Models Comparison for Criminals Arrest Classification

Tian Wu

University of California San Diego
San Diego, California
tiw118@ucsd.edu

Xinmeng Li

University of California San Diego
San Diego, California
xil397@ucsd.edu

Zhuojun Chen

University of California San Diego
San Diego, California
zhc111@ucsd.edu

ABSTRACT

This report is about the prediction of arrests for criminals in the reported crime in the City of Chicago based on the dataset upon the incidents of crimes in 2017 from the Chicago Data Portal. The prediction mainly depends on the feature IUCR code, the Primitive Type, the community area, and the District. We perform the prediction via eight different models. This prediction is helpful for the future deployment of the police in the City of Chicago.

Keywords
Prediction, Model Comparison, Arrest, Exploratory Data Analysis, Primitive Type.

1. INTRODUCTION

Public Safety draws more and more public attention in the United States, especially in the area with a high incidence of crimes like the City of Chicago. Although in recent years the number of reported incident of crimes reduces in the City of Chicago, the crimes are still a severe problem for public safety. To make people live in order, we decided to predict whether the criminal is arrested or not when the incidence of crime occurs.

2. EXPLORATORY DATA ANALYSIS

In this section, we list some statistics and properties for the dataset and perform the exploratory analysis.

2.1 Dataset analysis

The dataset contains the incidents of crimes that reported in the City of Chicago sorted by year and is extracted from the Citizen Law Enforcement Analysis and Reporting system of Chicago Police Department[1]. In this project, we specifically look at the data in year 2017, which is in the file Crime_2017.csv.

The file Crime_2017.csv contains 267824 data points and 22 features and the key features are:

- ID: Identifier for the crime incidence.
- Date: Date when the incident occurred.
- Block: Address where the incident occurred
- IUCR: The Illinois Uniform Crime Reporting code.
- Description: The secondary description of the IUCR code.
- Location Description: Type of site where incident occurs.
- Arrest: Whether an arrest was made.
- Domestic: Whether the incident was domestic-related.
- District: The police district where the incident occurred.
- Community Area: the community area where the incident occurred. There are 77 Community Areas in the City of Chicago.
- Location: The location where the incident occurred in a format that allows for creation of maps and other geographic operations on this data portal.

2.2 The Crime Occurrence Analysis

In this section, we analyze which factor contributes to the occurrence of crimes. We mainly examine the time and the location the incidents of crimes occur.

2.2.1 The Crime Occurrence Based on Month

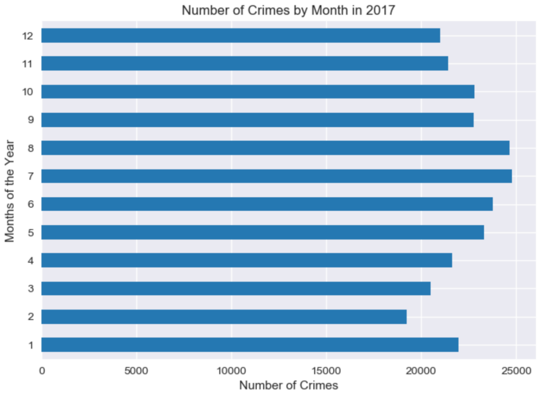

Figure 1: Number of Crimes by Month in 2017

From Figure 1, we observe that the occurrence of crimes reaches the peak in summer. In particular, there are 24785 crimes occurs in July and 24643 crimes occur in August.

2.2.2 The Crime Occurrence Based on Date of the Week

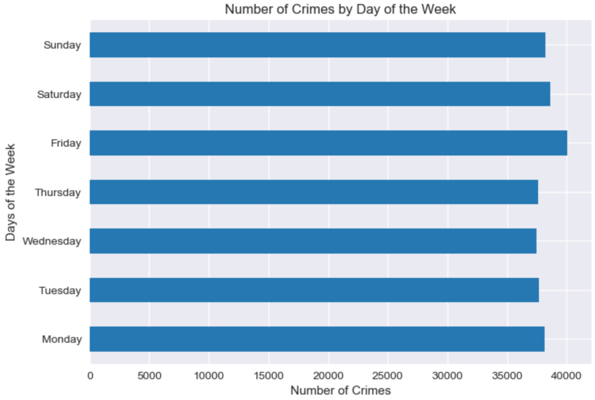

Figure 2: The Crime Occurrence by Date of the Week in 2017

From Figure 2, we observe that the crime occurrence reaches the peak on Friday with number 40062. The crime occurrences on the rest days of the week are almost the same.

2.2.3 The Crime Occurrence Based on Hour

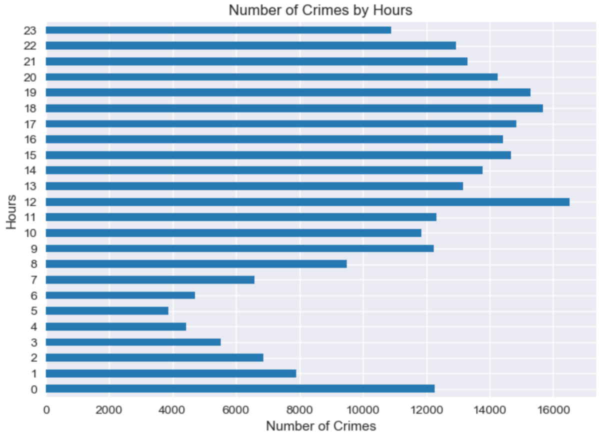

Figure 3: The Crimes by Hours in 2017

From Figure 3, the crime occurrence reaches the peak at 12:00-12:59 PM at noon yearly and obtains the second peak in the evening.

2.2.4 The Crime Occurrence Based on the Community Area

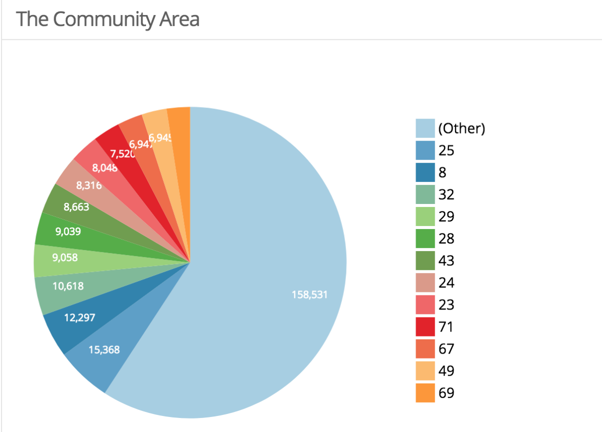

Figure 4: The Crime Occurrence by Community Area in 2017

From Figure 4, the crime occurs in some particular community area more often than others. In particular, the incidents of crime happens more in Community 25, 8, 32 than others.

2.2.5 The Crime Occurrence Based on the Location

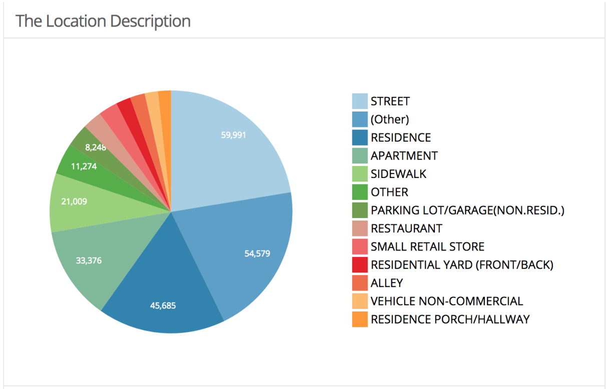

Figure 5: he Crime Occurrence by Location Description in 2017

From Figure 5, most of the crimes happen on the street, residence and the apartment. To be more specific, there are 54579 crimes on the street, 45685 crimes in residence, and 33376 crimes in the apartment.

2.3 The Arrest Ratio Analysis

In this section, we analyze which factor contributes to the criminal to be arrested. We mainly investigate the type of the crime, the location, and the police district.

2.3.1 The Ratio for Arrest

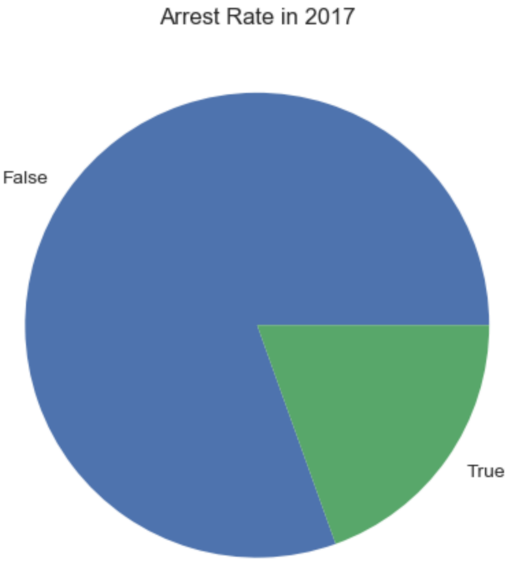

Figure 6: Arrest Rate in 2017

From Figure 6, the police arrest 80.5% of criminals in the City of Chicago in 2017.

2.3.2 The Arrest Ratio based on Primitive Type

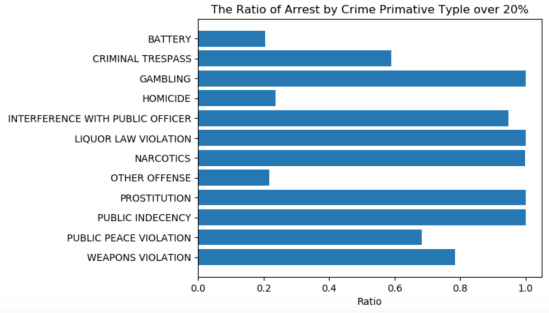

Figure 7: The Arrest Ratio by Crime Primitive Type over 25%

In this dataset, there are 32 primitive types of crimes, and 26 of them have over 100 person-time occurrences. From the Figure 7, there are 12 primitive types of crimes have over 20% arrest rate, in which the type of Gambling, Liquor Law Violation, Prostitution, and Public Indecency have 100% rate to be arrested.

2.3.3 The Arrest Ratio Based on IUCR

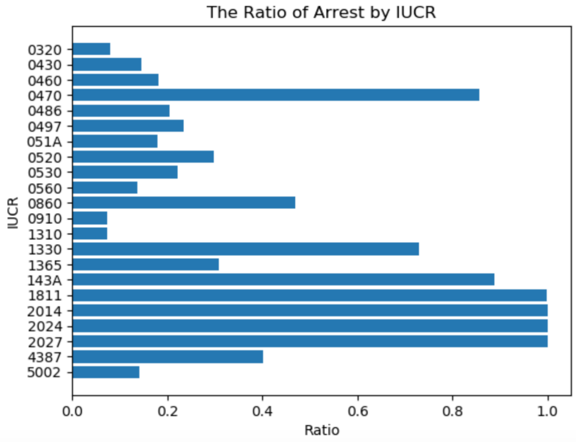

Figure 8: The Ratio of Arrest by IUCR

There is a significant difference among the arrest ratios in 331 IUCR codes. From the Figure 8, considering the 22 IUCR codes with high crime occurrence where the incidence > 1000, we found that if the crime is related the IUCR code 1811, 2014, 2024, and 2017, the criminal is 100% guaranteed to be arrested.

2.3.4 The arrest Ratio Based on Location Description

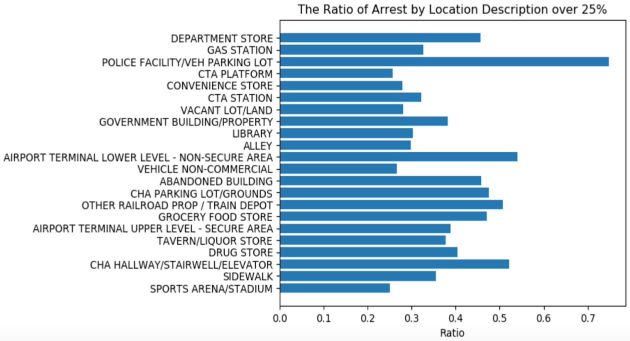

Figure 9: The Ratio of Arrest by Location Description over 25%

There are 128 locations where crime incidents occurred in total in the City of Chicago in 2017. Among these 128 locations, there are 67 locations that crime occurs over 100 person-time, and the rests are outliers. From Figure 9, we found that 22 locations that the arrest ratio is over 25% in those high-crime-occurrence locations. From the Figure 9, we notice that the crime occurs in the Police Facility / VEH Parking Lot has 0.7486 rates to be arrested.

2.3.5 The arrest Ratio Based on Community Area

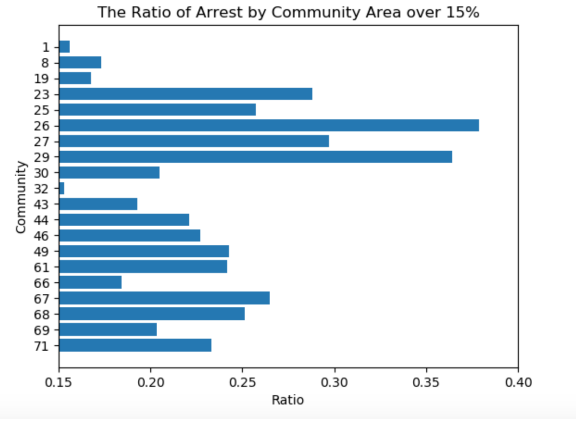

Figure 10: The Ratio of Arrest by Community Area Over 25%

Among the 77 Community Areas, 22 community areas have the arrest rate between 0.1 to 0.15 and 25 communities have high crime occurrences where the crime occurrence > 4000 person-time. From Figure 10, in the high-crime-occurrence communities, 20 community areas have the arrest ratio over 0.15. From the Figure 10, we observe that Community 26 and Community 29 have high arrest ratio, 0.378 and 0.364 separately.

2.3.6 The arrest Ratio Based on District

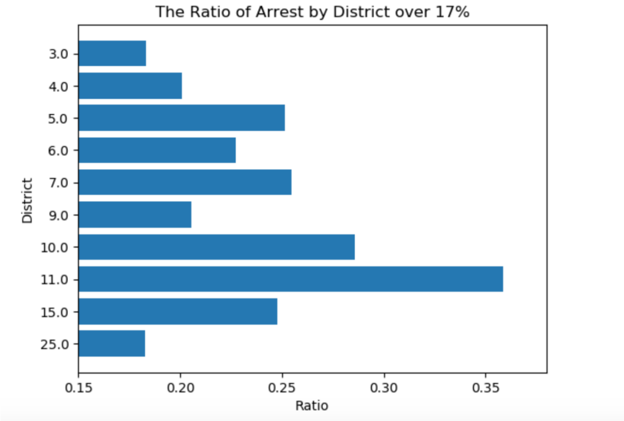

Figure 11: The Ratio of Arrest by District

10 police districts have the arrest ratio between 0.1 to 0.15 among the 23 districts. The FIgure 11 shows the correlation between the district and the ratio of arrest, and we found that the crime conducted in district 11 has a relatively high probability to be arrested.

2.3.7 The arrest Ratio Based on Domesticity

Domestic	Arrest	Ratio
False	False	0.80074589
	True	0.19925411
True	False	0.82712223
	True	0.17287777

Table 1: The Arrest Ratio Based on Domesticity

Table 1 show that there is no significant difference for the criminal to be arrested whether the crime is domestic or non-domestic.

3. PREDICTIVE TASK

Our predictive task is ---- given all information regarding a case, we predict whether an arrest has taken place; that is, we predict whether the suspects conducting the crime should be arrested. To this end, we randomly sampled 100k cases from our data; then we used the first 70k for training, next 20k for validation, the final 10k for testing.

3.1 Feature

From figure 6 of the previous exploratory data analysis, we found that roughly 20 percent of cases have successful arrests. From figure 8 and figure 7, we found that there are high correlations between the arrest ratio and the primitive type and IUCR; from table 1, there is a low correlation between the arrest ratio and domesticity; from figure 10, 11, and 9, there are somewhat significant correlations between the arrest ratio and the community area, and location description. Therefore, based on our analysis, we decide to use "IUCR," "Primitive Type," "District," and "Community Area" as our features.

3.2 Pre-process data

Since all the features we choose to use categorical variables only, we decided to use one hot encoding for all of them. For time series data, we extracted month, date, hour, and weekday information from our time data and process them with one hot encoding. Moreover, we perform the one hot encoding on all other features. Finally, we standardized all of our feature vectors to transform our variable to similar scales; accordingly, the feature vector for the training, validation and test set has size 70kx670, 20kx670 and 10kx670 respectively.

3.3 Evaluating classifiers:

The following metrics evaluate the performance and validity of the models on the predicted value in the test set:

- Accuracy: $\frac{correct\ predictions}{number\ of\ total\ predictions}$
- True Positive Rate(TPR): $\frac{true\ positives}{labeled\ positives}$
- True Negative Rate(TNR): $\frac{true\ negatives}{labeled\ negatives}$
- Balance Error Rate(BER): $1 - \frac{1}{2} \cdot \frac{TPR}{TPR+TNR}$

- Precision: $\frac{true\ positives}{predicted\ \ positives}$

Ideally, a well-performed prediction should have a high score on accuracy, TPR, and precision and having a low score on BER.

3.4 Baseline

Naive baseline: Since about 80% of the cases have arrest type with "false," the most naive baseline would predict "false" in every case, resulting in an accuracy of 80.83% on the validation set. However, this solution is not helpful in reality because it has a TPR of 0 and a BER of 0.5.

Improved baseline: From figure 7, we noticed cases with primary type being "prostitution," "liquid law violation," "gambling," and "public indecency" have high arrest ratio; hence our modified baseline solution predicts that, given a case, if its primary type is among those categories, we predict true and otherwise we predict false. This baseline has significantly higher performance than the naive baseline: its accuracy is 85.51%, TPR is 23%, and its balanced error rate is 38.5% on the validation set.

3.5 Models

Besides classification models taught in class such as logistic regression and SVMs, we also compare the performance of additional models including decision tree, random forest, gradient boosting, k nearest neighbors and multilayer perceptrons.

3.5.1 Logistic Regression

Strengths: First, Logistic Regression is one of the most widely used algorithms in binary classification. Our predictive task is to classify whether a criminal should be arrested or not, which is a perfect fit for Logistic Regression. Second, this model is robust to noise and has a low risk of overfitting. Our data have many attributes so that might be some noises. Logistic regression avoids the negative impact of noises in the large-scale data. Third, it has fast speed and thus performs efficiently on the large-scale dataset. As a result, comparing with a time-consuming model like KNN, Logistic Regression is more practical for us to tuning parameters and to train models for multiple times. Fourth, it returns probabilistic result. Consequently, we can tune the model by adjusting the classification threshold. Finally, if we want to extend the size of the training data, the model is updated easily and quickly.

Weaknesses: Logistic Regression assumes the data is linearly separable and we deduce that it only works well when there is a small portion of categorical variables in features. However, our original feature vector contains many categorical variables with more than 10 categories each. Thus, we are not sure whether there exists an accurate linear decision boundary.

3.5.2 SVM

Strengths: Comparing to logistic regression, SVM performs well on classification no matter the data is linear or not. This property guarantees that the failure risk of this model on our data is very low. SVM is also robust to overfitting in high-dimensional space, which is meaningful for us, as some of our features have more than 100 categories.

Weaknesses: SVM, especially with nonlinear kernels such as RBF, is usually not scalable due to its memory intensity. However, we have 70k training data, which requires a large amount of time and computation memory to tune and compare multiple SVM models to find the optimized one.

Tuning Parameters: We first compare the performance of Linear SVC with C values 0.01, 0.1, 1, and observe that the optimized C value is 0.01. Then we attempted to compare the linear kernel with RBF kernel, but we failed due to the time and memory limit for SVM with RBF kernel.

3.5.3 Decision Tree

Strengths: We choose decision tree as one of our models for three reasons: First, the decision tree performs useful classification by learning the decision rules on training data. The Figure 12 is an example of how does the tree structure looks like.

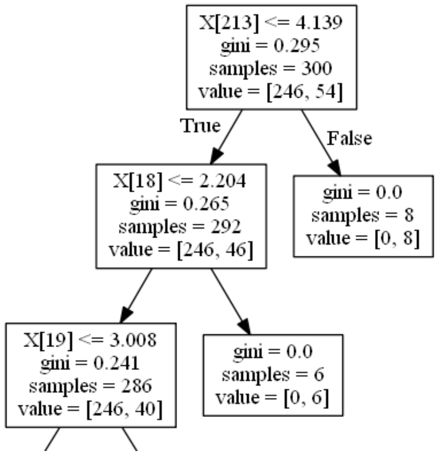

Figure 12: Decision Tree Structure

It is also robust to outliers, which is meaningful as our data is large scale and has many attributes. Second, compared with logistic regression, the decision tree is capable of modeling nonlinear decision boundaries as it automatically learns nonlinear feature interactions and it has a hierarchical structure. Third, the decision tree is fast, so we are capable of comparing models with various parameter values.

Weaknesses: Since we have 670 columns in the feature, a single decision tree will go deep and build many branches, which increases the risk of overfitting.

Tuning Parameters: We compare the performance of this classifier on two criterions: gini(minimizing the impurity) and entropy(based on information gain). Gini (attributes are assumed to be continuous) performs better than entropy (attributes are assumed to be categorical) because although most of our feature variables are categorical, we normalize our feature vectors, which cause the attributes values fed to the classifier become continuous.

3.5.4 Random Forest

Strengths: Random Forest, as an ensemble of independently randomized decision trees, reduces the probability of overfitting by averaging the prediction results of a diverse set of trees. Therefore, the random forest is supposed to have higher accuracy than an individual decision tree. This model is also fast as it allows parallel computing, which makes our training process on the large-scale data more efficient.

Weaknesses: Compared with the decision tree, the total bias of random forest is higher, as it trains independent trees on random samples of data. However, the final prediction result is the average of all individual trees, which alleviate the increased bias by reducing variance. Therefore, the accuracy of the random forest is always slightly higher than the decision tree.

3.5.5 Gradient Boosting

Strengths: Each new tree built in gradient boosting attempts to corrects errors occurred in the previous model, and thus performs well on both categorical and continuous features. Moreover, compared with random forest, gradient boosting provides more hyperparameters for tuning.

Weaknesses: This algorithm is slower than random forest because it is impossible to compute multiple trees simultaneously due to its sequential property.

Tuning Parameters: We compare the performance of this model on different values of learning rate, the number of boosting stages, and the maximum depth of an individual regression estimator. Among learning rates of 0.01, 0.1, 1, 2 and 5, the model performs best at 1. Among n_estimators of 10, 30, 50, 60 and 100, the model performs best at 50. Among max_depth of 2,3,4 and 5, the model performs best at 2.

3.5.6 K Nearest Neighbors

Strengths: KNN does not make any assumption about the linearity of data so that it is a qualified classifier for data with unknown linearity.

Weaknesses: This model requires high memory because it stores all of the training data. It is slow in both training and prediction process.

3.5.7 Multi-layer Perceptrons

Strengths: Generally, it performs well on any types of data and has high fault tolerance.

Weaknesses: This model is a black box algorithm, and there is no way for us to interpret the classifier. The computation can be expensive for multiple layers and large dataset.

4. EXPERIMENT

We applied the model of baseline, decision tree, random forest, gradient boosting, KNN, SVM, logistic regression, and MLP, the results are shown section 4.1.

4.1 Results

4.1.1 Accuracy Evaluation

By the formula for accuracy, we calculate the accuracy on each model. We found that Logistic Regression perform best on the test data and its speed is relatively fast.

Model	Train Accuracy	Validation Accuracy	Test Accuracy	Speed
Baseline	0.853	0.8483	0.8553	Trivially Fast
Decision Tree	0.9997	0.8443	0.8422	Fast
Random Forest	0.9852	0.8861	0.8902	Fast
Gradient Boosting	0.8920	0.8882	0.8917	Medium
KNN	0.8766	0.8386	0.8419	Slow
SVM	0.8948	0.8902	0.8935	Medium to Slow
Logistic Regression	0.8943	0.8915	0.8956	Fast
MLP	0.9019	0.8935	0.8948	Fast

Table 2: Accuracy Evaluation

4.1.2 Precision, BER, TPR, TNR Evaluation

We calculate Precision, BER, TPR and TNR on each model. We found that all the models perform better on the TPR and BER than the baseline.

Note that BER is a very worthwhile metric because it evaluate the error in the performance and our models have significant reduction on the error compared to the baseline model. Moreover, TPR is also a meaningful metric because true positive means that the ratio between the predicted arrest and the labeled arrest. From Table3, we observe that all our models have a significant improvement for TPR than the baseline model.

Models	Precision	BER	TPR	TNR

Baseline	1	0.385	0.230	1
Decision Tree	0.601	0.254	0.587	0.904
Random Forest	0.868	0.249	0.522	0.981
Gradient Boosting	0.850	0.238	0.547	0.976
K Nearest Neighbors	0.618	0.2449	0.5978	0.9124
SVM	0.865	0.238	0.545	0.979
Logistic Regression	0.832	0.240	0.546	0.974
MLP	0.8294	0.2298	0.568	0.9723

Table 3: Precision, BER, TPR, TNR Evaluation

Model	Train acc - Baseline acc	Validation acc - Baseline acc	Test acc - -Baseline acc
Baseline	0	0	0
Decision Tree	0.1467	-0.004	-0.0131
Random Forest	0.1322	0.0378	0.0349
Gradient Boosting	0.039	0.0399	0.0364
K Nearest Neighbors	0.0236	-0.0097	-0.0134
SVM	0.0418	0.0419	0.0382
Logistic Regression	0.0413	0.0432	0.0403
MLP	0.0489	0.0452	0.0395

Table 4: Baseline Difference

4.2 Model Evaluation

According to the Table 2, Table 3 and Table 4, we not only compare the performance and validity across models but also compare our expectation of each model with the observation.

4.2.1 Decision Tree

Decision tree is an unsuccessful model with the issue of overfitting, as its training accuracy almost reaches one but its validation and testing accuracies are lower than the baseline model. We expected this issue and mentioned in section 3.5.3 The true positive rate of the decision tree is very high, with a sacrifice of true negative rate.

4.2.2 Random Forest and Gradient Boosting

Random forest and Gradient Boosting are successful models. Although random forest reduces the possibility of overfitting compared with an individual decision tree, its training accuracy is still much higher than the rest of algorithms. Gradient boosting has higher validation and testing accuracies, and it has higher TPR and lower BER but lower TNR and lower Precision than random forest due to the difference these two ensembles.

4.2.3 K Nearest Neighbors

KNN is an unsuccessful model because KNN performs worse than the baseline model. KNN is sensitive to irrelevant attributes and outliers. Our large-scale data is highly likely to have some outliers, and we have so many attributes that make it more difficult for KNN to perform classification in high dimensional space.

4.2.4 SVM & Logistic Regression

SVM and Logistic Regressions are successful models. These two models perform better than our expectation. We were concerned that a large number of categorical variables in our features were likely to lead the failure of these two linear classifiers. (i.e., As we only train and test on LinearSVC, we use SVM with linearity assumption). In fact, these two classifiers made accurate predictions. They have very high accuracies, high precision, high true negative rate, and low balance error rate, which implies that our data is linearly separable.

4.2.5 Multi-layer Perceptrons

MLP is a successful model. We expected MLP to perform well on classification, as deep learning usually has high accuracy and validity as long as we use the appropriate hyperparameters values.

4.3 Issues and Failure Attempt

There are three main issues and failures in our prediction process, they are:

- When we explored and analyzed the dataset, we found that there were some missing values in a few features like locations, which makes these features useless.
- The memory and time that SVM classifier with rbf kernel requires exceeds our capability. Consequently, we are unable to compare nonlinear SVM with linear SVM.

- KNN and Decision tree are our failed attempts due to their low precision and accuracy.

5. Conclusion

5.1 Model Evaluation Comparison

Although our baseline model has extremely high precision and TNR, its BER is high, and its TPR is very low, which indicates that the baseline classifier is imbalanced and thus the prediction made by the baseline model is not valid enough.

Compared with baseline models, the decision tree model and the KNN model are unsuccessful attempts, while the rest of the models have a significant improvement of accuracy. Based on the more balanced TNR and TPR, we can conclude that although two of the models fail to improve the accuracy, all of our algorithms improve the validity significantly.

Combining these three tables, we conclude that Logistic Regression and MLP are the best models to predict criminal arrest, as both of these models are fast, valid and accurate. In Logistic Regression, we use C = 1 for regularization, liblinear for optimization. In MLP, we use relu as the activation function of the hidden layer, adam for weight optimization, alpha = 0.0001 for regularization and learning rate = 0.001 for step-size of weight update.

Compared to our baseline model, our results generally have improved true positive rate and better balanced error rate. The improved true positive rate in our prediction means that our model is more confident in correctly predicting if an arrest is successful. With a higher true positive rate, fewer arrests can go undetected. Lower balanced error rate means that our model's prediction makes fewer mistakes when balancing false negative and false positive. These improvements make our predictions more realistic and more likely to correspond to real world compared to our baseline solution.

5.2 Model Parameter Explanation

By counting the number of attributes with a relatively large weight in SVM, we observe that IUCR has the most significant impact on deciding the arrest label among features that we have used, which agrees with the exploratory data analysis in section 2.3. It is reasonable because the IUCR code records the severe level for the crime, and the more severe the crime is conducted, the more likely the criminal is to be arrested. The second important feature is the primitive type of crime. Similarly, the Primitive Type is a supplement for the IUCR code; thus, it also contributes to the severity of the crime.

Districts, Community Areas, and Location Descriptions have much less impact than IUCR and Primitive Type on the final decision. The districts represent the police districts in the City of Chicago. It is rational that some districts have some shortage of police force so that there does exist a difference between different districts, but the difference should not be too huge overall. We can explain the difference in Community Areas can in the same way. As for the Location Descriptions, some locations like Police Facility / VEH Parking Lot and Airport Terminal Lower Level do have a high arrest rate than others in Figure 9. However, in reality, the number of crime conducted in these areas are relatively small. Hence the impact is not as huge as IUCR code and Primitive Type.

6. RELEVANT LITERATURE

We obtained our dataset from the Chicago Data Portal, which contains various data in the Chicago. There are many similar data which can be found in Kaggle, including:

- San Francisco Crime Classification from 1934 to 1963
- Reported crime data in London by borough and LSOA
- State-wise crime data in India from 2001
- Crime data in Los Angeles from 2010 through 2017

Crime data has been thoroughly studied for many years. After completing our predictive task, we looked through various similar data sets and researched about how others study these data. We found that most people employed exploratory data analysis by plotting different features against crime records, which are in line with our previous work. In addition, we also noticed that some[2] used heatmap to represent the relationships between time and crime type visually, we found this method very intuitive and efficient when dealing these type of data.

Regarding predictive method, we found that some state-of-art approach to predicting crime data includes: Logistic Regression, Deep learning, Support Vector Machine, Random Forest, and XGBoost. On March 2018, Fateha *et. al.*[3] showed how to use geospatial features to predict different categories of crime. On June 2018, Alexander *et .al.*[4] build deep neural networks to predict crime counts using Chicago and Portland crime data. Since we are predicting if arrest happened in a case, we can not find suitable studies for us to compare results. However, results from our previous work suggest that multilayer perceptrons and logistic regression can significantly improve our results compare to other approaches, which is similar to what many others did.

7. REFERENCES

[1] Crimes - 2017 | City of Chicago | Data Portal. (n.d.). Retrieved December 2, 2018, from https://data.cityofchicago.org/Public-Safety/Crimes-2017/d62x-nvdr

[2] Fahd, A. (2016). Understanding Crime in Chicago. Retrieved December 02, 2018, from https://www.kaggle.com/fahd09/eda-of-crime-in-chicago-2005-2016

[3] Bappee, F. K., Júnior, A. S., & Matwin, S. (2018, March 12). Predicting Crime Using Spatial Features. *Advances in*

*Artificial Intelligence Lecture Notes in Computer Science,*367-373. doi:10.1007/978-3-319-89656-4_42

[4] Alexander, S., & Deigo, K. (2018, June 5). Deep Learning for Real Time Crime Forecasting - arXiv. Retrieved December 2, 2018, from https://arxiv.org/pdf/1707.03340.pdf