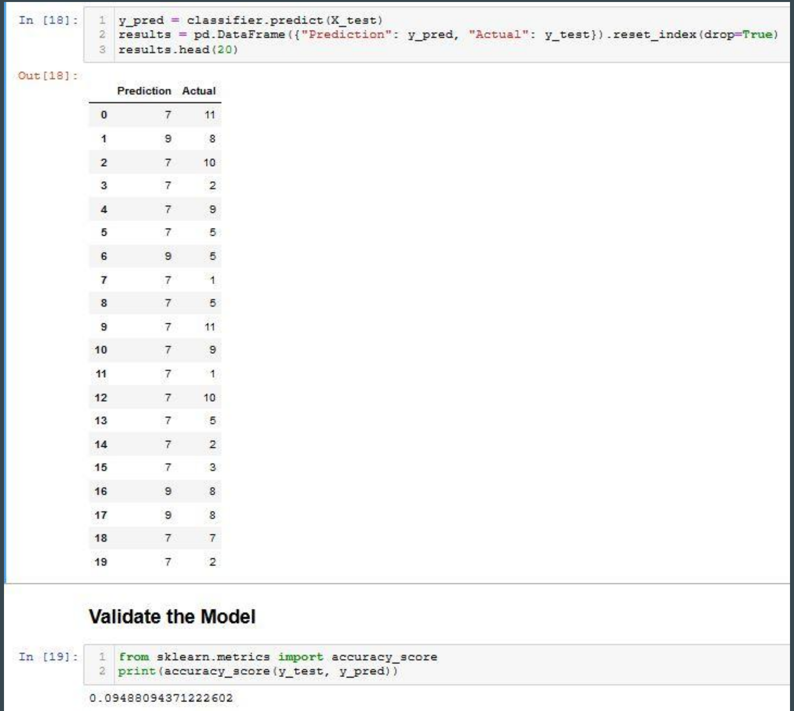
Fit (train) or model using the training data

In [17]:

```
classifier.fit(X_train, y_train)
```

Out[17]:

```
LogisticRegression(max_iter=200, random_state=1)
```



In [18]:

```
y_pred = classifier.predict(X_test)
results = pd.DataFrame({"Prediction": y_pred, "Actual": y_test}).reset_index(drop=True)
results.head(20)
```

Out[18]:

	Prediction	Actual
0	7	11
1	9	8
2	7	10
3	7	2
4	7	9
5	7	5
6	9	5
7	7	1
8	7	5
9	7	11
10	7	9
11	7	1
12	7	10
13	7	5
14	7	2
15	7	3
16	9	8
17	9	8
18	7	7
19	7	2

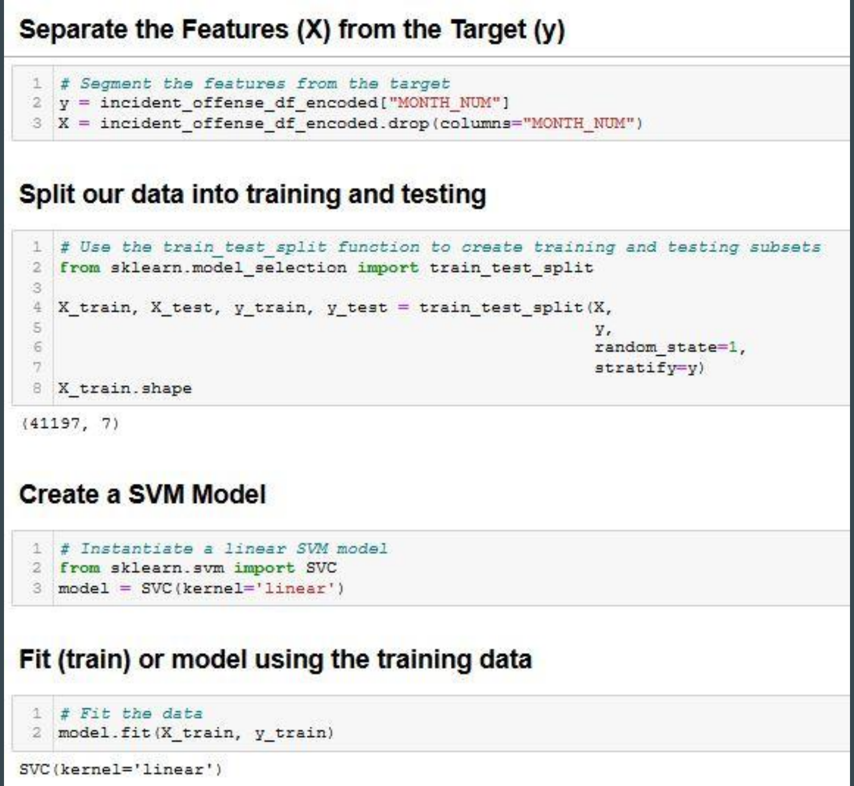
Validate the Model

In [19]:

```
from sklearn.metrics import accuracy_score
print(accuracy_score(y_test, y_pred))
```

```
0.09488094371222602
```

Predicting Crimes with SVM



Separate the Features (X) from the Target (y)

```python
# Segment the features from the target
y = incident_offense_df_encoded["MONTH_NUM"]
X = incident_offense_df_encoded.drop(columns="MONTH_NUM")
```

Split our data into training and testing

```python
# Use the train_test_split function to create training and testing subsets
from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(X,
                                                    y,
                                                    random_state=1,
                                                    stratify=y)
X_train.shape
```

```
(41197, 7)
```

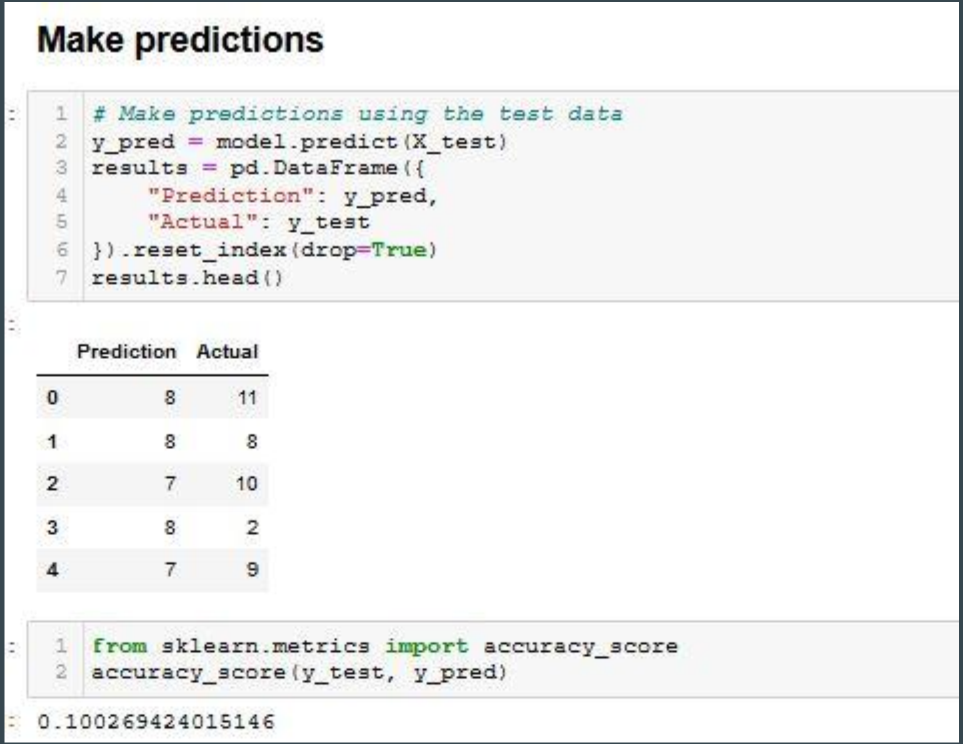
Create a SVM Model

```python
# Instantiate a linear SVM model
from sklearn.svm import SVC
model = SVC(kernel='linear')
```

Fit (train) or model using the training data

```python
# Fit the data
model.fit(X_train, y_train)
```

```
SVC(kernel='linear')
```



Make predictions

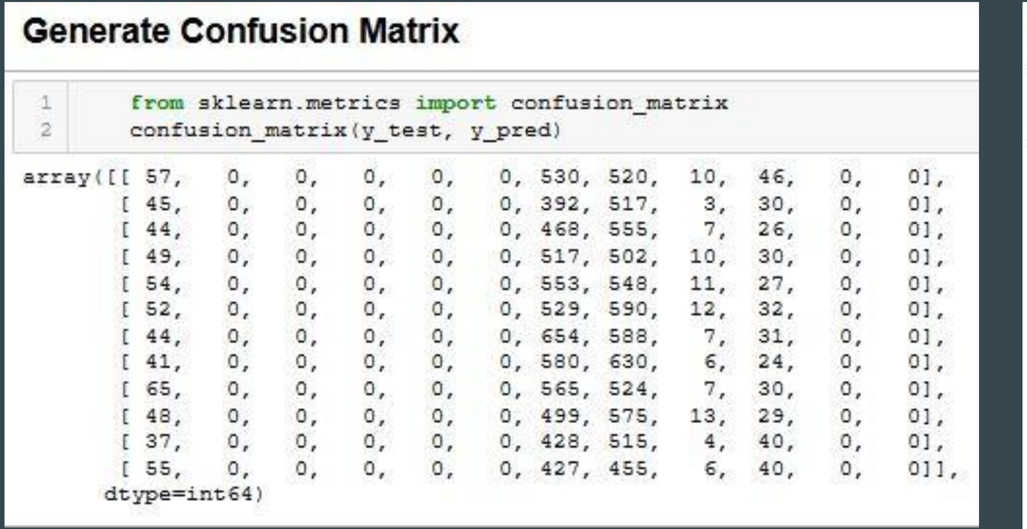
```python
# Make predictions using the test data
y_pred = model.predict(X_test)
results = pd.DataFrame({
    "Prediction": y_pred,
    "Actual": y_test
}).reset_index(drop=True)
results.head()
```

	Prediction	Actual
0	8	11
1	8	8
2	7	10
3	8	2
4	7	9

```python
from sklearn.metrics import accuracy_score
accuracy_score(y_test, y_pred)
```

```
0.10026942401514б
```

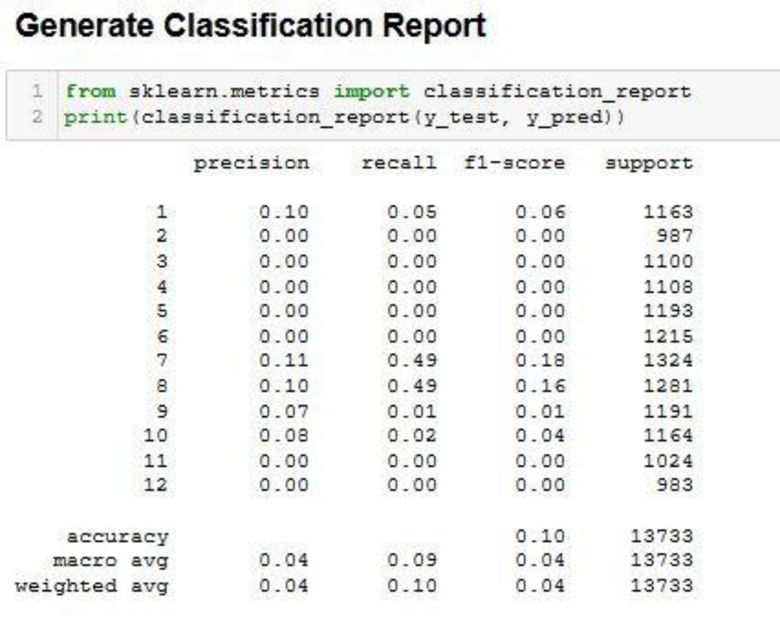
Predicting Crimes with SVM cont...



Generate Confusion Matrix

```
from sklearn.metrics import confusion_matrix
confusion_matrix(y_test, y_pred)
```

```
array([[ 57,   0,   0,   0,   0,   0, 530, 520,  10,  46,   0,   0],
       [ 45,   0,   0,   0,   0,   0, 392, 517,   3,  30,   0,   0],
       [ 44,   0,   0,   0,   0,   0, 468, 555,   7,  26,   0,   0],
       [ 49,   0,   0,   0,   0,   0, 517, 502,  10,  30,   0,   0],
       [ 54,   0,   0,   0,   0,   0, 553, 548,  11,  27,   0,   0],
       [ 52,   0,   0,   0,   0,   0, 529, 590,  12,  32,   0,   0],
       [ 44,   0,   0,   0,   0,   0, 654, 588,   7,  31,   0,   0],
       [ 41,   0,   0,   0,   0,   0, 580, 630,   6,  24,   0,   0],
       [ 65,   0,   0,   0,   0,   0, 565, 524,   7,  30,   0,   0],
       [ 48,   0,   0,   0,   0,   0, 499, 575,  13,  29,   0,   0],
       [ 37,   0,   0,   0,   0,   0, 428, 515,   4,  40,   0,   0],
       [ 55,   0,   0,   0,   0,   0, 427, 455,   6,  40,   0,   0]],
      dtype=int64)
```



Generate Classification Report

```
from sklearn.metrics import classification_report
print(classification_report(y_test, y_pred))
```

```
              precision    recall  f1-score   support

           1       0.10      0.05      0.06      1163
           2       0.00      0.00      0.00       987
           3       0.00      0.00      0.00      1100
           4       0.00      0.00      0.00      1108
           5       0.00      0.00      0.00      1193
           6       0.00      0.00      0.00      1215
           7       0.11      0.49      0.18      1324
           8       0.10      0.49      0.16      1281
           9       0.07      0.01      0.01      1191
          10       0.08      0.02      0.04      1164
          11       0.00      0.00      0.00      1024
          12       0.00      0.00      0.00       983

    accuracy                           0.10     13733
   macro avg       0.04      0.09      0.04     13733
weighted avg       0.04      0.10      0.04     13733
```

Outcome

Results

- Both Logistic Regression and SVM accuracy results were close to each other and very low.
- Both models were chosen for the types of data and both were resulted in similar accuracy, the dataset had issues or limited in rows or features.
- They did not predict when the crime will take place.

Recommendation

- Include dataset from other states
- Include more years of dataset
- Include other types of crimes
- Expand the project by adding weather and economic indicators

Anything the team would have done differently:

Longer time exploring the dataset, prototyping machine learning models and working with a team.