2021

Crime Analysis and Predictive Policing Report

DATA ANALYTICS AND TECHONOLOGICAL TOOLS

S5125260

Table of contents

Table of Tables

Table of Figures

1. Introduction

The application of data science tools and technologies has become very broad. All these applications aim towards the solution of people's problems or needs (Costa 2020). One problem to be solved is to predict possible future crimes and prevent them when possible by law enforcement agencies, 'to identify likely targets for police intervention and prevent crime or solve past crimes by making statistical predictions' (Perry el al. 2013)

Los Angeles Police Department uses spatial prediction to allocate patrols and has decreased crimes by 5.4% and homicides by 22.6%(Uchida et al., 2012). The systems involve using algorithms to analyze massive amounts of historical and real-time data to predict and help prevent future events. Generally, the systems fall within two categories: place-based and person-based predictive policing.

In this report, an experiment on crime data under the category of place-based predictive policing will be carried out on the San Francisco P.D Dataset with the following objectives:

- To perform an exploratory data analysis to gain insights about crimes and to map crime hotspots in San Francisco
- To develop predictive models to determine in which police district a crime may fall within
- To critically analyze the performance of the models and determine whether the system will provide insights for law enforcement officers or not

The report will address the objectives as follows:

1. The dataset is presented, described, analyzed and visualized
2. The approaches for preprocessing, the model selection, its implementation, the validation method chosen and evaluation metrics and the first experimental result will be provided
3. Based on the first experimental results, it will be determined whether the machine learning approach requires scaling up, and if so, the tools and techniques for scaling up will be discussed and implemented.
4. The results are presented and evaluated together with overall conclusions, lessons learnt, recommendations and future work.

Furthermore, all the tools required at all the stages throughout the project will be described.

2. The data

Due to the large number of features, the complexity and the volume of the original dataset, just a section of the San Francisco Police Department crime incident system dataset was selected and downloaded from Kaggle and read as a comma-separated value (.csv) file. The dataset consists of 150500 crime records and 13 features , most data types are object values, with texts on them. Finally, the class feature for the predictive model is represented by the "PdDistrict" feature with a total of 10 different classes. A broad overview of the dataset is provided in Table 1, and further details about the features are shown in Appendix 1.

Table 1. Overview of dataset

Dataset	Number of Features	Number of instances	Class feature	Total number of classes	Duplicates values	Null values	Years Covered
San Francisco Police Department Crime Incident Reporting System	150500	13	PdDistrict	10	None	1	2016

2.1 Data characteristics and Big data

As defining Big Data has been a challenging task, some data characteristics have been identified to determine whether a big data problem may be faced or not. Five of these characteristics as presented by Arbab-Zavar (2021), are described and evaluated in Table 2.

Table 2. Big data characteristics

Big Data Characteristic	Description	Evaluation
Volume	The volume refers to the size of the data; however, there is no exact measurement for such characteristic. Therefore, to understand whether this characteristic is present, it usually stands out when the following challenges are introduced: • It may not be able to maintain the integrity and security of large sets of data • The processing and access of data is not easy	As the dataset selected is just one small section of a larger dataset, these challenges may not affect the project deeply. In a real-world scenario, agencies do face volume challenges when dealing with these kinds of datasets as the data may be collected on a real-time basis (especially for person-based predictive policing), besides handling the historical data already collected.

Variety	Data can be structured, unstructured, semi-structured and may consist of a number, texts, images, audio files, among other types. This characteristic analyses the variety of data types encountered in a dataset. In Big Data problems, many kinds of data may be combined.	Most of the data types in the San Francisco dataset are objects and contain texts; nevertheless, float and integers are also present in the dataset. As most of the features are objects with text values on them, some preprocessing will be required for the data to be more useful.
Velocity	Addresses the speed at which data is generated—evaluated by the frequency of generation and the frequency of handling, recording and publishing data.	Crime data is generated every day, at any time. Place-based predictive policing collects data at a slower speed than person-based predictive policing, and the main reason behind this is because these types of systems work with real-time data gathered through various methods (e.g. IoT devices as CCTV cameras).
Veracity	Data veracity is concerned with the data provenance, ownership, quality, uncertainty, trust and trustworthiness. If the data achieves a high level of veracity, it could be spoken of it as data accuracy. Furthermore, data bring much bias; therefore, the main challenge here is to determine how much trust is given to the data	As crime data is usually provided by governments open data sources, and as law enforcement agencies use the data to develop systems to enhance their systems, it is concluded that the data is trustworthy.
Value	The value of the data is more of a goal. It is unclear to determine the value of big data; yet, it is possible to determine how valuable is or not the knowledge extracted from the data.	The main goal of this project is to be able to provide valuable information and insight for the police officers when critical decision making is required.

2.2. Data exploration via data visualization

2.2.1 Null values and duplicates

The dataset contains only one null value on the "PdDistrict" attribute, and no duplicates are encountered. For visualizations purposes, the instance with the null value was dropped.

2.2.2 Class Imbalance

Figure 1. shows the class distribution among the different districts in San Francisco. The distribution among classes is not balanced. The classes, their encoded labels, and the distribution among the dataset are shown in Appendix 1. A histogram using the Matplotlib library from Python was plotted for visualization purposes. The districts with the most concentration of crimes recorded are in the southern, northern, mission and central districts.

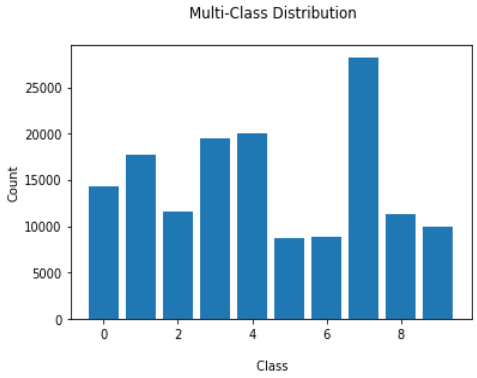

Figure 1. Class distribution

2.2.3 Outliers and Mapping Crime Data

The X and Y attributes have been selected to represent the crime's coordinates, allowing to map the crime hotspots in San Francisco. According to LatLong.net (2012), the latitude and longitude coordinates of San Francisco are 37.773972, -122.431297. The Matplotlib scatter plot class is used to map the distribution of the crimes recorded in San Francisco. As shown in Figure 2., some dots plotted seem to be outside the san Francisco city. As the project is only interested in mapping the crimes in San Francisco, the outliers have been dropped and plotted again in Figure 3.

Both shown a concentration of dots on the top right area of the diagrams. This area covers the northern and central districts of San Francisco and the concentration of crimes registered per coordinate. It is clear that these areas are the crimes hotspots in San Francisco.

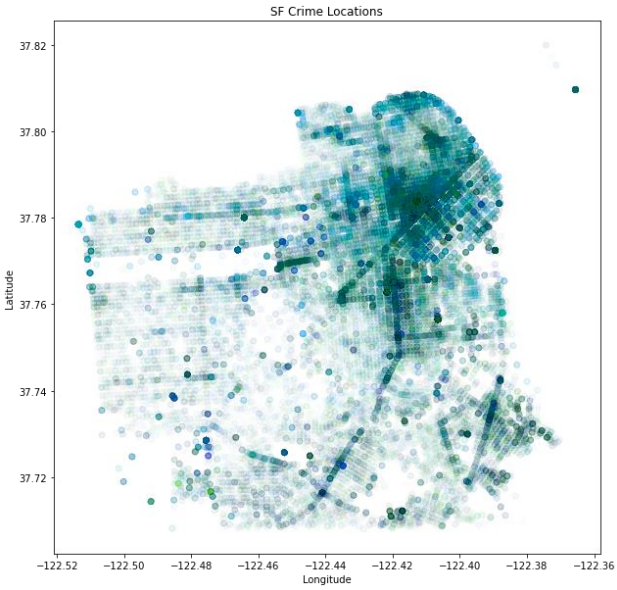

Figure 2. Crime map with outliers

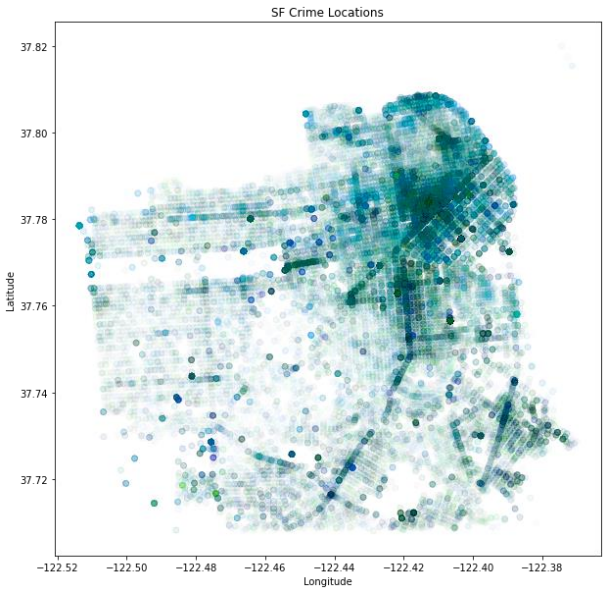

Figure 3. Crimes map without outliers

2.2.4 Insights Heatmap

To gain further insights into the crimes in San Francisco, heatmaps using the Seaborn library from Python for visualizations, provide very satisfying and informative statistical graph.

Two heatmaps have been plotted. First, Figure 4. was plotted to identify the rate of the crime category per district. It is observed that the LARCENY/THEFT crime category has the most significant number of records on the southern northern and central districts. It is then followed by other offences, assaults and non-criminal categories; however, they seem to be more usually recorded on the southern, mission and central districts. The second heatmap, shown in Figure 5. represents the rates of the crime category recorded per day of the week. The most recorded crimes are larceny/theft, assault, non-criminal and other offences throughout the whole week. Fridays and Saturdays tend to have a larger number of crimes recorded than the rest of the week for the categories mentioned. It is also observed that the number of recorded assault crimes is slightly higher during Wednesdays compared to the rest of the week. Vandalism, vehicle theft, warrant and suspicious occ crime categories are the most recorded crimes following the beforementioned ones.

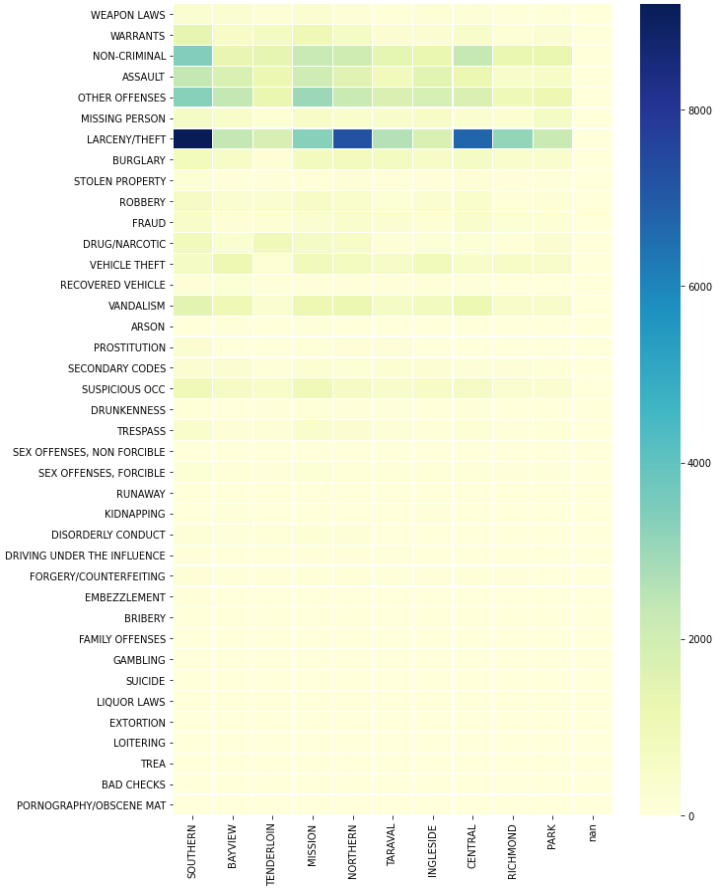

Figure 4. Type of crimes recorded per district

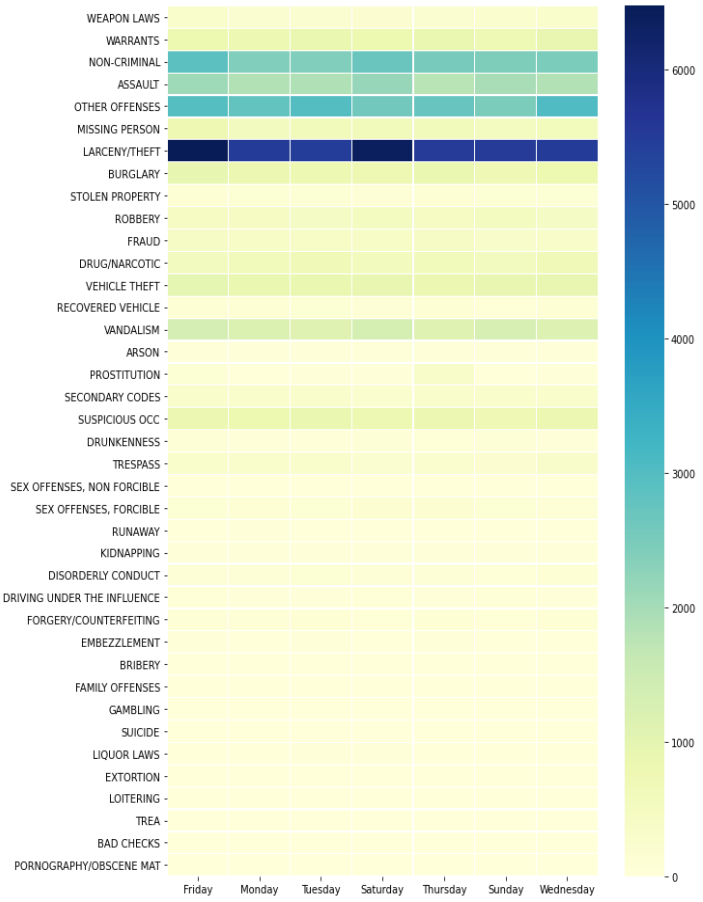

Figure 5. Type of crimes recorded per day of week

3. Knowledge extraction via machine learning

3.1 Methodology

3.1.1 Objectives

One of the project objectives is to develop a predictive model that determines the district a crime may fall within and compare its performance against other models. Furthermore, another aim is to determine whether the models will provide insights for the law enforcement officers. Knowledge extraction using machine learning, various Python libraries and data science tools will be used and discussed throughout the stages of the project to meet the objectives.

3.1.2 Data processing

Before deploying the model, data cleaning and processing is required. To achieve this, the Python library Scikit-Learn is used.

1. The feature "IncidentNumber" is dropped.
2. All features with data type value different to float64 are selected and encoded using the label encoder utility class from Scikit-Learn. It can transform non-numerical values into numerical labels (Pedregosa et al. 2011).
3. As mention in section 2.2.1, the features X and Y contain outliers, and the unique missing value is dropped.

3.1.3 Classification algorithms

Scikit-Learn is a simple and efficient tool for performing machine learning in Python. The models chosen for the experiment are supervised and instance-based algorithms; the Scikit-Learn library is highly suitable for the chosen classifiers. The classifiers that will be implemented are K-Nearest Neighbor(KNN), Decision Trees(DT) and Random Forests(RF). The main objective of the classifiers is to be able to identify to which set of categories a new observation belongs. The class feature is represented by the "PdDistrict" feature with ten different classes; therefore, the classifiers will be dealing with a multi-class classification problem. As the classifiers implement different methods themselves, the performance among them will be evaluated and compared.

3.1.4 Validation method

To test whether the classification models work correctly, they must undergo a validation process. The selected method is cross-validation. The Scikit-Learn library from Python also provides this method. This method will randomly distribute the data into train and test set that generalization is unlikely to happen. Due to the randomness in the train and tests sets, the outputs of the models gain more trusts. Furthermore, the method provides functions to calculate scores and evaluate the performance of the classifiers with different metrics (Pedregosa et al., 2011).

3.1.5 Evaluation metrics

The evaluation metrics is the method to evaluate the model's performance. Scikit-Learn provides a series of evaluation metrics and estimators, which are very useful for the project. The selected evaluation metrics for this project are described in Table 3.

Table 3. Evaluation metrics description

Evaluation Metric	Description
Accuracy	Measures the accuracy of all prediction. It divides the total number of correct classifications by the total number of instances
Precision	Calculates the accuracy of the true positive predictions
Recall	Is the ratio of true positive instances correctly classified by the models
F1-score	Could be considered as the weighted average of precision and recall into one only metric
Confusion Matrix	It is used to evaluate the accuracy of the classification. It will be shown as a heatmap, rather than as a numeric value due to the type of classification carried out (multi-class).
Time	The time the algorithms take to perform will be measured and recorded as wall time.

3.2 Experimental results

The table below presents the results after executing the three different models. The KNN model has not succeeded when performing the classification compared to the DT or the RF. Furthermore, the computing time of both K-NN and RF are not optimal; the RF is taking more than three minutes to execute. The DT model has proven to be the most accurate when performing the classification task and taking the shortest time to provide an output.

Table 4. Experimental results 1

Experimental Results 1						
MODEL	Accuracy(training)	Accuracy (testing)	Precission	Recall	F1-Score	Wall Time
K-NN	0.70(+/-0.01)	0.26(+/-0.22)	0.2	0.19	0.19	40.4sec
DT	1.00 (+/- 0.00)	0.99(+/- 0.00)	0.98	0.98	0.98	6.02sec
RF	1.00 (+/- 0.00)	0.99(+/- 0.00)	0.98	0.98	0.98	3min53sec

As the performance of the K-NN model has not been successful, to further understand the model, a confusion matrix has been plotted using the Scikit-Learn library, as shown below.

*1e+04 = 2650

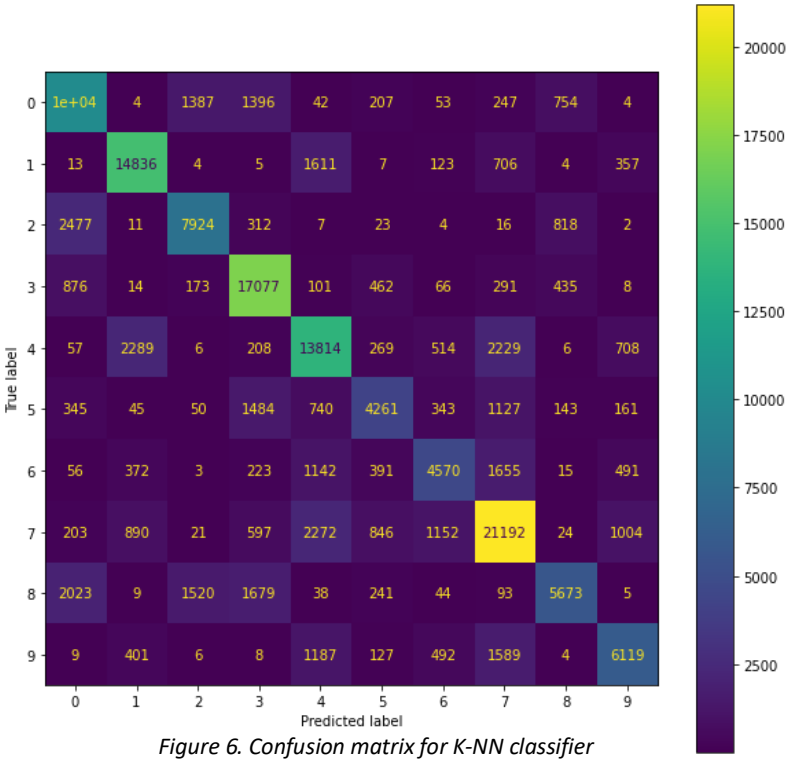

Figure 6. Confusion matrix for K-NN classifier

4. Scaling up the machine learning

The experimental results shown in section 3.2 were not desirable. The machine learning approach's main reason for scaling up is due to computing bound issue.

4.1 Approach

4.1.1 Dimensionality Reduction

Two feature selection methods were used to evaluate the importance of the features when determining the class. The Scikit-Learn library and the Pandas data frame perform a Recursive Feature Elimination and a Principal Component Analysis, which provide a feature importance ranking. Table 7. shows the rankings from both methods.

Table 5. Feature selection rankings

Encoded value	Feature Name	Ranking	
		RFECV	PCA
7	X	1	1
8	Y	1	1
9	Location	1	1
10	PdId	2	2
6	Address	3	3
3	Time	4	4
2	Date	5	5
0	Desript	6	6
1	DayOfWeek	7	7
4	Category	8	8
5	Resolution	9	9

4.1.2 Repeated Stratified K-fold Cross-Validation

The K-NN model will be executed with two different validation methods. In addition to the method mentioned in section 3.1.4, the K-NN will also be executed with a repeated stratified k-fold cross validation provided by Scikit-Learn. This method ensures a balance in the class frequencies in the train and test folds. Moreover, it helps handle the class imbalance beforementioned in the data analysis (Pedregosa et al., 2011).

4.1.3 Scikit-Learn, Joblib & Dask

The Scikit-Learn library provides all the algorithms used in this project. Many of the algorithms provided by Scikit-Learn are already written for parallel execution using Joblib. Joblib provides a transparent parallelization, inspecting and deducing which operations can be done in parallel (Joblib 2018). Furthermore, Dask can scale up the algorithms backed up by Joblib, providing an alternative Joblib backend. This alternative mainly focuses on scaling up machine learning approaches with computing bound problems(Rocklin 2015).

4.2 Deployment & Experimental results

4.2.1 Preprocessing

Before deploying the scaled-up model, some additional preprocessing was required:

1. Based on the dimensionality reduction results the number of features has been reduced from 10 to 7.

4.2.2 Experimental Results

The table below represents the results after executing the three different models after implementing the approaches mentioned in the section above.

Table 6. Experimental results 2

Experimental Results 2						
MODEL	Accuracy(training)	Accuracy (testing)	Precission	Recall	F1-Score	Wall Time
K-NN	0.89 (+/- 0.00)	0.81 (+/- 0.02)	0.78	0.77	0.78	36.2sec
K-NN (rkf)	0.89 (+/- 0.00)	0.83 (+/- 0.00)	0.78	0.77	0.78	42.2sec
DT	1.00 (+/- 0.00)	0.99(+/- 0.00)	0.99	0.99	0.99	4.79sec
RF	1.00 (+/- 0.00)	0.99(+/- 0.00)	0.99	0.99	0.99	1min24sec

The results have shown some improvement. The computing time has reduced compared to the first experimental results. The RF has reduced its computational time drastically; however, it is still not optimum. The DT has proven to be the most suitable model for this problem as it achieves e a very high accuracy in both experiments, and it can also improve the computing time. Finally, the KNN model has shown to improve its accuracy and its computing time. The change in the validation method has proven to improve the accuracy of the K-NN model, yet there was not much improvement in the computing time. The following confusion matrix was plotted for a better comparison between the same KNN model before and after scaling up.

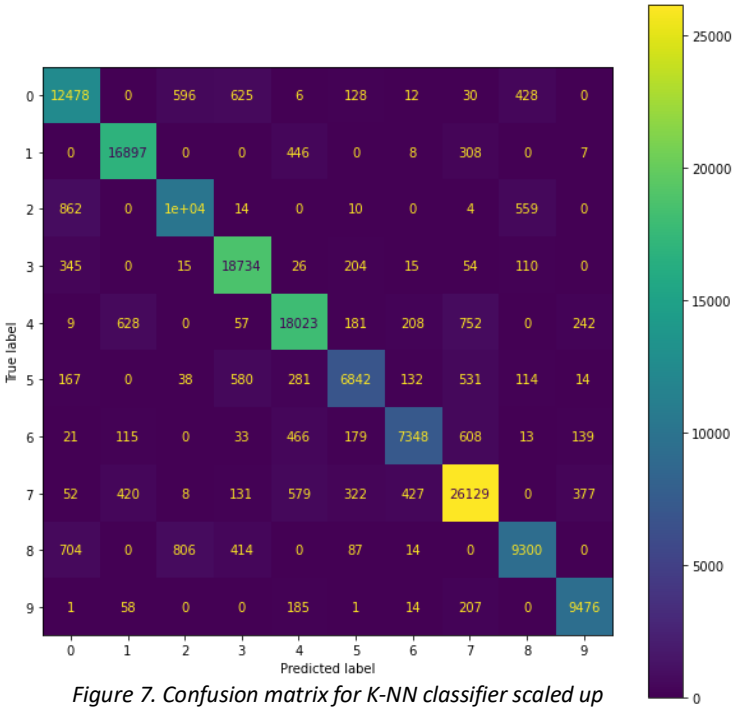

Figure 7. Confusion matrix for K-NN classifier scaled up

5. Conclusions and discussions

The application of data science tools and technologies in the law enforcement area seems to be promising. This report provides a starting point for a place-based predictive policing system. After being analyzed, it has been proven that the crime data is beneficial to provide insights either for officers or, the general public. Visualization tools as Seaborn and Matplotlib are handy to plot graphs for more straightforward interpretation. Knowledge extraction via machine learning has also been successful. The Scikit-Learn classifiers, made it possible to identify a model which can perform classification tasks accurately and in a short time. As shown in both experimental results presented in the previous sections, the tree models take the lead at the classification accuracy; however, the RF has proven to have computational issues. All the models have shown improvement after scaling up the initial approach with the Joblib and Dask methods. Finally, a more in-depth exploratory data analysis should be done. Further experiments should be carried out with a multi-label classification approach for the system to predict more specifically the type of crime that might fall within a district on a more specific location. If a model successfully performs with historical data, the last step would be to evaluate its performance with real-time data and hopefully deploy it among law enforcement agencies. However, some deployment challenges may be encountered, according to Groff and La Vigne (2002). Table 7. Provides an overview of the challenges.

Table 7. Deployment challenges

Challenge	Description
Focusing on prediction accuracy instead of tactical utility	Working with historical data does not provide the officers with any new insight. Identifying hotspots are not as accurate due to the large area they cover. The main challenge is to accurately predict on a more specific location.
Relying on poor-quality data	This challenge relates to the velocity characteristic mention in section 2.2. Furthermore, challenges related to data omission, bias and usefulness are also presented.
Misunderstanding the factors behind the prediction	Predictive systems do not generally highlight the factors behind their prediction. For this reason further techniques such as regression or data mining are recommended to understand such factors
Underemphasizing assessment and evaluation	It seems like this type of system lack evaluation in the effectiveness of the prediction. This could be achieved by keeping the data current; furthermore, a system could be considered effective if its deployment has been successful when working together with the law enforcement officers

Overlooking civil and privacy rights	The fact of labeling areas and people causes concerns about civil liberties and privacy rights. This challenge raises more often on person-based predictive policing.

6. References

[1] Arbab-Zavar, B., 2021. Tools and Technologies of Data Science: Big Data! [Power Point Presentation]. Bournemouth University: Brightspace. Unpublished.

[2] Costa, C., 2020. *12 Cool Data Science Projects Ideas for Beginners and Experts* [online]. Towards Data Science:Medium. Available from: https://towardsdatascience.com/12-cool-data-science-projects-ideas-for-beginners-and-experts-fc75b5498e03 [Accessed 15 Mar 2021].

[3] Groff, E.R., La Vigne, N.G., 2002. *Forecasting the future of predicting crime mapping*[online]. Research Gate. Available from: https://www.researchgate.net/publication/228793764_Forecasting_the_future_of_predictive_crime_m [Accessed 3 April 2021].

[4] Joblib, 2018. *Joblib: running Python functions as pipeline jobs*[online]. Joblib.readthedocs.io. Available from: https://joblib.readthedocs.io/en/latest/why.html#benefits-of-pipelines [Accessed 12 Apr 2021].

[5] LatLong.net, 2012. *San Francisco, CA, USA*[online]. Available from: https://www.latlong.net/place/san-francisco-ca-usa-594.html#:~:text=The%20latitude%20of%20San%20Francisco,%C2%B0%2025'%2052.6692''%20W.San%20Francisco,%20CA,%20USA [Accessed 7 April 2021].

[6] Pedregosa, F., et al., 2011. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*[online]. 12, 2825-2830. Available from https://scikit-learn.org/stable/modules/preprocessing_targets.html#preprocessing-targets). [Accessed 2 Apr 2021].

[7] Pedregosa, F., et al., 2011. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*[online]. 12, 2825-2830. Available from: https://scikit-learn.org/stable/modules/cross_validation.html [Accessed 2 Apr 2021].

[8] Pedregosa, F., et al., 2011. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*[online]. 12, 2825-2830. Available from: https://scikit-learn.org/stable/modules/generated/sklearn.metrics.accuracy_score.html [Accessed 5 Apr 2021].

[9] Pedregosa, F., et al., 2011. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*[online]. 12, 2825-2830. Available from: https://scikit-learn.org/stable/modules/generated/sklearn.metrics.precision_score.html [Accessed 5 Apr 2021].

[10] Pedregsa, F., et al., 2011. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*[online]. 12, 2825-2830. Available from: https://scikit-learn.org/stable/modules/generated/sklearn.metrics.recall_score.html [Accessed 5 Apr 2021].

[11] Pedregosa, F., et al., 2011. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*[online]. 12, 2825-2830. Available from: https://scikit-learn.org/stable/modules/generated/sklearn.metrics.f1_score.html [Accessed 5 Apr 2021].

[12] Pedregosa, F., et al., 2011. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*[online]. 12, 2825-2830. Available from: https://scikit-learn.org/stable/modules/generated/sklearn.metrics.confusion_matrix.html [Accessed 5 Apr 2021].

[13] Pedregosa, F., et al., 2011. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*[online]. 12, 2825-2830. Available from: https://scikit-

learn.org/stable/modules/generated/sklearn.metrics.confusion_matrix.html [Accessed 5 Apr 2021].

[14] Pedregosa, F., et al., 2011. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*[online]. 12, 2825-2830. Available from: https://scikit-learn.org/stable/modules/cross_validation.html#repeated-k-fold [Accessed 5 Apr 2021].

[15] Perry, W.L., Mclnnis, B., Price, C.C., Smith, S.C., Hollywood, J.S., 2013. *Predictive Policing – The Role of Crime Forecasting in Law Enforcement Operations*[online]. RAND corporation. Available from: https://www.ojp.gov/pdffiles1/nij/grants/243830.pdf [Accessed 20 Mar 2021].

[16] Rocklin, M., 2015. *Dask: Parallel Computation with Blocked algorithms and Task Scheduling*[online]. Proceedings of the 14th Python in Science Conference: 130-136. Available from: https://ml.dask.org/joblib.html [Accessed 10 Apr 2021].

[17] Rocklin, M., 2015. *Dask: Parallel Computation with Blocked algorithms and Task Scheduling*[online]. Proceedings of the 14th Python in Science Conference: 130-136. Available from: https://distributed.dask.org/en/latest/ [Accessed 12 Apr 2021].

[18] Rocklin, M., 2015. *Dask: Parallel Computation with Blocked algorithms and Task Scheduling*[online]. Proceedings of the 14th Python in Science Conference: 130-136. Available from: https://examples.dask.org/machine-learning.htm[Accessed 13 Apr 2021].

Dataset downloaded from the following Kaggle source:

Sharma, R., 2019. *San Francisco Crime Dataset (1)[*online]. Kaggle. Available from: https://www.kaggle.com/roshansharma/sanfranciso-crime-dataset

Appendix 1.

Table 8. Features further description

Feature Name	Data Type	Duplicates	Null Values	Descrption
IncidentNum	int64	None	None	Identification number given to all instances
Category	object	None	None	Crime category
Descript	object	None	None	Brief description of crime
DayOfWeek	object	None	None	Day of the week (Mon-Fri)
Date	object	None	None	Date of crime (mm/dd/yyyy)
Time	object	None	None	Time (hh:mm)
PdDistrict	object	None	1	District name
Resolution	object	None	None	Crime outcome
Address	object	None	None	Full address where crime happened
X	float64	None	None	Latitud
Y	float65	None	None	Longitud
Location	object	None	None	Latitud and Longitud putten together
PdId	int64	None	None	District identification number

Appendix 2.

Table 9. Class distribution details

Class Number	Class Name	Number of ocurrencias	%
Class 0	BAYVIEW	14303	9.519%
Class 1	CENTRAL	17666	11.758%
Class 2	INGLESIDE	11594	7.716%
Class 3	MISSION	19503	12.980%
Class 4	NORTHERN	20100	13.378%
Class 5	PARK	8699	5.790%
Class 6	RICHMOND	8922	5.935%
Class 7	SOUTHERN	28445	18.769%
Class 8	TARAVAL	11325	7.537%
Class 9	TENDERLOIN	9942	6.617%