**San Francisco Crime Classification**

- Nandan Nayak

**1. Description of Problem and its Data Set:**

San Francisco was infamous for housing some of the world's most notorious criminals during 1934 to 1963. From Sunset to SOMA, and Marina to Excelsior, this dataset provides nearly 12 years of crime reports from across all of San Francisco's neighborhoods. Using the location attribute that is the X and Y co-ordinates of the location, the model tries to predict the category of crime that occurred. The problem and the dataset have been taken from Kaggle[1].

This project tries to implement two machine learning algorithms namely Decision tree and Clustering to train the model and predict the 39 categories of crime that are mentioned in the project. The categories of crime are:

1.Warrants; 2.Other-Offenses; 3.Larceny/Theft; 4.Vehicle-Theft; 5.Vandalism; 6.Non-Criminal; 7.Robbery; 8.Assault; 9.Weapon-Laws; 10.Burglary; 11.Suspicious-Occ; 12.Drunkenness; 13.Forgery/Counterfeiting; 14.Drug/Narcotic; 15.Stolen-Property; 16.Secondary-Codes; 17.Trespass; 18.Missing-Person; 19.Fraud; 20.Kidnapping; 21.Runaway; 22.Driving-Under-The-Influence; 23.Sex-Offenses-Forcible; 24.Prostitution; 25.Disorderly-Conduct; 26.Arson; 27.Family-Offenses; 28.Liquor-Laws; 29.Bribery; 30.Embezzlement; 31.Suicide; 32.Loitering; 33.Sex-Offenses-Non-Forcible; 34.Extortion; 35.Gambling; 36.Bad-Checks; 37.Trea; 38.Recovered-Vehicle; 39.Pornography/Obscene;

**2. Background**

Machine learning is the study, design and implementations of the algorithms that help computers to optimize the performance criterion using example data or past experience. Some examples of such algorithms are Clustering, decision trees, Naïve-Bayes classifier and so on. One of the widely used aspect of machine learning algorithms is to predict the values of some category by establishing a relationship(called the model) between the output values and input values taken from training data set and then implementing the model on the test data set(to test the accuracy) and real time data set. The concept of prediction is widely used in online shopping websites, medical field, determining the crimes and so on which helps in better decision making and preparation for the future.

**3. Dataset [1]**

The problem and the dataset that has been taken for this project is taken from Kaggle. The training dataset has 878049 observations and 9 attributes.

Below are the attributes that describe the dataset:

Dates - Timestamp of the crime incident

Category - Category of the crime incident (only in train.csv). This is the target variable you are going to predict.

Descript - Detailed description of the crime incident (only in train.csv).

DayOfWeek - The day of the week.

PdDistrict - Name of the Police Department District.

Resolution - How the crime incident was resolved (only in train.csv).

Address - The approximate street address of the crime incident.

X – Longitude of the location in degrees

Y – Latitude of the location in degrees

Of the above attributes, the location attributes such as X and Y co-ordinates are the ones which give the relevant information required to predict the category of crime.

**4. Method**

**a. Tools:**

**Pandas [2]:**

Pandas is a Python package providing fast, flexible, and expressive data structures designed to make working with “relational” or “labeled” data both easy and intuitive. It aims to be the fundamental high-level building block for doing practical, real world data analysis in Python. Additionally, it has the broader goal of becoming the most powerful and flexible open source data analysis / manipulation tool available in any language. In this project we use Pandas to read from .CSV file and obtain the independent variable and the response variables to use in the decision tree algorithm.

**StringIO [3]:**

This module implements a file-like class StringIO that reads and writes a string buffer (also known as memory files). In this project we have made use of this module to store the pictorial decision tree generated from the Pydot.

**Pydot [4]:**

Pydot is an open source program that allows you to easily create both directed and non directed graphs from Python. The program supports all attributes implemented in the Dot language (up to Graphviz 2.26.3). You can also inline the output into interactive scientific environments.

In this project we have made use of Pydot to create a pictorial representation of the decision tree.

**Scikit-learn [5]:**

Scikit-learn (formerly scikits.learn) are an open source machine learning library for the Python programming language. It features various classification, regression and clustering algorithms including support vector machines, random forests, gradient boosting, k-means and DBSCAN, and is designed to interoperate with the Python numerical and scientific libraries NumPy and SciPy.

In this project we have made use of Scikit-learn to use the DecisionTreeClassifier which is required to create the decision tree and also Train\_test\_split function to split the training dataset into training and test data set.

**Numpy [6]:**

NumPy is an extension to the Python programming language, adding support for large, multi-dimensional arrays and matrices, along with a large library of high-level mathematical functions to operate on these arrays. NumPy is open source and has many contributors.

Numpy library of python has been implemented in our code in order to convert the dataframes generated by Pandas library to arrays. These arrays are easier to train the algorithm and are useful means to plot the graph using Matplotlib.

**Matplotlib [7]:**

Matplotlib is a plotting library for the Python programming language and its numerical mathematics extension NumPy. It provides an object-oriented API for embedding plots into applications using general-purpose GUI toolkits like wxPython, Qt, or GTK+. Matplotlib is designed to closely resemble the functionalities of MATLAB.

Matplotlib library of python has been implemented in our code in order to plot the clusters generated by k-means model on a graph.

**Homogenity Score [sklearn.metrics.homogeneity\_score][8]:**

Homogeneity metric of a cluster labeling given the ground truth of the model is known. A clustering result that satisfies homogeneity if all of its clusters contain only data points which are members of a single class. This metric is independent of the absolute values of the labels: a permutation of the class or cluster label values will not change the score value in any way. The score ranges between 0 and 1 where 1.0 stands for perfectly homogeneous labeling.

**Completeness Score [sklearn.metrics.completeness\_score][9]:**

Completeness metric of a cluster labeling given a ground truth. A clustering result satisfies completeness if all the data points that are members of a given class are elements of the same cluster. This metric is independent of the absolute values of the labels: a permutation of the class or cluster label values won’t change the score value in any way. The score ranges between 0 and 1 where 1.0 stands for perfectly complete labeling.

**V Measure Score [sklearn.metrics.v\_measure\_score][10]:**

V-measure cluster labeling given a ground truth. The V-measure is the harmonic mean between homogeneity and completeness.

v = 2 \* (homogeneity \* completeness) / (homogeneity + completeness)

This metric is independent of the absolute values of the labels: a permutation of the class or cluster label values won’t change the score value in any way. The score ranges between 0 and 1 where 1.0 stands for perfectly complete labeling.

**Adjusted Random Score [sklearn.metrics.adjusted\_rand\_score][11]:**

The Rand Index computes a similarity measure between two clusters by considering all pairs of samples and counting pairs that are assigned in the same or different clusters in the predicted and true clusters.

The raw RI score is then “adjusted for chance” into the ARI score using the following scheme:

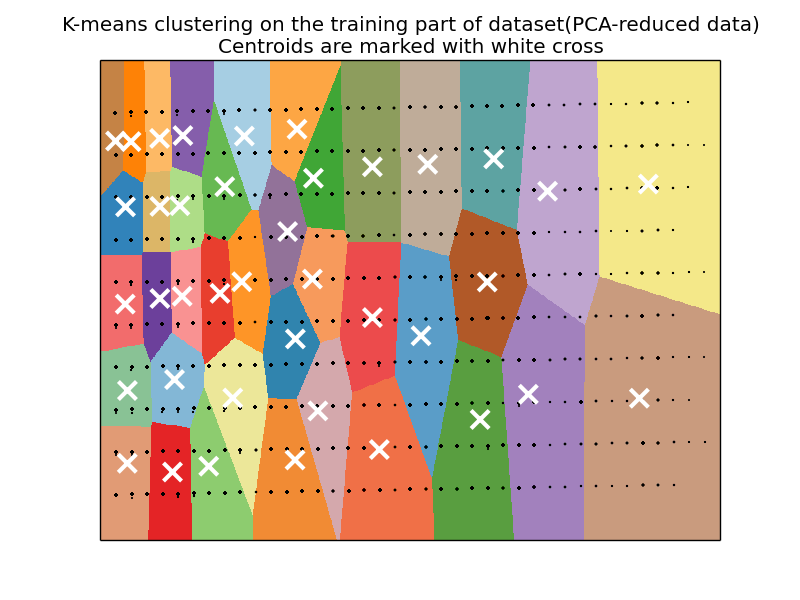
ARI = (RI - Expected\_RI) / (max(RI) - Expected\_RI)

The adjusted Rand index is thus ensured to have a value close to 0.0 for random labeling independently of the number of clusters and samples and exactly 1.0 when the clusters are identical (up to a permutation).

**b.Algorithms**

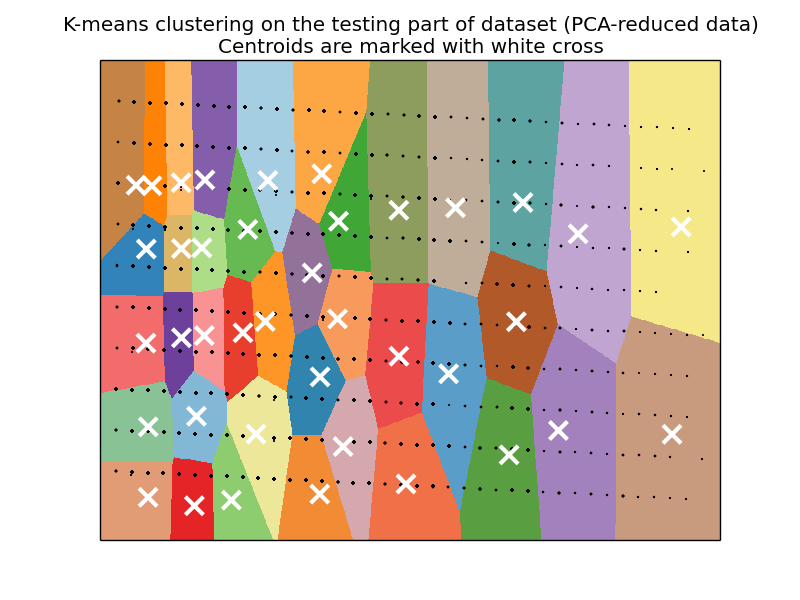
**K-Means Clustering [12]:**

K-means clustering is a method of vector quantization that is used to analyze clusters in data mining. The goal of k-means clustering is to partition n observations into k clusters in which each observation belongs to the cluster with the nearest mean, serving as a prototype of the cluster. This results in a partitioning of the data space into Voronoi cells.



**Fig1. K-Means cluster on training data set**

The above figure shows the clusters formed from the training data set in our project.



**Fig2. K-Means cluster on test data set**

The above figure shows the clusters formed from the test data set in our project. There are 39 categories and we have 39 clusters. The test data is projected onto the trained K-means cluster.

**Decision Tree [13]:**

A decision tree is a decision support tool that uses a tree-like graph or model of decisions and their possible consequences. It is a graphical representation of the abstract data into tree-like graph representing decisions and their consequences. The Decision tree starts with the root node and grows branches based on the complexity of the dataset. The decisions are represented by the nodes and the consequences are represented by branches. The algorithm is simple to implement and is very intuitive. Decision trees are commonly used in operations research, specifically in decision analysis, to help identify a strategy most likely to reach a goal.

The results of this algorithm are available in pictures DeciTestDataTree.pdf (Test set results) and DeciTrainDataTree.pdf (Training set results).

**PCA [Principle Component Analysis][14]:**

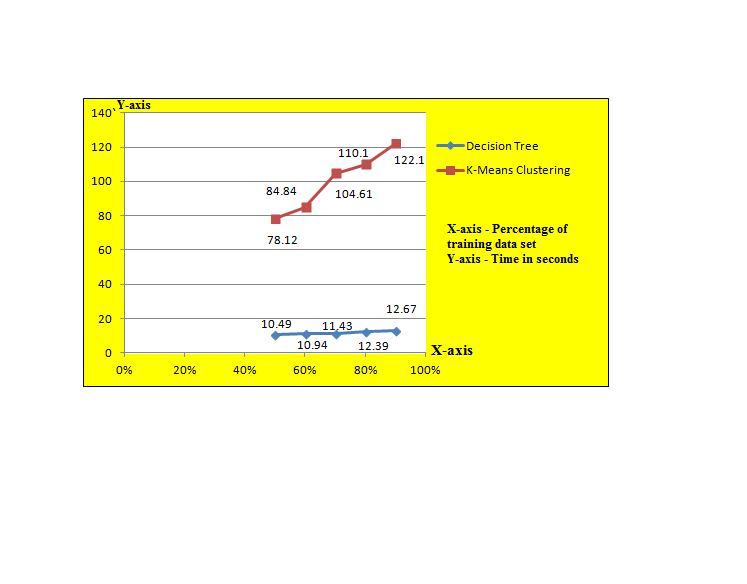
We have implemented PCA decomposition algorithm on our data before we proceeded to K-Means clustering. Principal component analysis (PCA) is a statistical procedure that uses an orthogonal transformation to convert a set of observations of possibly correlated variables into a set of values of linearly uncorrelated variables called principal components. PCA helps in reduction of computations which in turn enhances the performance of algorithm depending on the distribution and correlation of data. In our algorithm we have used PCA mainly reduce the number of dimensions in our data set.

**5. Experiment: performance analysis and results**

**Time Performance Analysis:**

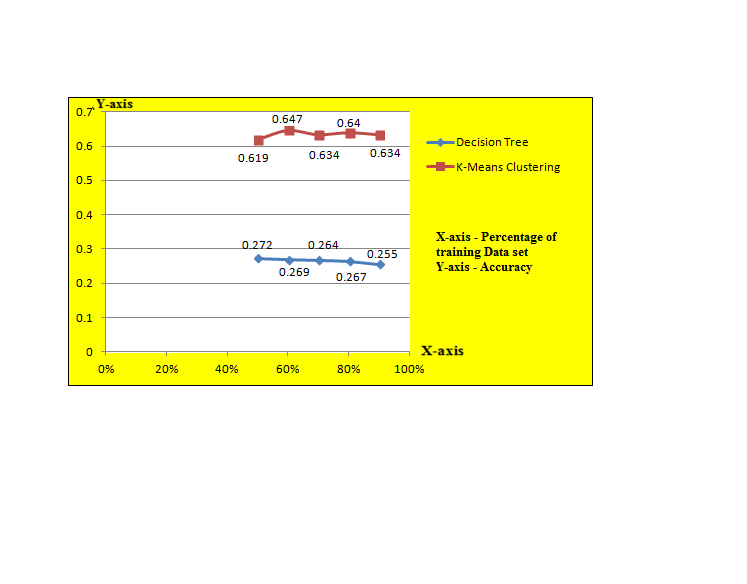
In the below figure, the X-axis refers the fraction of the dataset (80% = 80% of data set) and the Y axis represents the time taken in seconds.

We see increased the size of the data set keeping the test data set to 10% of the total observations in all the cases in order for better comprehension of the performance of both the algorithms. We see that the Decision tree algorithm takes pretty much the same time to learn whereas the K-Means algorithm's learning time increases with the increase in the size of the data set.

****

**Fig3. Time Performance Analysis of Decision Tree with K-Means Clustering**

**Accuracy Analysis:**

****

**Fig4. Accuracy Analysis of Decision Tree with K-Means Clustering**

In the above figure the X-axis refers the fraction of the dataset (0.8 = 80% of data set) and the Y axis represents the accuracy of the model (0.7 represents that the model is predicts 70% of correct values).

We see that the accuracy of the Decision tree remains almost constant with increase in the data set and very much stable. The accuracy of the Clustering model varies with the increase in the size of the dataset but produces high accuracy in comparison to the decision tree.

**6. Conclusion**

Decision Trees are a non-parametric supervised learning method used for classification and regression. Decision trees are more useful when the dataset follows simple rules of prediction. They are easy to implement and the time for learning is almost the same for the size of the input data. In our project the categories follow complex rules as many outcomes are linked to similar set of input data resulting in deeper tree and poor accuracy.

K-Means Clustering algorithm is a simple unsupervised learning algorithm that solves many clustering problems. The algorithm is more useful when we are certain or have a clear idea on the value of k that is to be defined. They are easy to implement when compared to other clustering algorithms. In our project, we already know the best value of k and hence we obtained more accurate cluster results [63%] as depicted in the graph. One major disadvantage that was observed during the course of project was that different initial partitions resulted in different final clusters. Therefore rerunning the same program provided more optimal clusters.

**7. References:**

[1]. Kaggle Problem and the Dataset. <https://www.kaggle.com/c/sf-crime>

[2]. Pandas. <http://pandas.pydata.org/pandas-docs/version/0.17.1/>

[3]. StringIO. <https://docs.python.org/2/library/stringio.html>

[4]. Pydot. <http://python-pydot.software.informer.com/1.0/>

[5]. Scikit-Learn. <https://en.wikipedia.org/wiki/Scikit-learn>

[6]. Numpy. <https://en.wikipedia.org/wiki/NumPy>

[7]. Matplotlib. <https://en.wikipedia.org/wiki/Matplotlib>

[8]. Homogenity Score.

<http://scikit-learn.org/stable/modules/generated/sklearn.metrics.homogeneity_score.html>

[9]. Completeness Score.

<http://scikit-learn.org/stable/modules/generated/sklearn.metrics.completeness_score.html#sklearn.metrics.completeness_score>

[10]. V Measure Score.

<http://scikit-learn.org/stable/modules/generated/sklearn.metrics.v_measure_score.html#sklearn.metrics.v_measure_score>

[11]. Adjusted Random Score.

<http://scikit-learn.org/stable/modules/generated/sklearn.metrics.adjusted_rand_score.html#sklearn.metrics.adjusted_rand_score>

[12]. K-Means Clustering. <https://en.wikipedia.org/wiki/K-means_clustering>

[13]. Decision Tree. <https://en.wikipedia.org/wiki/Decision_tree>

[14]. PCA. <https://en.wikipedia.org/wiki/Principal_component_analysis>