**CTEC3702: Big Data and Machine Learning**

**Assignment: Problem Specification**

**Karan Modi**

**P2761604**

(Word count 1759)

**A logo with a lion head

Description automatically generated**

Table of Contents

[1. Introduction 3](#_Toc181840182)

[2. Part A: Data Exploration and Visualization 5](#_Toc181840183)

[2.1 Data Loading and Preprocessing 5](#_Toc181840184)

[2.2 Descriptive Data Analysis 6](#_Toc181840185)

[2.3 Data Visualisation 8](#_Toc181840186)

[3. Part B: Machine Learning 10](#_Toc181840187)

[3.1 Data Preprocessing 10](#_Toc181840188)

[3.2 Model Training and Evaluation 12](#_Toc181840189)

[3.3 Model Comparison 14](#_Toc181840190)

[4. Conclusion 15](#_Toc181840191)

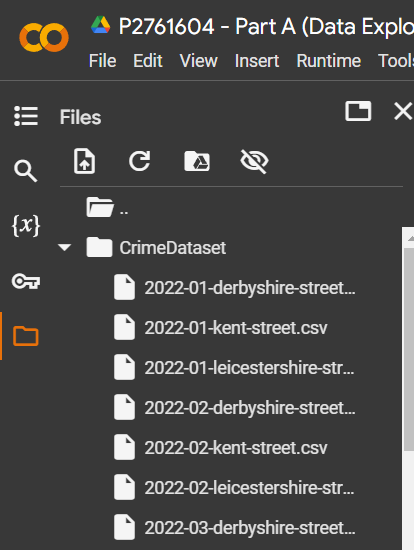
# Introduction

* **Overview of the Assignment**:

In the first part of the assignment, various crime data is taken and this data is cleaned, transformed and analyzed using Spark followed by data visualization. This data is pre-filtered for objects including county names to be included in distributed data sets, and then explored with Spark-SQL over county-wise crime density, type of crime and monthly trends. Conclusions are presented through additional graphic displays such as the line, bar and pie charts.

The second part is more about constructing a loan eligibility prediction model using Apache Spark’s MLlib library. While loading the loan dataset, missing value imputation is performed on pre-processing stage at the basic level for each categorical and numerical column with the mode value. The readings are then preprocessed for machine learning, with possible feature engineering for excellence in model identification. With MLlib, Spark offers the necessary scalability of a package designed for actual-world predictive use.

* **Datasets Overview**: Two Datasets have been used in this assignment within two parts respectively, Part one has been done by using the crime dataset (Derbyshire, Leicestershire and Kent Crime Datasets.)
* **Crime Dataset**: A dataset from the police forces of Derbyshire, Leicestershire, and Kent covering crimes from 2022.



* **Loan Prediction Dataset**: This dataset is done containing personal information of individuals. And based on those details, an algorithm can analyse that a person is eligible for loan approval or not.

A screenshot of a computer

Description automatically generated

# Part A: Data Exploration and Visualization

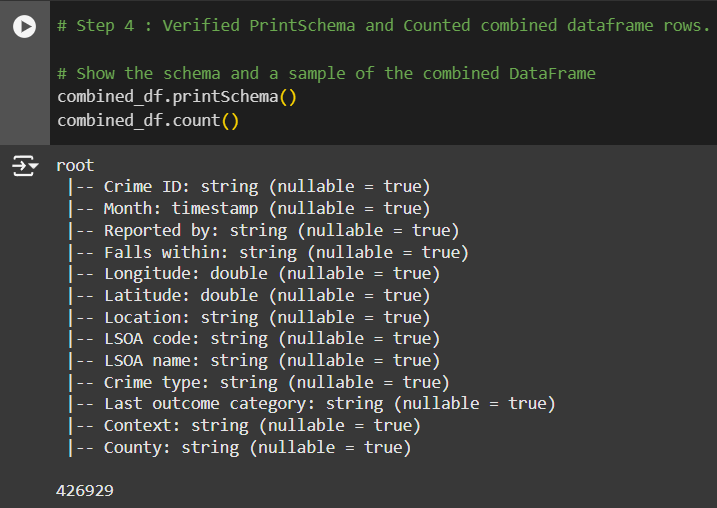
## 2.1 Data Loading and Preprocessing

* **Data Loading**:

To examine the information about crime statistics for Derbyshire, Leicestershire, and Kent. I imported data from CSV files using Apache Spark. First, I created Spark session, and then I loaded each dataset into Spark. Making sure that Spark correctly inferred schema, I followed that up by joining the data frames to form a single data frame called combined\_df. This merged Data Frame provided a common structure of data for analysis, and each row contains a reported crime instance.

* **Standardising County Names**:

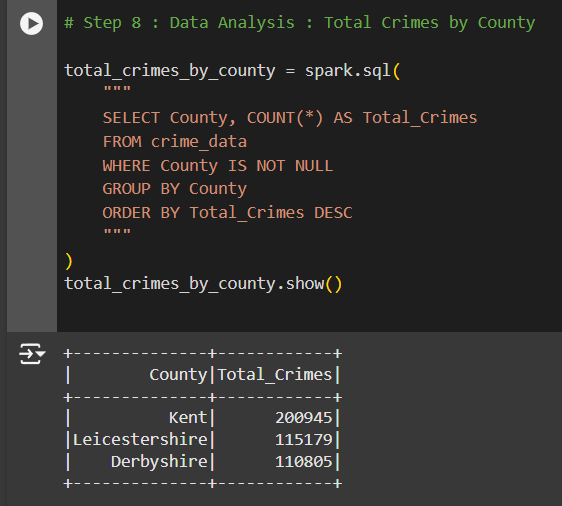
There were also issues with the dataset in that at the start of the dataset, police force names were listed in the “Reported by” column instead of county names. To standardise I condensed the named police forces and their respective affiliations and equated them for clarity (for example, the name “Derbyshire Constabulary” was reduced to “Derbyshire”). This step enabled discharge of each crime’s county in the last Data Frame in a very obvious manner. The resulting Data Frame added the county name as a new column to the data set and presented a clearer picture of all observed variables and counties for analysis purposes.



## 2.2 Descriptive Data Analysis

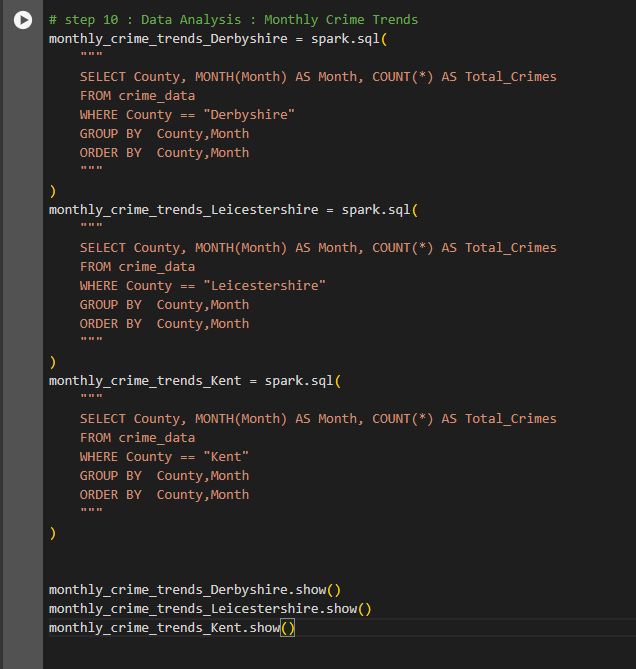
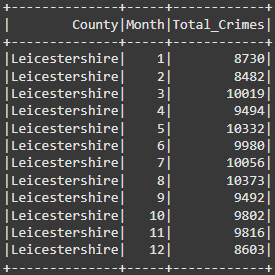
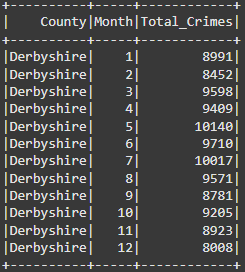
* **Total Crimes by County**:

To explain the distribution of crimes across the counties, total crime count for each county was computed using Spark SQL. This analysis established that [example results as shown below: Kent committed the greatest number of crimes, followed by Leicestershire then Derbyshire].

.

* **Monthly Crime Trends**:

I compared the crime trend per month in order to identify which of the months, have more or less cases of crime in the counties. This was done by sorting data by region and by month, and the general outcomes were as follows; [example of the result, for instance, June had the highest crime rate in Derbyshire while January had the lowest in Leicestershire]. This pattern was useful for defining whether certain type of crime was on the rise during different seasons or months.

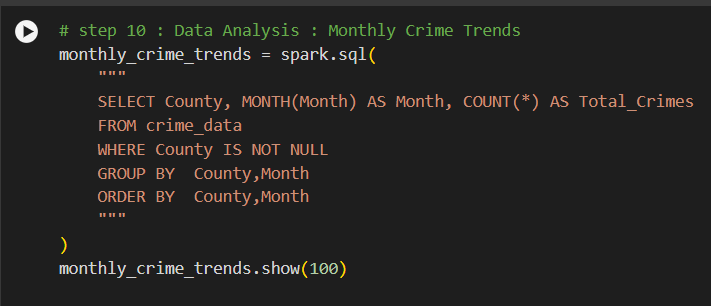


A screenshot of a computer screen

Description automatically generated

* **Crime Type Analysis**:

Thus, for each county, I assigned each crime type a rank according to its frequency. As evidenced in this analysis, [For instance, “Violence and Sexual Abuse” emerged as most prevalent, “Bicycle Theft or Robbery” as least frequent]. Knowledge of the distribution of crime types is helpful when it comes to distribution of resources.

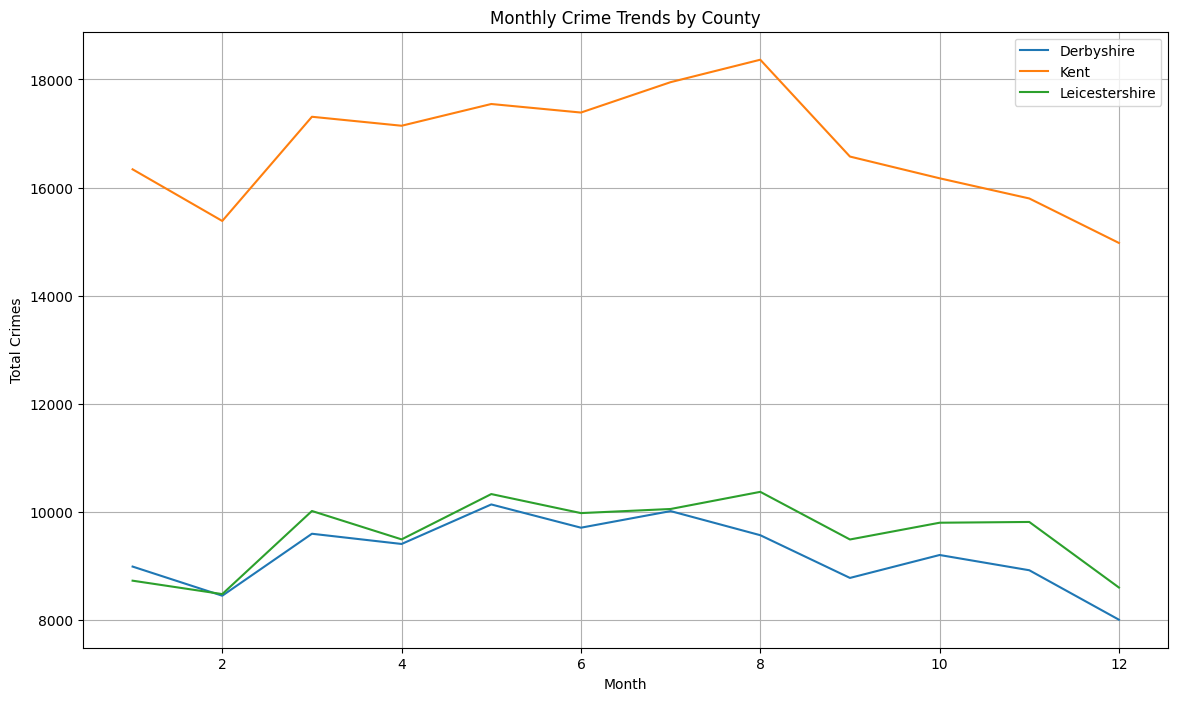
A screenshot of a computer screen

Description automatically generated

## 2.3 Data Visualisation

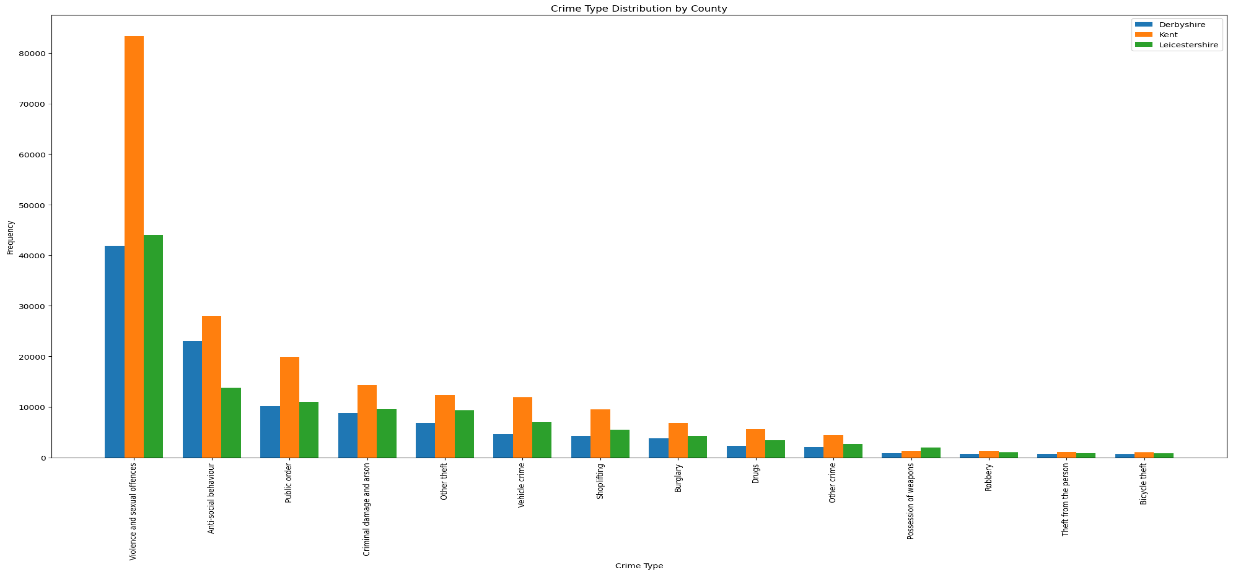
* **Monthly Crime Trends (Line Plot)**:

All the monthly crime data are displayed graphically in a line plot format and coloured differently for Derbyshire, Leicestershire, and Kent. The plot illustrates [notable trends, e.g., in a lowest Day light time, April, May].



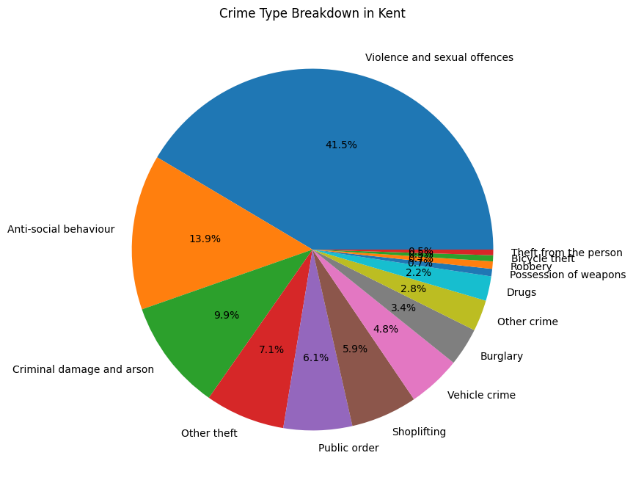
* **Crime Type Distribution (Bar Chart)**:

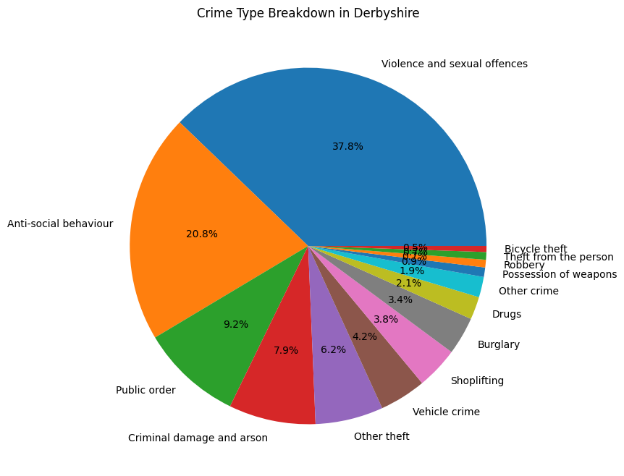
Another example of analysis that a bar chart provides is [example insight, e.g., comparing the types of crimes in Kent, Derbyshire, Leicestershire].



* **Crime Type Breakdown by County (Pie Charts)**:

Pie charts for each county display the percentage breakdown of crime types.

 Most of the crimes happened for Violence and Sexual Offences.



A pie chart with different colored circles

Description automatically generated

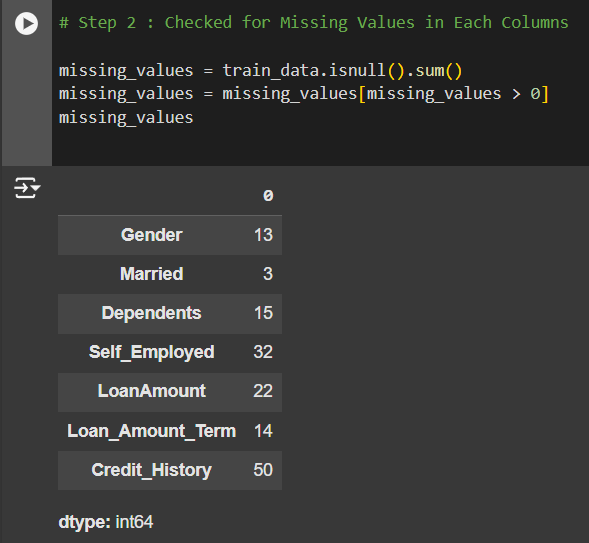
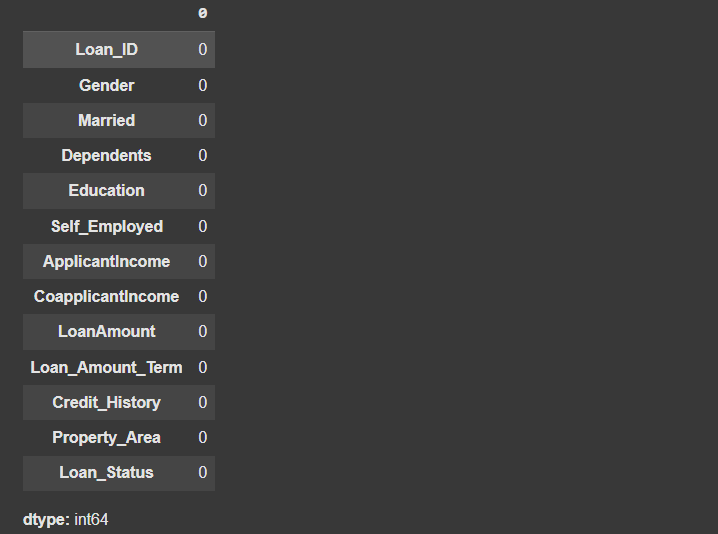
# Part B: Machine Learning

## 3.1 Data Preprocessing

* **Handling Missing Data**:

After loading the train dataset into the pandas, I checked on how many missing values are in the uploaded dataset. Then I divided the missing data columns into two groups, one is the categorical (e.g., Gender: Male, Female). And second one is the numerical (e.g., Loan\_Amount\_Term: 1,2,180). I removed all the missing values using the fillna package which can add the value which comes under mode method, that returns the mode of the series or list.

A computer screen shot of text

Description automatically generated

Result After Removing Nulls

Result Before removing nulls

Query for removing nulls

* **Encoding Categorical Variables**:

Having ordinal data the features, Gender, Education, and Property\_Area were first converted into numerical format for better comprehension by the machine learning algorithms. In the encoding stage, each category is assigned with a unique integer value that made these features amenable to numerical processing.

For instance:

The gender variable was coded as binary data sets; hence ‘Male’ was coded as ‘0’ while ‘Female’ was coded as ‘1’.

With regard educational variable, codes were assigned where Graduate = 1 and Not Graduate = 0.

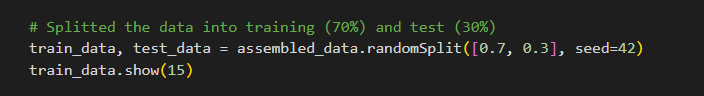
The four variables Property Area, which has three possible values or categories: Urban, Rural and Semiurban, each different integer value was assigned to each variable.

A computer screen with text on it

Description automatically generated

* **Data Splitting**:

The dataset was split into train and test dataset for model assessment to be completed. In particular, the authors devoted 70% of the data to chronologically preceding developing data and used 30% as the testing set. This split was done to enable the model to identify patters on most of the data with some of the data being set aside for testing. The Loan\_Status column was chosen for being the predictor of loan approval or rejection.



## 3.2 Model Training and Evaluation

* **Algorithm Selection**:

For this work, only two classifications algorithms namely, the Logistic Regression and Decision Trees were chosen for the purpose of predicting loan eligibility.

Logistic Regression was chosen for binary classification task, and also for interpretability of the results. Logistic Regression can build the probability of loan approval by input features because the dataset shows a binary response variable – approved or not approved loans.

Decision Trees were chosen given the fact that handle both numerical and categorical data, and do not over-fit their models, especially with a small set of data. This model can also be helpful in seeing the decision-making flow concerning loans approval thus helpful in estimating feature significance.

These algorithms were chosen because while all of them are relatively easy to interpret, they also have reasonable flexibility and are well suited for classification problems that include categorical variables.

* **Model Training**:

The models were both built using the training set within the Spark MLlib. The training process involved:

Feature Transformation: Specifically, to bring all parameters to numerical form to match the programs’ requirements and scale them in the way necessary to optimise algorithms.

Model Initialization and Fitting: Logistic Regression was started with default parameters to debug but Decision Trees were set criteria to control Maximum depth in order to prevent over training.

Model Fitting: Every built-up model was then trained for the data garnered from training data phase to help identify the patterns of the data models’ association with the loan approval (Target Variable: Loan\_Status).

* **Model Evaluation**:

As a result, to estimate the quality of models, the following indicators were used: Accuracy, Precision, Recall, the/confusion matrix for both the Logistic Regression and Decision Tree models on the testing set. All these metrics give overall information on how well each of the models is in classifying the status of the loans.

Accuracy: Logistic Regression was the highest in the amount of correct prediction being 81.06% and that Decision Trees was 76.9% correct.

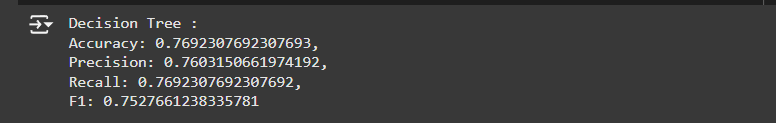
Precision: Accuracy of positive sentiments which included approved loans were 83.19% for Logistic Regression and 76.03% for Decision Trees indicating the ability of each model to correctly identify the approved loans.

Recall: Recall of Logistic Regression was 81.06%, while the Decision Trees was 76.92% which signified the competency of the models to find all the true approved loan in the test data.

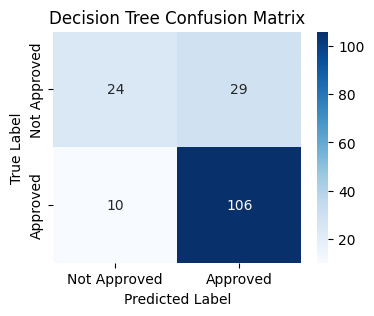
The confusion matrices of both models were represented in the form of heat maps. These visual communications directly showed the number of TP, FP, TN and FN for each model, which helped to understand the potential of each model for correct Loans approval assessment and rejection.

A black rectangular object with white text

Description automatically generated



A graph showing a logistic regression confusion matrix

Description automatically generated

## 3.3 Model Comparison

* **Performance Comparison**:

|  |  |  |
| --- | --- | --- |
| **Metric** | **Logistic Regression** | **Decision Tree** |
| Accuracy | 81.06% | 76.92% |
| Precision | 83.19% | 76.03% |
| Recall | 81.06% | 76.92% |

* **Confusion Matrix Visualisation**:

The confusion matrices as heat map are displayed showing the true positive, false positive, true negative, and false negative values for each of them. This visualization is used to distinguish between each model’s prediction distribution to reveal their strengths or weaknesses in estimating the loan statuses.

* **Discussion**:

Compared with Accuracy and precision and recall measures, it was found that the Logistic Regression model was even slightly better and therefore is more reliable for loan approvals. The most useful evaluation here was the precision since accurate prediction of approved loan samples is always essential. It can be assumed that Logistic Regression performed better than Decision Trees because the former does not have the problem of overfitting as a result of a small sample size.

# Conclusion

* **Summary of Findings**:

In Part A, the crime statistics were scrutinized to expose aspects of crime diffusion based on county and time series achieved through Spark data analytics and data visualization tools. Spark sql processing made it easier and faster to aggregate and compute the crime density and types as well as the frequency and hotspots over time.

In Part B machine learning models were built to predict loan eligibility. Apache Spark MLlib was used in the initial cleaning of the data where missing values of the evidence cases were handled, and categorical data was encoded for the model. The performances of the two built models: Logistic Regression and Decision Tree were compared and it was learned that, Logistic Regression was a tiny edge better in terms of predicting the loan statuses. This part used an example to show how Spark can construct scalable predictive models for possible actual financial applications. In total, this project enhanced the appreciation for Spark and its analytic and machine learning capabilities as well as exposed one of the main problems of data preparation and the potential of Spark’s MLlib for solutions.

* **Reflection**:

This project has widen the knowledge about such steps of data preprocessing, feature engineering, model evaluation. That a few methods had to be omitted, and the selection of the most suitable algorithms was difficult, can be seen as key challenges. It is possible to improve the performance in the future by attempting other models or changing some other setting known as hyperparameters.