San Francisco Crime Classification

Rahul Aravind Mehalingam rxm151730 | Kunal Krishna kxk155230 | Radhika Simant rrs150130 | Adithya Adithya axa155930 | Sarat Chandra Varanasi sxv153030

***Abstract*—Classification is a common and a useful analysis to perform on real-world data. The domains on which classification can be performed are widely diversified. With the ever increasing capability to harness large volumes of data it is only logical to tackle classification as a big data problem. In this project, we have used the San Francisco Crime Data (available on Kaggle) to perform classification. The classification that we have done is on the category of crime ranging from larceny to burglary in the San Francisco area. Because of the size of the data (containing atleast 300,000 records) and the various ad-hoc classification queries that can be posed on the crime data, we have ued Apache Spark to effectively run the classification algorithms on the given crime data.**

***Keywords*—Big Data project, Classification, Crime Classification, San Francisco Crime Data, Apache Spark, Tableau**

I. Introduction

Crime classification is a really important task when it comes to handling criminal cases. The type of a crime sets its appropriate judicial course of action. This work is usually done manually. The SF data published on Kaggle is an example of this. The data contains crimes of past 12 years. Developing a machine learning classifier is the obvious next step to automate this process. Given various features of a new incident, one should with reasonable accuracy be able to predict the kind of crime attached to the incident. Once we have a prediction of crimes, we can analyze the crime rate and the type of crimes across various neighborhoods. Therefore a layman can have a notion of safe and unsafe areas with better accuracy.

The rest of the report is organized as follows - We briefly describe the data format of the SFO crime data. Then we describe how we leveraged the attributes' information to generate additional features. We then go on to explain how we used the feature engineering to build machine learning classifier pipelines done in Apache Spark. Finally we give an analysis of the various metrics and a comparison of various classifiers we implemented

II. The SFO Crime Dataset

The SFO crime dataset contains attributes such as the date and time of occurance of the crime incident, the district of the police department which handles the incident, day (day of week) of the incident, the street address and the type of crime. All these attributes constitute the training dataset. The test data set for predicting the class of crime does not include the attributes such as the day of week.

III. Feature Engineering

We have six feature Dates,DayOfWeek,Address which is unusable as is (23 228 distinct addresses),PdDistrict,X and Y which is basically one single feature: the coordinates of the incident

we have done various Feature Engineering based on above feature like Weekend feature,Time-related features,Neighborhood feature,Address features,Day or night feature,Weather features. Weekend feature in which used, for this created a group of weekday and weekend(Friday,Saturday,Sunday) and count of different types of crimes. In order to feature engineering based on address we saw address are in two forms 1) street 1 / street 2 to denote an intersection 2) number Block of street. So with the help of this idea we introduced two types Address Type(which indicates whether the incident took place at an intersection or on a particular street) and street(where I attempted to parse the Address variable to a single street name).

Day or night feature, Along the same lines of the HourOf-Day feature, we reasoned that it would be interesting to see if an incident occurred during the day or the night. We used the sunrise-sunset.org API. It basically makes a request to the API(for a given latitude and longitude) for each day present in the dataset to retrieve the time of sunrise and sunset, parses the json-formate. Note:A request is made every five seconds in order not to overload the API.Time-related features we group date based on year and get the count of different crimes.

Neighborhood feature, we came up with the idea of trying to find the neighborhood where the incident occurred thinking that particular types of crimes are more inclined to happen in particular neighborhoods.To find San Francisco neighborhoods as latitude/longitude polygons we used the Zillow API which fitted the bill perfectly providing neighborhoods for California as a shapefile. Also used the ESRI geometry api which lets you do all kinds of spatial data processing and, for us, check if a point (corresponding to an incident) is inside a polygon (corresponding to a neighborhood).

Finally ,Weather features, weather could have an influence as opposed to other types of incidents which would occur indoor and for which the weather wouldnt have any impact. We assembled a dataset containing the most occurring weather condition and average temperature of every day in the dataset using wunderground.com(gives data in csv).

IV. Machine learning pipeline

We employed following models for classification and compared the performance of them.

1 . Random Forest

The random forest is an ensemble approach that can also be thought of as a form of nearest neighbor predictor. It is considered to be one of the most effective and versatile in solving

almost any prediction task. Random Forests are a collection of decision trees (AKA Classification And Regression Trees). Through random sampling, Random Forests gains predictive power. Random Forests can be used for classification tasks, e.g. credit risk, patient disease risk, mechanical failure. Here we have used random forest to classify the type of crime based on several features like weather, address, dayOfWeek, year etc. Some excellent features of random forest are : it is unexcelled in accuracy among current algorithms, it runs efficiently on large data sets, It can handle thousands of input variables without variable deletion.

2 . Decision Tree

Decision tree is one of the most widely used and practical methods for inductive inference. It is a method for approximating discrete valued target functions, in which the learned function is represented by a decision tree. Learned trees can also be represented as sets of if-then rules to improve human readability.

The central choice in the ID3 algorithm is selecting which attribute to test at each node in the tree. We would like to select attribute that is most useful for classifying. We will define a statistical property, called Information Gain, that measures how well a given attribute separates the training examples according their target classification. ID3 uses this information gain measure to select among the candidate attributes at each step while growing the tree.

3 . Naive Bayes

Naive Bayes assumes that all attributes are independent of each other given the context of the class. Because of the independence assumption, the parameters for each attribute can be learned separately, and this greatly simplifies learning, especially when the number of attributes is large. Naive Bayes Classification algorithm is commonly used in text classification tasks. The training and test features used for Naive Bayes are usually represented as counting vectors or TF-IDF vectors, in which numeric values are always non-negative. In order to apply Naive Bayes model into our prediction task, we transform the X and Y values to the corresponding absolute values. Naive Bayes assumption states that feature are conditionally independent, which may be false with our feature vector as the filed PdDistrict is related to coordinate (X,Y) to some extent.

4. Logistic Regression

V. Analysis

We have done analysis based on feature regarding importance of each feature, below is the result of our experiment:

Importance of Feature	
Feature Name	Importance
DayOfWeekIndexed	7.91653437668286E-4
PdDistrictIndexed	0.22761362959319115
DayOrNightIndexed	0.009991092069811355
WeekendIndexed	2.246177273116106E-4
HourOfDayIndexed	0.04103688343200673
MonthIndexed	0.0
YearIndexed	0.011272557834208486
AddressTypeIndexed	0.22968325796525954
StreetIndexed	0.13919189341735466
weatherIndexed	0.0
NeighborhoodIndexed	0.07545966678900771
X	0.08989258961685669
Y	0.17484215811732376

we splitted our traning dataset to 7:3 ratio for traning and testing to get the prediction accuracy of our model. Below are accuracy which we got in different classifier:

Accuracy of Model	
Model Name	Accuracy in percentage
Random Forest	87.43
Decision Tree	81.65
Logistic Regression	47.39
Naive Bayes	78.49

Best accuracy we got in Random forest and least accuracy in logistic regression

VI. Visualizations

A. *Naive Bayes*

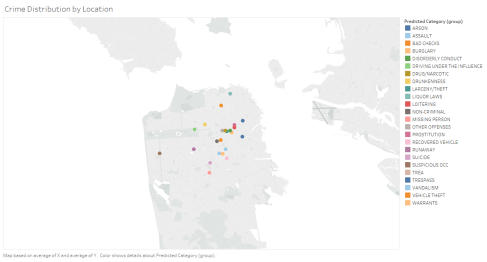

Fig. 1. Crime Distribution by Locations

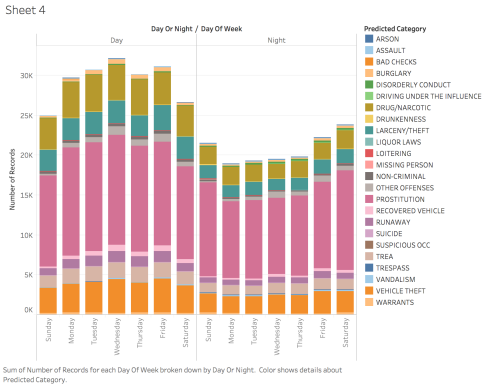

Fig. 2. Crime Distribution by Day/Night and Day of Week

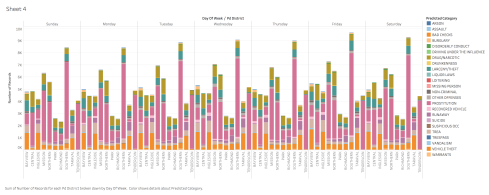

Fig. 3. Crime Distribution by day of week and District

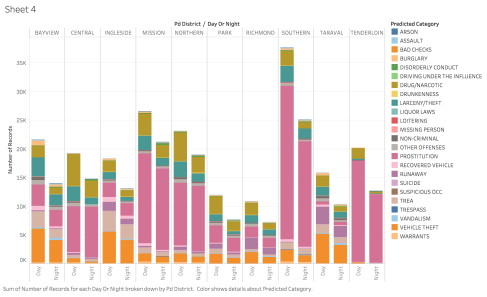

Fig. 4. Crime Distribution by Day/Night and District

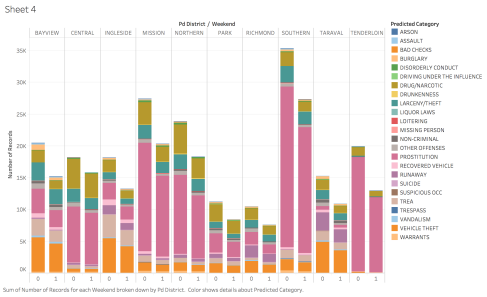

Fig. 5. Crime Distribution by Weekend and District

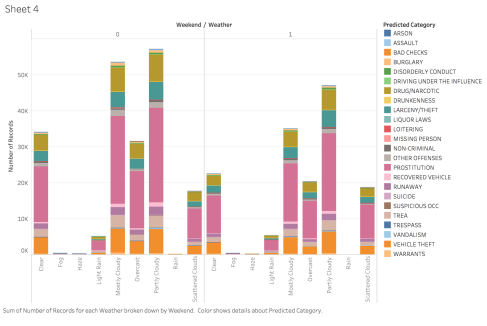

Fig. 6. Crime Distribution by Weather and Weekend

B. Random Forest

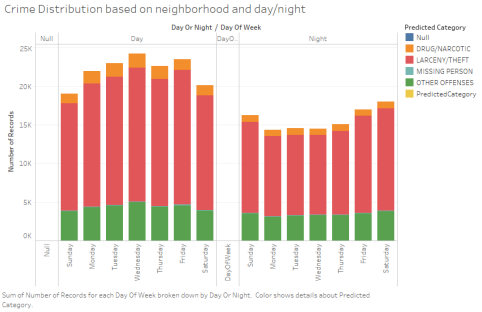

Fig. 7. Crime Distribution by Day/Night and Day of Week

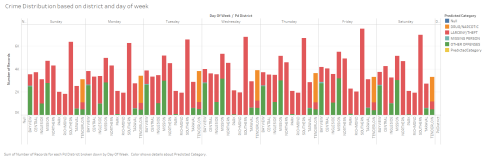

Fig. 8. Crime Distribution by day of week and District

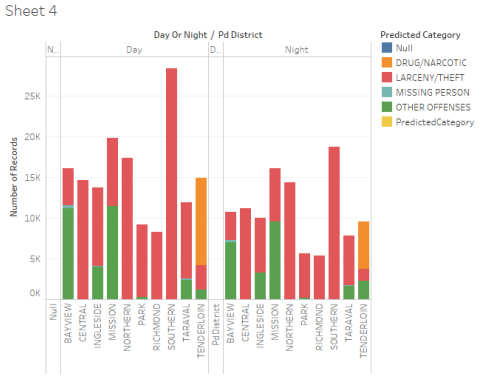

Fig. 9. Crime Distribution by Day/Night and District

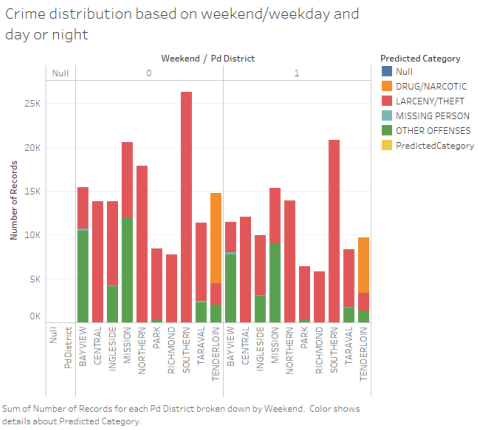

Fig. 10. Crime Distribution by Weekend and District

C. Decision Tree

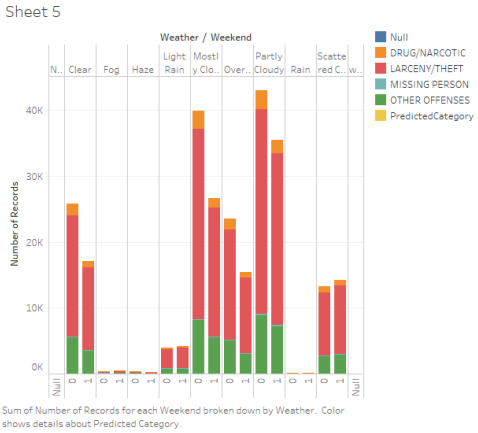

Fig. 11. Crime Distribution by Weather and Weekend

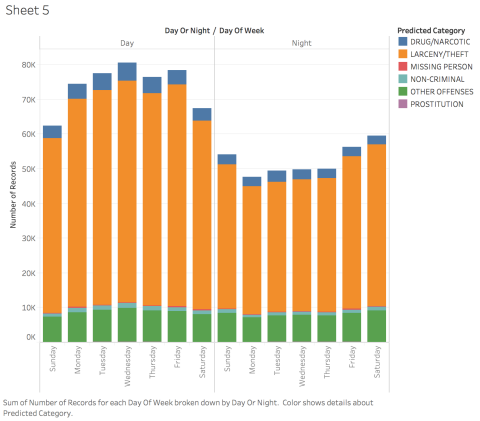

Fig. 12. Crime Distribution by Day/Night and Day of Week

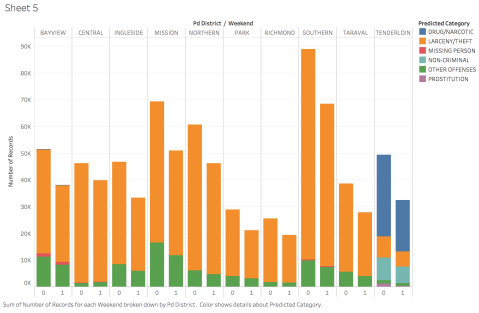

Fig. 13. Crime Distribution by day of week and District

VII. Conclusion

For this prediction task, we started from pre-processing the data set from SFPD Crime Incident Reporting system. Then, we attempted to select some helpful features to represent the attributes of the samples in a proper manner. Adopting feature engineering process based on location, time, weather information turned out to improve the performance of our models by a lot. Finally, by training some models with the features we employed and calculating the probabilities of different categories of crimes, we completed the whole task. As presented in the results,Decision tree performance was also good it gave good accuracy, we tried logistic regression but

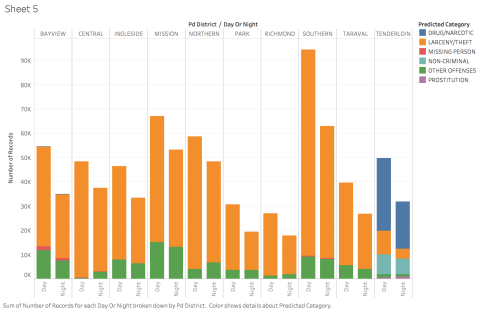

Fig. 14. Crime Distribution by Day/Night and District

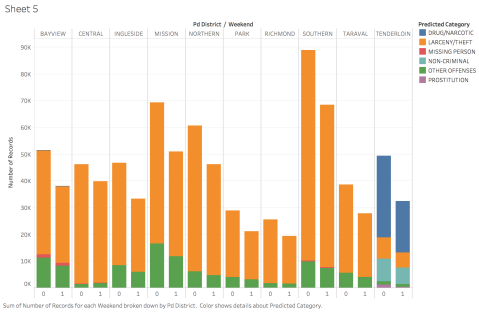

Fig. 15. Crime Distribution by Weekend and District

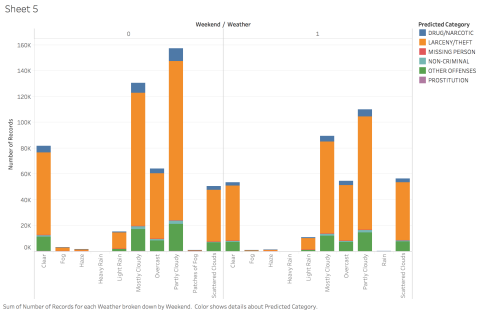

Fig. 16. Crime Distribution by Weather and Weekend

we were not able to get good results, Nave Bayes is not a perfect model for this task because some of the features do not represent the count or frequency. Random forest turned out to be the best model in our experiment, but it is relatively time consuming. We used tableau to visualize result based on certain criteria like Crime distribution based on geographical location, Crime Distribution by Day/Night and Day of Week ,Crime Distribution by day of week and District,Crime Distribution by Day/Night and District,Crime Distribution by Weekend and District,Crime Distribution by Weather and Weekend.

Acknowledgment

The authors would like to thank Dr. Latifur Khan and M. Solaimani for guidance.

References

[1] H. Kopka and P. W. Daly, *A Guide to LaTeX*, 3rd ed. Harlow, England: Addison-Wesley, 1999.

[2] San Francisco Crime Dataset(2015). Available from: https://www.kaggle.com/c/sf-crime/data [3] San Francisco Crime Classification Evaluation(2015). Available from: https://www.kaggle.com/c/sfcrime/details/evaluation [4] Data Transformation / Learning with Counts(2015). Available from: [5] UCI Machine Learning Repository (2012). Available from: http://archive.ics.uci.edu/ml/datasets.html https://msdn.microsoft.com/enus/library/azure/dn913056.aspx [6] Breiman, Leo, et al. Classification and regression trees. CRC press, 1984.