1- Introduction

There is lot of crimes in America the number crimes is increasing by the time so here , we want to move our store to place the have less crimes and more safe so this will be our main task focus on using the data that are provided through police department we will divide the areas and cluster them to determine the best option for moving our business .

Problem

If someone is looking to open Restaurant, where would he open it?

So, there is a recommendation needed for the owners of the restaurants so we will try to analyze and understand the data to extract the best area.

2- Methodology

My methodology will consist of the following

1- collect the data

2- Understand the Data

3- Preprocess the data

4- Modeling

5- Evaluation

3- collect the Data

Data is downloaded from

(website using the link (https://data.sfgov.org/api/views/wg3w-h783/rows.csv)

4- Understand the Data

In this section I will describe the data that will be used to analyses and predict the best places and neighborhood the following Data have 27 attributes

Attribute	Description
Incident Datetime	**The date and time of the incident**
Incident Date	**The date of the incident**
Incident Time	**The Time of the incident**
Incident Year	The year the incident occurred
Incident Day of Week	**The day the incident occurred**
Report Datetime	**when the report was filed**

Row ID	**Identifier for the data set**
Incident ID	**identifier for incident reports**
Incident Number	**number issued on the report**
CAD Number	**Computer Aided Dispatch Number**
Report Type Code	**This code is used for the report types**
Report Type Description	**The description of the report type**
Filed Online	**Reports that are filed online**

Incident Code	**Code is used to describe the type of the incidents**
Incident Category	**Category mapped with incident code**
Incident Subcategory	**Subcategory mapped with incident code**
Incident Description	**description of the incident**
resolution	**resolution of the incident at the time of the report**
Intersection	**street names that intersect closes to the incident**
CNN	**identifier of the intersection**

Police District	
Analysis Neighborhood	
Supervision District	**The districts are numbered**

Attribute	Description
Incident Datetime	**The date and time of the incident**
Incident Date	**The date of the incident**
Incident Time	**The Time of the incident**
Incident Year	The year the incident occurred
Incident Day of Week	**The day the incident occurred**
Report Datetime	**when the report was filed**

Row ID	**Identifier for the data set**
Incident ID	**identifier for incident reports**
Incident Number	**number issued on the report**
CAD Number	**Computer Aided Dispatch Number**
Report Type Code	**This code is used for the report types**
Report Type Description	**The description of the report type**
Filed Online	**Reports that are filed online**

Incident Code	**Code is used to describe the type of the incidents**
Incident Category	**Category mapped with incident code**
Incident Subcategory	**Subcategory mapped with incident code**
Incident Description	**description of the incident**
resolution	**resolution of the incident at the time of the report**
Intersection	**street names that intersect closes to the incident**
CNN	**identifier of the intersection**

Police District	
Analysis Neighborhood	
Supervision District	**The districts are numbered**

3- Data Preprocessing

First I will start with calculating the correlation for the attributes

```
[99]: df2 = df.corr()

[100]: df2

[100]:
```

Then after that I will drop the unnecessary attribute for the analysis

```
df.drop(['Row ID' , 'Incident ID' , 'Incident Number' , 'CAD Number' , 'Report Type Code' , 'Filed Online' ,'Incident Code' , 'Incident Subcategory' ,'CNN','Supervisor District' , 'point' ,'
       'Current Police Districts' , 'Current Supervisor Districts' , 'Analysis Neighborhoods' , 'HSOC Zones as of 2018-06-05' , 'OWED Public Spaces' , 'Central Market/Tenderloin Boundary Po
       'Parks Alliance CPSI (27+TL sites)' , 'ESNCAG - Boundary File' , 'Areas of Vulnerability, 2016'] , axis = 1 , inplace = True)
```

```
df
```

After that I will use that data that are remaining in this data frame

Attribute	Description
Incident Date	**The date of the incident**
Incident Time	**The Time of the incident**
Incident Year	The year the incident occurred
Incident Day of Week	**The day the incident occurred**
Report Datetime	**when the report was filed**

Report Type Description	**The description of the report type**
Incident Category	**Category mapped with incident code**
Incident Description	**description of the incident**
resolution	**resolution of the incident at the time of the report**
Intersection	**street names that intersect closes to the incident**
Police District	**Name of the police district location**
Analysis Neighborhood	

Latitude	**Numeric value for location**
Longitude	**Numeric value for location**

Drop the missing values

The second preprocessing step that I did for the data is to remove the rows that contains Nan missing values

```
[131]: df = df.dropna(axis =0 )
```

Then after that I reduced some of the rows for the prediction and modeling but before starting the modeling I will visualize the highest 5 types of crimes

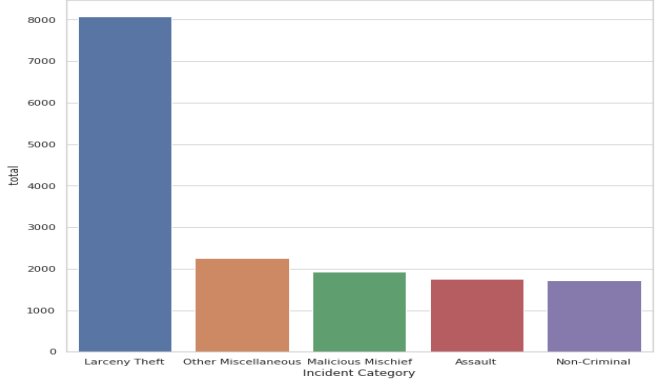
Visualize the top 5 crimes

```
[33]: DF_Category_top5 =DF_Category.head(5)



DF_Category_top5

ax = plt.subplots(figsize=(10, 9))
sns.set(style="whitegrid")
ax = sns.barplot(x="Incident Category", y=" total", data=DF_Category_top5)
```



Modeling

The Data can considered as classification problem I will use the classification models

1- Decision Tree

Decision Tree modelling is done with criterion=entropy and depth value of 10 to achieve the proper results

```
[144]: X_trainset, X_testset, y_trainset, y_testset = train_test_split(X, y, test_size=0.5, random_state=3)
```

```
[145]: drugTree = DecisionTreeClassifier(criterion="entropy", max_depth = 10)
       drugTree # it shows the default parameters
```

```
[145]: DecisionTreeClassifier(class_weight=None, criterion='entropy', max_depth=10,
                   max_features=None, max_leaf_nodes=None,
                   min_impurity_decrease=0.0, min_impurity_split=None,
                   min_samples_leaf=1, min_samples_split=2,
                   min_weight_fraction_leaf=0.0, presort=False, random_state=None,
                   splitter='best')
```

```
[146]: drugTree.fit(X_trainset,y_trainset)
```

```
[146]: DecisionTreeClassifier(class_weight=None, criterion='entropy', max_depth=10,
                   max_features=None, max_leaf_nodes=None,
                   min_impurity_decrease=0.0, min_impurity_split=None,
                   min_samples_leaf=1, min_samples_split=2,
                   min_weight_fraction_leaf=0.0, presort=False, random_state=None,
                   splitter='best')
```

```
[147]: predTree = drugTree.predict(X_testset)
```

```
[148]: print (predTree [0:5])
       print (y_testset [0:5])
```

```
['Mission' 'Tenderloin' 'Mission' 'Mission' 'Mission']
                Analysis Neighborhood
2280              Castro/Upper Market
15244                      Nob Hill
18705                    Mission Bay
6116   Financial District/South Beach
17406                 Haight Ashbury
```

```
[150]: from sklearn import metrics
       import matplotlib.pyplot as plt
       print("DecisionTrees's Accuracy: ", metrics.accuracy_score(y_testset, predTree))
```

```
DecisionTrees's Accuracy:  0.15014522976092948
```

The accuracy of the model = 0.150145

KNN

Preprocessing steps is that I did standers scaler transformation for the numeric value and then make the value of k is 9

And test the model

The accuracy of the model = 0.14463271576203332