**SALIFOU SYLLA DECEMBER 16TH, 2020**

**DATA 205: Capstone Experience in Data Science**

**Fall 2020**

**Instructor:** Michael Iapalucci

**FINAL REPORT**

**PROJECT: Predicting Car Crashes in Montgomery County:  
Can Machine Learning Help Us Understand What Factors Lead to Car Crashes?**

1. **Project Plan Overview**

**I.1 Dataset**

The dataset used is Crash Reporting Data set. Those datasets are about car crashes that have occurred in the Montgomery County, they provide information about the driver, the incident, and the non-motorist. They are from <https://www.montgomerycountymd.gov/>. The dataMontgomery program seeks to provide residents and constituents with direct access to County datasets in consumable formats, so they may be viewed, sorted and used in various ways, including being potentially leveraged in the development of new applications and services by interested parties. Providing this information offers the public an opportunity to review and analyze raw data, and the opportunity to use it for a variety of purposes. The following are those three datasets:

1. Crash Reporting - Incidents Data: <https://data.montgomerycountymd.gov/Public-Safety/Crash-Reporting-Incidents-Data/bhju-22kf>

This dataset provides general information about each collision and details of all traffic collisions occurring on county and local roadways within Montgomery County, as collected via the Automated Crash Reporting System (ACRS) of the Maryland State Police, and reported by the Montgomery County Police, Gaithersburg Police, Rockville Police, or the Maryland-National Capital Park Police.

64.9K rows and 44 columns

1. Crash Reporting - Drivers Data <https://data.montgomerycountymd.gov/Public-Safety/Crash-Reporting-Drivers-Data/mmzv-x632>

This dataset provides information on motor vehicle operators (drivers) involved in traffic collisions occurring on county and local roadways. The dataset reports details of all traffic collisions occurring on county and local roadways within Montgomery County, as collected via the Automated Crash Reporting System (ACRS) of the Maryland State Police, and reported by the Montgomery County Police, Gaithersburg Police, Rockville Police, or the Maryland-National Capital Park Police. This dataset shows each collision data recorded and the drivers involved.

115K rows and 43 columns

1. Crash Reporting - Non-Motorists Data <https://data.montgomerycountymd.gov/Public-Safety/Crash-Reporting-Non-Motorists-Data/n7fkdce5>

This dataset provides information on non-motorists (pedestrians and cyclists) involved in traffic collisions occurring on county and local roadways. The reports details of all traffic collisions occurring on county and local roadways within Montgomery County, as collected via the Automated Crash Reporting System (ACRS) of the Maryland State Police, and reported by the Montgomery County Police, Gaithersburg Police, Rockville Police, or the Maryland-National Capital Park Police. This dataset shows each collision data recorded and the non-motorists involved.

3723 rows and 32 columns

**I.2 Goal**

The goal of this project is to find the factors contributing to the traffic collisions in Montgomery County. Analyzing the County traffic collisions using visualization tools and Machine Learning Algorithms could provide the answers to the following questions:

* When are the most dangerous times to be driving?
* Which parts of county has the most traffic fatalities and/or crashes?
* Condition of roadway surface on collisions
* Non-Motorists involvements in car crashes
* Crashes by type of vehicles
* Number of crashes by year
* Predicting factors that contribute to crashes through Machine Learning Algorithms

**I.3 Approach**

Supervised learning describes a class of problem that involves using a model to learn a mapping between input examples and the target variable. There are two main types of supervised learning problems: they are classification that involves predicting a class label and regression that involves predicting a numerical value. I will approach the modeling part of this problem in different ways. But I decided to approach it as a **classification problem.** I will take it as **multi-class classification** problem and predict the outcome of a crash as Fatal, Injury or Property Damage (Three Classes)

To accomplish this project, I rely on the programming language python. Throughout my code, I used Python visualization tools and the Machine Learning Algorithms to achieve my goal.

Visualizations tools

* Anaconda: Open-source of python for scientific computing (Machine learn. Applications)
* Jupyter Notebook: Web application to create and share documents that contain live code (e.g., Python code) visualizations. explanatory text (written in markdown syntax)
* Pandas: Library for data manipulation
* Matplotlib: python package for visualization
* Seaborn:  library for making statistical graphics in Python

Machine Learning Algorithms

* Decision Tree used on incidents dataset
* Random Forest used on driver dataset
* Naïve Bayes used on Non-Motorist dataset

1. **Data Cleaning**

It is a process of preparing data for analysis by removing or modifying data that is incorrect, incomplete, irrelevant, duplicated, or improperly formatted. It is not simply about erasing information to make space for new data, but rather finding a way to maximize the dataset’s accuracy without necessarily deleting information. I first loaded all three datasets, then merged them into an unique dataframe.

**II.1 Loading Datasets**

Before loading the three datasets, I imported P**ython necessary classes, modules and libraries**

* NumPy: Python library used for working with arrays
* Pandas: module mainly works with the tabular data (DataFrame)

All three datasets have read from **the *data*Montgomery URL each time in the code.**

**incidentsData = pd.read\_csv(url)**

**1. Crash Reporting - Incidents Data (66,648 records, 44 attributes, weekly update) at**

[**https://data.montgomerycountymd.gov/Public-Safety/Crash-Reporting-Incidents-Data/bhju-22kf**](https://data.montgomerycountymd.gov/Public-Safety/Crash-Reporting-Incidents-Data/bhju-22kf)

**2. Crash Reporting - Drivers Data (118,300 records, 43 attributes, weekly update) at**

[**https://data.montgomerycountymd.gov/Public-Safety/Crash-Reporting-Drivers-Data/mmzv-x632**](https://data.montgomerycountymd.gov/Public-Safety/Crash-Reporting-Drivers-Data/mmzv-x632)

**3. Crash Reporting - Non-Motorists Data (3,849 records, 32 attributes, weekly update) at**

[**https://data.montgomerycountymd.gov/Public-Safety/Crash-Reporting-Non-Motorists-Data/n7fkdce5**](https://data.montgomerycountymd.gov/Public-Safety/Crash-Reporting-Non-Motorists-Data/n7fkdce5)

**II.2 Combining DataFrames with Pandas**

**II.2.1 Concatenating the three datasets**

Before combining the dataframes, I check for missing values in Pandas DataFrame using function isnull(). Each row has at least a missing value and several columns with greater number of missing values. I used ‘pandas.concat()’ function concatenates the three DataFrames and returns a new dataframe with the new columns as well. The dataframe row that has no value for the column will be filled with NaN.

**incidentsData (66 648 rows, 44 columns),**

**driversData(118 300 rows, 43 columns),**

**nonMotoristsData(3 849, 32 columns)**

**concat1 = pd.concat([incidentsData, driversData])**

(182286, 65)

**df = concat2 = pd.concat([concat1, nonMotoristsData])**

(185691, 72)

The final dataframe has 185 691 records and 72 attributes.

**II.3 Cleaning Each Dataset**

**II.3.1 Incidents Dataset**

**# Remove spaces in columns name**

incidentsData.columns = incidentsData.columns.str.replace(' ','\_').str.lower().str.replace('/','\_').str.replace('-','\_')

**# Drop unnecessary columns.**

Columns with high number of missing values are dropped

incidentsData.drop(['local\_case\_number', … 'report\_number' ], axis=1 ,inplace=True)

**# Remove rows with missing values**

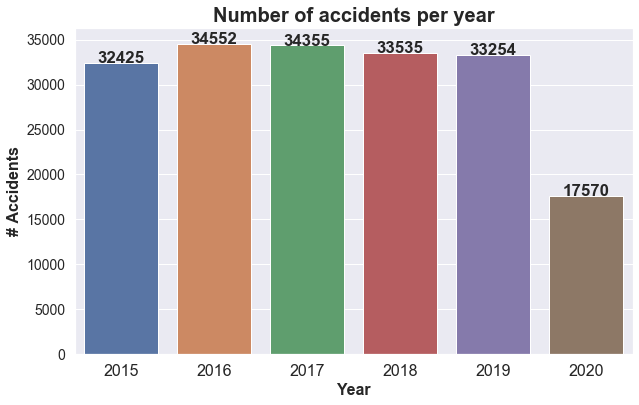
incidentsData.dropna(inplace=True)

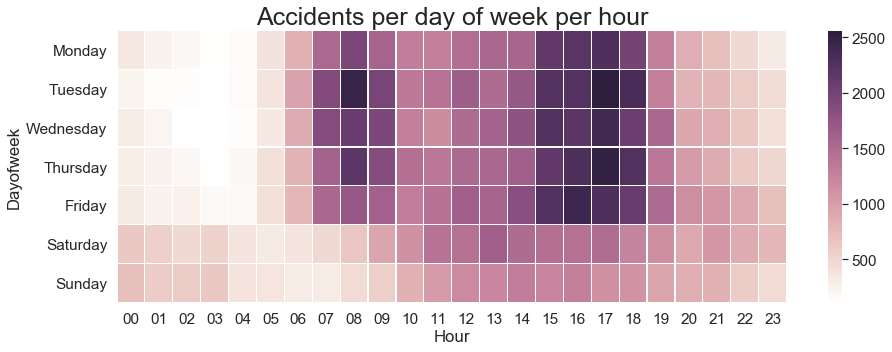
After cleaning up the incident Dataset, I end up with a much-reduced dataset of 21 603 records and 27 attributes. The same process was applied to the driver dataset (69 045, 24) and the non-motorist dataset (1 001, 22).

The column “crash\_date\_time”(Date and Time of crash.) is split in 5 columns: Day, Month, Year,

Hour and Dayofweek

1. **Descriptive Statistics**

****

****

Most of crashes occur during the workweek between 3:00 and 6:00 PM.

1. **Data Product Description**

**IV.1 Approach**

I used Supervised Learning techniques. I approached the problem as a **classification problem, precisely** as **multi-class classification** problem and predict the outcome of a crash as fatal, Injury or Property Damage. I used three different Machine Learning Algorithms for modeling, Decision Tree was used on incidents dataset, Random Forest Algorithm on drivers’ dataset, and Naïve Bayes Algorithm on non-motorists’ dataset.

**IV.2 Incidents Dataset 🡪 Decision Tree**

**IV.2.1 Feature selection using Mutual Information**

MI measures how much **information** the presence/absence of a term contributes to making the correct classification decision on.

Chart, bar chart

Description automatically generated  
**IV.2.2 Decision Tree**

A decision tree is a flowchart-like tree structure where an internal node represents feature(or attribute), the branch represents a decision rule, and each leaf node represents the outcome. The topmost node in a decision tree is known as the root node.

**IV.2.3 Classification Report**

The classification report displays the Precision, Recall , F1 and Support scores for the model.

**Table

Description automatically generated**

**Precision** score means the level up-to which the prediction made by the model is precise. **Recall** is the amount up-to which the model can predict the outcome. **F1** and **Support** scores are the amount of data tested for the predictions.

**IV.2.4 Confusion Matrix**

Confusion matrix is a table which describes the performance of a prediction model. A confusion matrix contains the actual values and predicted values.

**Text

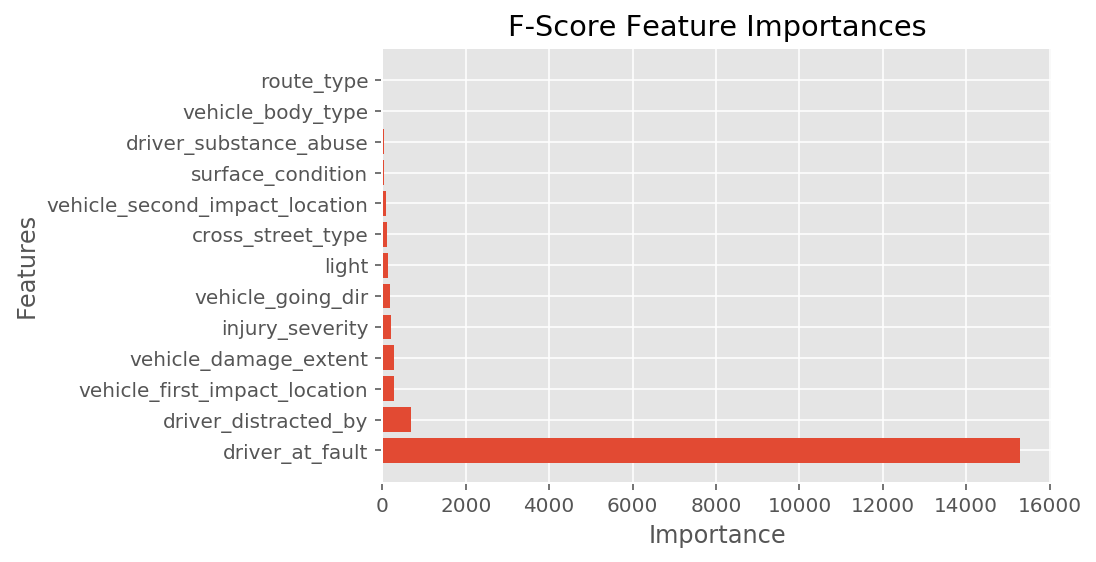
Description automatically generated**

**IV.2.5 Accuracy Score**

Accuracy score is the percentage of accuracy of the predictions made by the model.

**IV.3 Drivers Dataset 🡪 Random Forest**

**IV.3.1 Feature selection**

****

**IV.3.2 Random Forest Algorithm**

**It builds multiple decision trees and merges them together to get a more accurate and stable prediction.**

**IV.3.3 Classification Report**

**Table

Description automatically generated**

**IV.3.4 Confusion Matrix**

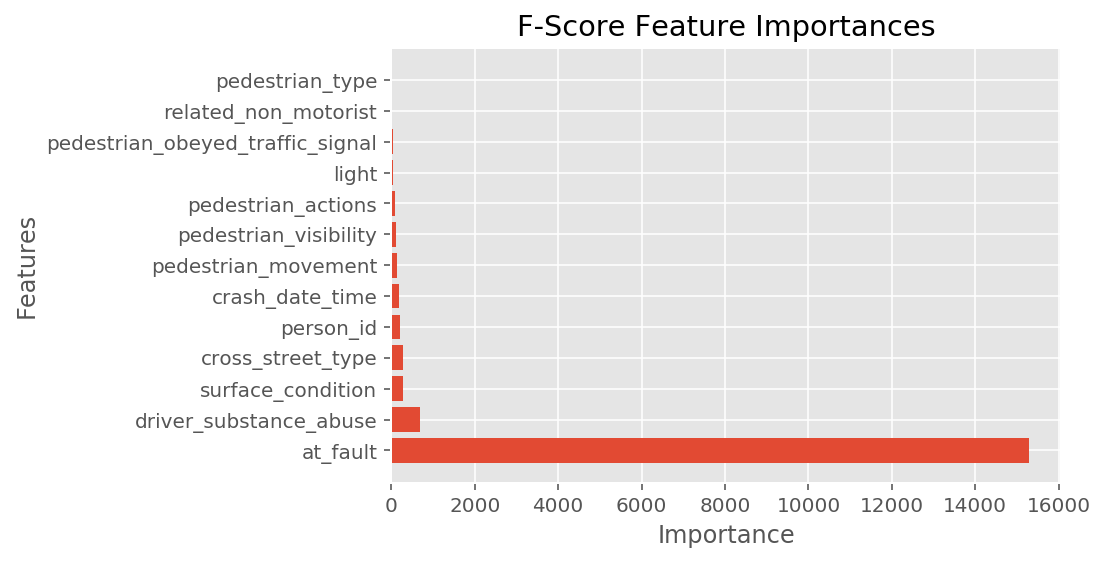
**Text, letter

Description automatically generated**

**IV.3.5 Accuracy Score**

**IV.4 Drivers Dataset 🡪 Naïve Bayes**

**IV.4.1 Feature selection**

****

**IV.4.2 Naïve Bayes Algorithm**

It is a classification technique based on Bayes’ Theorem with an assumption of independence among predictors. In simple terms, a Naive Bayes classifier assumes that the presence of a particular feature in a class is unrelated to the presence of any other feature.

**IV.4.3 Classification Report**

**Table

Description automatically generated**

**IV.4.4 Confusion Matrix**

**A picture containing text, clock

Description automatically generated**

**IV.4.5 Accuracy Score**

**0.96**

**V. Data Stories**

Data stories explore and explain how and why data changes over time, usually through a series of linked visualizations. The dataset used is Crash Reporting Data set. It includes three datasets. Those datasets are about car crashes that have occurred in the Montgomery County, they provide information about the driver, the incident, and the non-motorist. They are from <https://www.montgomerycountymd.gov/>. The dataMontgomery program seeks to provide residents and constituents with direct access to County datasets in consumable formats, so they may be viewed, sorted and used in various ways, including being potentially leveraged in the development of new applications and services by interested parties. Providing this information offers the public an opportunity to review and analyze raw data, and the opportunity to use it for a variety of purposes. The data within these datasets are structured, handled in a professional and secure manner, and can be trusted because it is owned by the county. They are weekly updated. As today the incidents dataset has **66 494 records and 44 attributes**. Each row has at least one missing value (NaN). A list of all the summations of each column (incidentsData.isnull().sum()) shows 12 attributes (out of 44) without missing values. The drivers’ dataset has **118036 records and 43 attributes**. Only 17 out of 43 attributes have no missing values. The Non Motorists dataset has few records with **3839 records and 32 attributes**. 12 out of 32 have no missing values. Removing the rows for each null value will greatly size down the datasets resulting in a skewed model. I decided to keep all three datasets and concatenate them for their visualizations. By bricking down the crash\_date\_time into variables Day, Month, Year, Hour and Dayofweek , I was able to show through visualizations, the most dangerous times to be driving. Most accidents tend to occur between 3 pm and 6 pm. With other visualizations, the dataset shows the following discoveries:

* Number of accidents per year
* Number of fatalities per year
* How roadway surface (type) impact crash
* Non motorists (type) involvement in crash
* Type of vehicle involved in crash  
  There is lack of precision in the number of fatalities, the variable 'acrs\_report\_type' has value fatal crash without reporting how many. By lack of information about the driver age and gender, I was unable to investigate the following
* The age group that has the highest number of crashes (or fatalities)
* Gender ratio in the crashes

The difficulties of joining the three datasets without losing substantial information and shirking the data size, I decided that the modeling being done in each dataset. The Multi-class Classification by Decision Tree on the incidents’ dataset and Multi class Classification using Random Forest on Non-Motorists dataset.

1. **DataMontgomery Experience**

The data is *good* if it accomplishes its intended task.

* Consistent Data

The data within these datasets are structured.

* Data Security:

The data is handled in a professional and secure manner.

* Well defined data
* Reliability refers to the degree to which you can *trust* your data.

This data can be trusted because it is owned by the county.

However, the datasets are lack of information about the driver such as age, sex. I intended to provide some charts about the rate (percentage) of fatal accident by group age (16-19y; 20-24y; ….; 65-74; 75+). There is not specific about fatalities, the number of fatalities in each crash.