**Project Report**

**Enterprise Cloud Computing and Big Data (BUDT737)**

**Project Title:** *Enhancing Law Enforcement Efficiency through Machine Learning: A Crime Classification System Using Apache Spark*

**Team Members:**

Fuad Khasiyev

Ansh Dugar

Akshat Shah

**ORIGINAL WORK STATEMENT**

We the undersigned certify that the actual composition of this proposal was done by us

and is an original work.

|  |  |  |
| --- | --- | --- |
|  | **Typed Name** | **Signature** |
|  | Fuad Khasiyev | FK |
|  | Ansh Dugar | AD |
|  | Akshat Shah | AS |

**Executive Summary:**

The utmost aim of this project is to potentially help law enforcement agencies to prepare reports promptly by correctly identifying the category of the committed crime. This can be an essential factor in budget allocation and general planning of the governmental organization responsible for law enforcement. Using the prepared models with high accuracies can assist the officers to reduce the time spent on paperwork and allocate most of their time to much more useful fieldwork to ensure community safety.

Utilizing Machine Learning Models through PySpark was at the heart of the project. The used data from Kaggle platform regarding crime in San Francisco helped us to concentrate our efforts on creating the models through vast amount of data. Leveraging Spark's Machine Learning Library (MLlib) and its Pipelines API, we were able to build a robust end-to-end solution for multi-class text classification. The project was able to identify the classification models that resulted in statistically high accuracy rates, proving the effectiveness of machine learning in categorizing the vast amount of data present in law enforcement agencies. Logistic Regression Models using Term Frequency-Inverse Document Frequency (TF-IDF) and Count Vector Features both yielded over 97% of accuracy, which was further upgraded to approximately 99.1% through development of the Cross Validation model. However, the results of the project showed that Random Forest and Decision Tree Classification Models did not perform well in accurately predicting crime categories based on the given features. The former model had only approximately 37.8%, reflecting that it's not ideal for high-dimensional sparse data, while the latter had a relatively better accuracy with 41.5%. Decision Tree models tend to create overly complex models, which can lead to overfitting, where the model learns to memorize the training data rather than generalize to new, unseen data, especially in the cases of text classification.

What is novel and interesting about our study is not just exceptionally high accuracy rates that we got through utilized classification models, but how we were able to compare the accuracy of different models in text classification context. Additionally, to go further beyond the scope of the project, we combined the Cross-Validation and Random Forest Models to achieve the highest accuracy for the classification tool. However, the computing power available for the completion of the project was not sufficient to run the sophisticated model we were trying to develop. Nevertheless, the created system can significantly help government agencies to handle crime data, making it faster to allocate trends within categories, distribute resources more efficiently, and consequently, keep communities safer.

**Data Description:**

We have sourced the dataset from <https://www.kaggle.com/c/sf-crime/data>

The testing dataset has: -

A sample size of 884,262 observations with 7 variables

Variables and their Characteristics:

Id: Numerical identifier for each observation (numerical, treated as categorical).

Dates: Timestamp of the crime incident (date-time format).

DayOfWeek: Day of the week when the crime occurred (categorical).

PdDistrict: Police district where the crime occurred (categorical).

Address: Address or location of the crime incident (textual, categorical).

X: Longitude coordinate of the crime location (numerical).

Y: Latitude coordinate of the crime location (numerical).

The training dataset has 878,049 observations with 9 variables  
Variables and their Characteristics:

Dates: Timestamp of the crime incident (date-time format).

DayOfWeek: Day of the week when the crime occurred (categorical).

PdDistrict: Police district where the crime occurred (categorical).

Address: Address or location of the crime incident (textual, categorical).

X: Longitude coordinate of the crime location (numerical).

Y: Latitude coordinate of the crime location (numerical).

Category: Predefined category of the crime (categorical, target variable).

Descript: Description of the crime incident (textual, categorical).

Resolution: How the crime incident was resolved (categorical).

Sample of the used data can be seen in *Appendix D*

**Research Questions:**

1. How do different machine learning algorithms perform in classifying crime descriptions?
2. How does the performance of the model improve with cross-validation and hyperparameter tuning?
3. What are the potential real-world applications and implications of deploying such a system in law enforcement agencies?
4. What impact does feature engineering, including techniques like TF-IDF and count vectorization, have on the performance of crime classification models?

**Methodology:**

The development of Crime Classification System revolved around the PySpark and its Machine Learning Library to efficiently process present large volume of data through capabilities of the latter tool had to offer.

Initially, the simple data cleaning and preprocessing was completed to make sure that while completing machine learning models, no unnecessary information was present. This step of the project was crucial to manage the extensive dataset within the capabilities of present computing power. Furthermore, the short exploratory data analysis helped to identify trends and patterns that can inform further analysis and modeling efforts.

Thereafter, the machine learning pipeline was constructed, comprised of several stages:

* *Regex Tokenizer:* breaking down crime descriptions into individual words for further text analysis.
* *Stopwords Remover*: eliminating common words that had little significance for the classification task.
* *Count Vectors*: transforming the text data into numerical format through count vectorization, in order to quantify the frequency of the words in the processed data.
* *StringIndexer*: converting the target variable into numerical labels to make it compatible for model training.

A screenshot of a computer

Description automatically generated

In the next stage of the project, various supervised machine learning algorithms were explored to identify the ones with high efficiency in text classification. The scope of the analysis was beyond the scope of the covered models throughout the course. The selected models included the Logistic Regression that proved to be very effective in handling large and sparse amounts of data, as well as Random Forest and Decision Tree Classifier. The Logistic Regression model was further improved through cross validation by tuning Count Vectors to achieve the highest accuracy, as it enhanced its performance and increased generalizability of the model. The significant improvement in classification accuracy through cross validation proved the potential of our system in assisting the law enforcement agencies to make informed decisions.

Finally, the project also implemented the cross-validation of the Random Forest model to increase its accuracy; yet, as it can be seen in *Figure.1,* after running for nearly 3.5 hours, the limited computational resources at hand hindered the potential of achieving the objective of higher accuracy.

**Results and Findings:**

In the first part of the analysis, the frequency of different crime categories was demonstrated through a comprehensive bar chart. It helped us to get a better visualization of the data and create a better understanding for the further analysis using the classification models. As it can be seen on *Appendix A*, Larceny and Theft was by far the most frequent crime in San Francisco.

When it comes to the results and findings of the Machine Learning models, the performance of the models was evaluated and compared based on the calculation of their accuracy – proportion of correct predictions. Our Logistic Regression model achieved an accuracy of an impressive **97.20%.** The top 10 predictions with highest probability for label 0 can be seen in *Figure. 2.* Afterwards, we implemented TF-IDF to our Logistic Regression model, where the TF-IDF is a statistical measure to evaluate the significance of a word in a document within a collection of relative documents. However, this process failed to improve the accuracy of the model as its accuracy was still approximately **97.20%.** Similarly, implementation of Count Vector Features to Logistic Regression did not improve the accuracy of the model by just resulting in a similar percentage. A table of numbers and a list of theft

Description automatically generated with medium confidence

Cross-validating the Logistic Regression proved to be a successful strategy in our attempts to maximize the efficiency of the system. Through cross-validation using Count Vectors Logistic Regression performed, we were able to fine-tune the model’s hyperparameters and significantly optimized its ability to generalize to new data. As a result, we were able to achieve nearly an additional **2% accuracy**, a significant rise considering that the initial Logistic Regression had already very high accuracy.

Random Forest and Decision Tree Classification Models failed to overcome the accuracy of the Logistic Regression, where Random Forest had 37.8%, while Decision Tree Classifier model had 41.5% accuracy. These findings proved that when it comes to datasets with many features that have low predictive power, decision trees can struggle with splitting decisions. As Random Forest model also depends on decision trees, it also performed poorly in Crime Classification System. The examples of prediction tables of Random Forest and Decision Tree classifier can be seen in *Appendices B and C*, respectively.

Overall, while Logistic Regression with cross-validation stood out for its performance, other models’ potential was evaluated within the project. Unfortunately, they failed to improve the performance of the system. Moreover, the challenge of computational limitations that was faced while applying cross-validation to the Random Forest Classification model, underscored the importance of our model selection not only on the theory, but also depending on their feasibility with resources at hand.

**Conclusion:**

To conclude, by utilizing PySpark and Machine Learning, our project has successfully demonstrated the potential to improve the accuracy of crime classification and reduce time allocated to achieve it. Through data preparation, classification model analyses and selection, optimization processes, our project successfully achieved its goal by yielding an accuracy of over **99%** by cross validating the original Logistic Regression. In comparison analysis with other classification models used within the project, the Decision Tree and Random Forest model were found to be not useful in the text classification scenario. While the project encountered constraints because of limited computational resources, particularly with the cross-validation of the Random Forest model, these challenges provided invaluable insights into the scalability and resource management aspects of deploying complex machine learning solutions.

As we look ahead, we believe that similar Crime Classification Systems will prove to be a successful representation of machine learning in law enforcement. Implementing these systems all over the country will lead to the time optimization in governmental agencies and improve the response time to the crimes as more officials will be available for fieldwork.

**Appendix A – Bar Chart of Crime Categories.**

**A graph of a number of bars

Description automatically generated**

**Appendix B – Random Forest Classifier Results**

A screenshot of a computer

Description automatically generated

**Appendix C – Decision Tree Classifier Results**

A screenshot of a computer code

Description automatically generated

**Appendix D – Sample Data Screenshots**

Test Data

A screenshot of a computer

Description automatically generated

Train Data

A screenshot of a computer

Description automatically generated