8/25/2020

Road Accidents in Canada Explored (Project RACE)

Tolu Fatoki & Syed Tasrif Ahmed

Team Member And Tasks

	Team Member	Task
1.	Tolu Fatoki	• Data Analysis and Exploration with python • Build machine learning model • Testing • Project Documentation
2.	Syed Tasrif Ahmed	• Web development • Build interactive visuals with chart.js • Build Machine learning model • Testing • Documentation

ABSTRACT

The importance of road transportation cannot be over-emphasized on our daily lives, however there is the risk of road accidents attached with this method of commuting that has become an everyday part of our lives. In this project we will be exploring the road accidents in Canada and with the access to reported road accident data provided by the government of Canada we will be determining road configurations that contribute the most to accidents on Canadian roads. We will perform different analysis on the dataset to have a clear understanding of reported road accidents in Canada, seeing how different features contribute to these accidents. We developed a machine learning model that will predict the collision hotspot (which is simply the type of road configuration with the greatest where most accidents have occurred and will probably occur in the future). Our model was developed using different tools but to make it access to different users we developed a web application where users can interact with the model.

Keywords: Machine Learning, Random Forest, Web Application, Road Accident

INTRODUCTION

Transportation is the process of moving from one location to another for various reasons, from business, personal to leisure. Road transportation is one of the most popular and reliable means of transportation. Road transport has become an important part of our everyday life, we commute via road to go to work, school, and leisure. Roads also play an important role in the economic development of any country and serve as a backbone for any strong economy and as such, it is very important to keep transporting on these roads safe and secure for everyday users. No wonder the Canadian Council of Motor Transport Administration (CCMTA) in its 2016 publication developed the Road Safety Strategy(RSS) 2025 with the aim to make Canadian roads one of the safest in the world (CCTMA, 2016). One of the strategies they are going to employ to achieve this is "improving the safety of vehicles and road infrastructure". Having better and improved road infrastructure is key to ensuring road safety and reducing road accidents and in this work, we will be exploring how road configurations have contributed to the number of road accidents and how collision hotspots can be predicted based on available and help the government make right decisions in improving infrastructure in our roads.
In the spring of 2018, a now very popular road accident occurred near Armley, Saskatchewan claiming 16 lives and 13 fatal injuries (The Canadian Encyclopedia, 2018). The accident occurred at the intersection of Saskatchewan Highways 35 and 335 (Figure 1) between a semi-trailer truck and a coach bus carrying players of a junior hockey team from Humboldt, Saskatchewan. The main cause for this accident has been generally agreed to be the semi-truck driver's failure to yield at a flashing stop sign at the intersection. According to this report by (CBC News, 2018) a similar accident occurred on this same intersection about 2 decades before this particular Humboldt broncos incident which prompted the installation of the stop sign to prevent future occurrence of any form accidents at the intersection. It is reported that accidents at this intersection are even lower when compared to other intersections in the Saskatchewan and even in Canada as a whole as this intersection does not even feature on the list of the top 20 deadliest intersections in Canada according to this report (Markham Mitsubishi, n.d.).

The intersection on any road is just one of the many types of road configuration where accidents occur and just one of the many other reasons why accidents happen around the world. However, our work will focus on accidents that happen as a result of various road configurations as we view this cause of accidents as a major contributing factor to road accidents. Our dataset shows that road configuration contributes about 30% of the total road accidents that occur in Canada and we believe that tackling this particular issue will contribute immensely to reducing road accidents and help the government in reaching its vision of tending towards zero road accidents.

PROBLEM STATEMENT

The challenge of road accidents has been an ever-present one all over the world and especially in Canada. According to WHO's 2018 report about 1.35 million people die each year from road accidents all over the world and this number is always increasing as many governments are not taking the required steps to reduce and curb the present challenges of road accidents. The Canadian government is committed to ensuring the safety of lives and properties in its road and has made different efforts to ensure that the number of accidents on our roads is reduced significantly. This commitment can be seen in the gradual reduction of accidents and injuries sustained over the years on our roads. We hope that with this research, we can contribute to the continued efforts of the government to reduce the number of deaths on our roads.
Several factors are always responsible for different road accidents and different categories of these factors can be identified from issues like the weather condition, road configuration, age of the driver, mental state of the driver, road surface, traffic control, time of the day, the period of the year, type and age of the vehicle, usage of recommended safety equipment. As all these factors need to be researched to get a holistic view of reducing the problem of road accidents, our main focus for this project will be on the accidents caused by road configuration which we observed has received little or no attention as no mention of any analysis done on it was available on the government of Canada website and no resources were found on the internet as regarding this crucial composition of our daily commuting. Road configuration as the name implies refers to the configuration of the road, that is how the road is built for commuting. Examples of road configuration which we will explore based on our dataset include intersection road (like the Humboldt case-study), railroad level crossing, bridges, overpass, viaduct, tunnel, underpass, ramp e.t.c.
How a road is constructed might seem like a negligible detail when it comes to road accidents but like in the case study presented above our project will highlight how road configuration has had a huge effect on the number of road accidents and how having a good understanding of collision hotspots around the country and what measures can be put in place to reduce collision in those types of roads.

RELATED WORK

The authors (Kaya, Ayas, Ponnambalam and Donmez, 2018) focused on how drivers are distracted at various intersection and most importantly how vulnerable road users like pedestrian and cyclists are mostly ignored at these intersections which largely contribute to road accidents. They performed a survey of drivers in different age groups creating a scenario for the drivers about 70% of the driver failed at least in one of the test. They discussed that drivers with less familiarity with certain roads and intersection were more cautious.

(Espinosa, 2015) suggested that a way to reduce the different factors that affect safety at signalized intersection and ultimately reduce crashes by the introduction of Automated Vehicle(AV). Automated Vehicle have the ability to communicate with other vehicles and road infrastructure. They performed a simulation that showed driver conflict is a huge potential cause of road crashes and how AV at signalized intersection can help reduce this conflict.

The paper written by (Roos, 2016) aims to explore the factors that contribute to fatal rural vehicle collisions and identify how the factors differ between rural and urban areas.

(Arason, 2019) explored dataset from the province of British Columbia from 2004 to 2015 to determine the count and proportion of fatal injuries that occur on the road.

These papers reviewed the different factors that contribute to road accidents with a focus on intersection, vulnerable users and vehicles and they were all limited to some city within Canada. They performed some simulations to support their different findings, however, our work provides a different perspective on a broader and larger domain where we explore data from road accidents all around Canada for a period of about 20 years. We developed a model that can used to predict or determine future occurrence of accidents in important intersection (collision hotspot) all over the country which we are certain can help decision makers to implement necessary safety measures to help protect our roads.

.

SOLUTION OVERVIEW

This data science project Road Accidents in Canada Explored proposed a model where road accidents in Canada can be predicted based on a unique factor that has not been well covered called Road Configuration. The goal of this project is to ultimately develop a web application (as shown in figure 1) where the results of our analysis and machine learning algorithm can be accessed by users. We employed a combination of different tools to achieve this goal and they include python, CSS, HTML, chart.js, MongoDB. Our project was developed as a business solution for different stakeholders who make rules and regulation concerning road accidents and it ultimately assist them to put in place good safety measures taking into consideration the predictions of collision hotspots from our model, also different users who may be interested in having information about road accidents in Canada can find this work very useful.

Figure 1: Solution Overview

We are certain that this model will be useful for Transport Canada and various policy makers and stakeholders in the transportation industry to develop interventions aimed at reducing road accidents in Canada.

METHODOLOGY

In order to successfully develop this project, we followed the typical data analytical life cycle which includes the following stages (Figure 2); Discovery, Data Preparation, Model Planning, Model Building, Communicate Result and Operationalize.

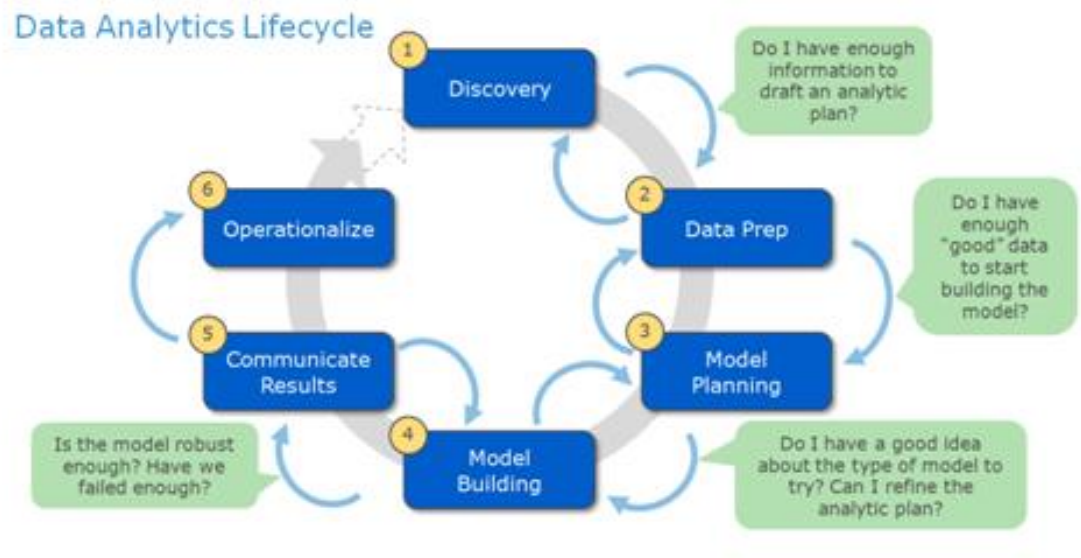

Figure 2: Data Analytics Lifecycle

In the discovery stage, we discussed how can formulate the problem of road accidents in Canada as analytic problem, we identified the possible stakeholders who may benefit from this project, we researched the different tools that we may use to complete the project and ultimately determined how we could get a comprehensive dataset that could capture enough information that we can use to achieve our goal.

The next stage which is the data preparation stage gave a good feel of the dataset we had and how we can answer different questions based on the dataset. The stage involved the preliminary steps taken to clean and transform the data into a form that will be usable for our purpose.

The model planning stage involved discussing which machine learning model will be suitable for our purpose and how we can develop a web application where details of our analysis can be explored, we determined which variables can be combined with road configuration to provide better results. We extensively discussed the tools that may be required to complete the project.

With our model well spelt out in the previous stage, we were ready to execute the project at the model building stage. We proceeded by training and testing the dataset, predicting the results

and observing the performance of the model on the test data. We were able to compare different algorithm and choose an effective and robust one for our purpose.

The communication phase involved developing well documented reports and visualizations for different stake holders which included a user guide, a formal presentation of our process and the results of our work, code/setup guide.

The operationalize stage involves deployment of the solution and we clearly explain how the system works and adequate support is provided for the users of the system.

We will explore this lifecycle and how we achieved each stage in our project with some of the shortcomings encountered.

THE DISCOVERY STAGE

At this stage, we discussed what problems we could want to solve and how the problem could be formulated as data analytics challenge. The problem we are considering has to be a real life business problem that can positively affect and improve a business operation. We considered different domains like, Internet of Things, security, retail, transportation, Agriculture, sports and different datasets that we can explore. After careful consideration we decided to tackle the **problem road accidents in Canada**. We did some research on different road accidents in Canada and we quickly discovered that there were many causes of road accident around the world and here in Canada but very few work has been done to determine how road configuration has place a major part in these accidents. We were further motivated by the 2018 Humboldt car crash that happened in Saskatchewan at one of the most popular identified road configuration (road intersection). After we achieved the successful selection of the problem, we tried to formulate it as an analytic problem. The process involved discovering how the problem of road accidents can be reduced with the help of technology, with the help of analyzing data and predicting potential causes of accidents. We identified how we can gain access to datasets or develop datasets that can help us with the problem and what tools can be helpful.

The discovery stage typically involves conducting interview with different stakeholders, however because of the restriction to movement due to the ongoing pandemic we could not physically contact anyone to discover business need as regards road accidents however we are convinced that results from our project will help stakeholders to improve the business process.

Business Problem: How can road accidents in Canada be reduced considering the effect of road configuration.

DATA PREPARATION

In this stage of our project, we set out gather the dataset that we will use to execute this project. The process of getting our data simply involved searching the internet for the best and reliable data that contains adequate information that can be used by our model to predict road accidents and ultimately reduce their occurrence. After considering different datasets, we were directed to the government of Canada website where they had details of reported road accidents in Canada from the year 1999 to 2017. The dataset contained 23 columns and over 6 million rows and it was store as a Comma Separated Value(CSV) (Figure 3).

```
In [3]: dataset
```

Out[3]:

	C_YEAR	C_MNTH	C_WDAY	C_HOUR	C_SEV	C_VEHS	C_CONF	C_RCFG	C_WTHR	C_RSUR	...	V_TYPE	V_YEAR	P_ID	P_SEX	P_AGE	P_PSN
0	1999	1	1	20	2	02	34	UU	1	5	...	06	1990	01	M	41	11
1	1999	1	1	20	2	02	34	UU	1	5	...	01	1987	01	M	19	11
2	1999	1	1	20	2	02	34	UU	1	5	...	01	1987	02	F	20	13
3	1999	1	1	08	2	01	01	UU	5	3	...	01	1986	01	M	46	11
4	1999	1	1	08	2	01	01	UU	5	3	...	NN	NNNN	01	M	05	99
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
6772558	2017	UU	U	UU	2	UU	UU	01	U	U	...	01	UUUU	01	F	20	11
6772559	2017	UU	U	UU	2	UU	UU	01	U	U	...	01	UUUU	01	F	47	11
6772560	2017	UU	U	UU	2	UU	UU	01	U	U	...	07	UUUU	01	M	24	11
6772561	2017	UU	U	23	2	01	03	01	1	1	...	16	UUUU	01	M	45	96
6772562	2017	UU	U	23	2	01	03	01	1	1	...	16	UUUU	02	F	45	96

6772563 rows × 23 columns

Figure 3: Overview of the dataset

The following processes were conducted in the data preparation stage.

DATA COLLECTION

We accessed the government of Canada website through the National Collision Database (NCDB) where we have updated data of all the reported road accidents that occurred in Canada between the years 1999 to 2017. The dataset is detailed for the period recorded and it is constantly updated once new data becomes available. The size of the CSV formatted file was about 700MB.

A detailed description of the columns can be seen in the data dictionary provided with the dataset. The data appears to be in a structured format but a closer look shows that we have different columns with some irregularities, this implies that some analysis and transformation are required to get the data in our desired form. We have 20 categorical columns that can be used in this analysis to categorize the dataset and 3 numerical columns which will provide the required figures for the different categories in this analysis.

DATA STORAGE

The data was originally formatted as a CSV file, which made accessing it very simple. We downloaded the file to a local machine and also created copies on google drive. The copies created google drive served as our analytic sandbox where we used google colab to play around the dataset and have a good understanding of the metadata. From these storage locations we could access the dataset for preprocessing, building our model and developing the web application.

DATA EXPLORATION, TRANSFORMATION AND CONDITIONING

Here we highlight the different processes we explored to transform and clean the data for our model. We checked for null data in our dataset and discovered we had no null values as shown in figure 4.

```
In [4]: #check null values at each feature column
        dataset.isna().sum()

Out[4]: C_YEAR    0
        C_MNTH    0
        C_WDAY    0
        C_HOUR    0
        C_SEV     0
        C_VEHS    0
        C_CONF    0
        C_RCFG    0
        C_WTHR    0
        C_RSUR    0
        C_RALN    0
        C_TRAF    0
        V_ID      0
        V_TYPE    0
        V_YEAR    0
        P_ID      0
        P_SEX     0
        P_AGE     0
        P_PSN     0
        P_ISEV    0
        P_SAFE    0
        P_USER    0
        C_CASE    0
        dtype: int64
```

Figure 4: Checking for null values

We also checked for distinct values in the dataset to determine unnecessary data, figure 5 shows this;

```
In [5]: # Find all distinct value of each column
        for col in dataset.columns:
            print(col, set(dataset[col]))
            print('\n')
```

```
C_YEAR {1999, 2000, 2001, 2002, 2003, 2004, 2005, 2006, 2007, 2008, 2009, 2010, 2011, 2012, 2013, 2014, 2015, 2016, 2
017}


C_MNTH {1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, '12', 'UU', '02', '01', '11'}


C_WDAY {1, 2, 3, 4, 5, 6, 7, '7', '2', '4', '3', '6', '5', '1', 'U'}


C_HOUR {'18', '13', '14', '19', '22', '16', '21', '00', '01', '17', '11', '15', '08', '20', '12', '03', '07', '06',
'02', '05', '23', '10', 'UU', '09', '04'}


C_SEV {1, 2}


C_VEHS {1, 2, 3, 4, 5, '13', 6, 8, 7, 10, 9, 12, 13, 11, 15, 14, 17, 18, 19, 20, 21, '19', 16, '21', 25, 26, 27, 28,
22, 24, 31, 33, 35, 36, 37, 38, 39, '54', 43, 44, 46, 47, '01', '17', 56, '41', 58, 57, '15', '71', '29', 72, 77, '0
```

Figure 5: Distinct values

We identified that some of the columns are not categorical and some of them values are stored as strings. In order to balance this data and prepare it for our use, all the values were converted to integers and then we made all the columns categorical. We can see the updated dataset in figure 8 after these transformations.

```
In [7]: # Make all coumns int

        def make_int(df, *arg):
            for col in arg:
                df[col] = df[col].astype(int)
            return df

        df = make_int(df, 'C_YEAR','C_MNTH','C_WDAY','C_HOUR','C_SEV','C_VEHS','C_CONF','C_RCFG','C_WTHR','C_RSUR',
                      'C_RALN','C_TRAF','V_ID','V_TYPE','V_YEAR','P_ID','P_AGE','P_PSN','P_ISEV','P_SAFE','P_USER','C_CASE')
```

Figure 6: Convert to Integers

```
#make all column categorical data
def convert_to_cat(df, *arg):
    for column in arg:
        df[column] = df[column].astype('category')
        df[column] = df[column].cat.codes

    return df

df = convert_to_cat(df, 'C_YEAR','C_MNTH','C_WDAY','C_HOUR','C_SEV','C_VEHS','C_CONF','C_RCFG','C_WTHR','C_RSUR',
             'C_RALN','C_TRAF','V_ID','V_TYPE','V_YEAR','P_ID','P_AGE','P_PSN','P_ISEV','P_SAFE','P_USER','C_CASE', 'P
```

Figure 7: Convert to categorical

df

	C_YEAR	C_MNTH	C_WDAY	C_HOUR	C_SEV	C_VEHS	C_CONF	C_RCFG	C_WTHR	C_RSUR	...	V_TYPE	V_YEAR	P_ID	P_SEX	P_AGE	P_PSN
52	0	0	0	9	1	1	14	1	0	0	...	0	86	0	0	32	0
54	0	0	0	9	1	1	14	1	0	0	...	0	86	0	0	69	0
125	0	0	0	20	1	0	2	2	0	0	...	0	82	0	0	37	0
141	0	0	0	5	1	1	0	1	2	1	...	2	89	0	1	33	0
142	0	0	0	5	1	1	0	1	2	1	...	2	89	1	1	29	2
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
6772521	18	11	6	18	1	1	15	1	2	1	...	0	101	0	0	23	0
6772522	18	11	6	23	1	0	3	0	1	4	...	0	90	0	0	41	0
6772529	18	11	6	19	1	1	13	1	2	1	...	0	100	0	0	22	0
6772534	18	11	6	13	1	1	6	1	1	0	...	0	101	0	0	18	0
6772540	18	11	6	22	1	0	3	0	3	4	...	1	104	3	1	37	5

Figure 8: Updated dataset after transformation

DATA VISUALIZATION

This involved the initial exploration of the dataset that we have to adequately understand it. We were able to see the different characteristics of the data at a glance. For example, figure 9 shows the total reported cases per year and the age distribution of the data. Because we are visual beings we performed this step to provide clarity on the various features in the data.

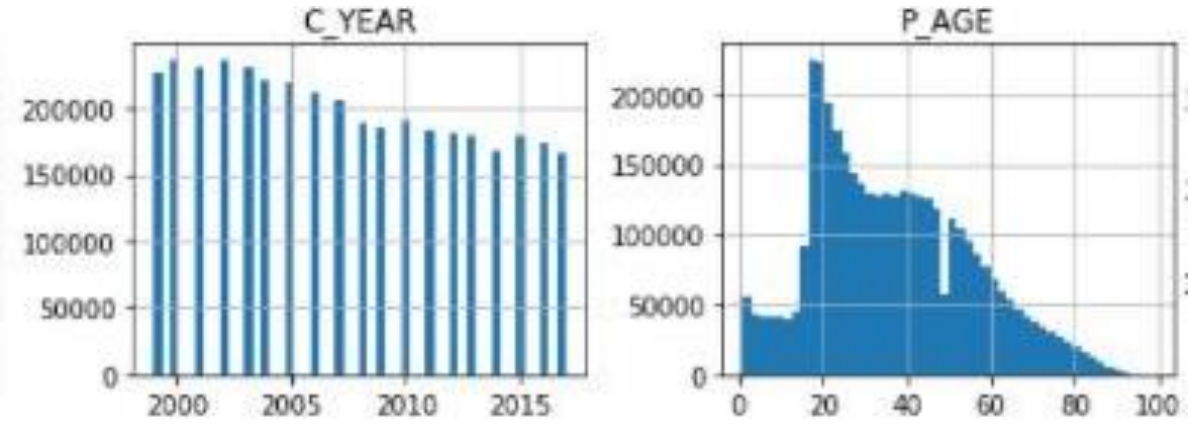

Figure 9: Preliminary Visuals

MODEL PLANNING

In this stage, after we have successfully determined that the road configuration was an important feature in the dataset that we want to concentrate on and predict, we went ahead to decide if other features can be added to the model. We also detailed the machine learning model and tools used to develop this project.

LEARNING TECHNIQUES

Typically, we have three types of machine learning which include supervised learning, unsupervised learning and Reinforced Learning;

- Supervised learning: This is most basic type of machine learning. The machine learning algorithm in supervise learning is trained on labeled data. The algorithm is given a sample dataset which gives a broad overview of the actual data from the original dataset and provides the labeled parameters required to solve the analytic problem.
- Unsupervised Learning: This works on unlabeled datasets and it implies that much larger and unstructured dataset can be explored with these algorithms. Unlike the supervised learning where we have relationships between input and output values, here the algorithm perceives the relationship in an abstract manner without any input from human beings.
- Reinforced Learning: The algorithm learns new situation using trial and error method, the favorable outputs are 'reinforced' and non-favorable outputs are 'punished'

For our purpose we adopted the supervised learning approach to solve our problem. The problem has I have mentioned earlier is to predict the collision hotspot on Canadian using the road configuration feature from our dataset. We considered KNN algorithm, Decision Tree and Random Forest algorithm and the accuracy results as shown in the table below shows that we get similar results from these three algorithms and either of them can be employed for our model. However, we decided to use to the Random Forest algorithm.

MACHINE LEARNING ALGORITHM	TRAINING ACCURACY	TEST ACCURACY
KNN	0.9989065177868144	0.9840061862864363
Decision Tree	0.9998788949060914	0.9771441433526953
Random Forest	0.9988805844214179	0.9838515787578622

Table 1: Confusion matrix comparison

Random Forest Algorithm

Random forest can be described as a classification algorithm that contains many decision trees where each tree is fully developed and the more the trees the more accurate the algorithm is. The concept behind this algorithm is that it uses bagging and feature randomness to build each individual tree which can then be used to create unrelated trees that gives a better accuracy result than any of the individual trees. So instead of using the average of the trees two techniques are employed that provides the randomness (An Implementation and Explanation of the Random Forest in Python, 2020)

- Random sampling of training data points when building trees: This involves the process by which each tree in the forest learns from a random sample of data when training. Some of the samples will be used multiple time and this process is called bootstrapping. By training each tree, the overall bias of the model will be greatly reduced and the entire forest will have a lower variance even in a situation where we have individual trees with high variance. During testing the predictions are made by finding the average of predictions made by each tree
- Random subsets of features considered when splitting nodes: The subset of all the features that are considered for splitting

One major advantage of the random forest algorithm is that it handles the problem of overfitting very well. The below illustrates a simple random forest algorithm

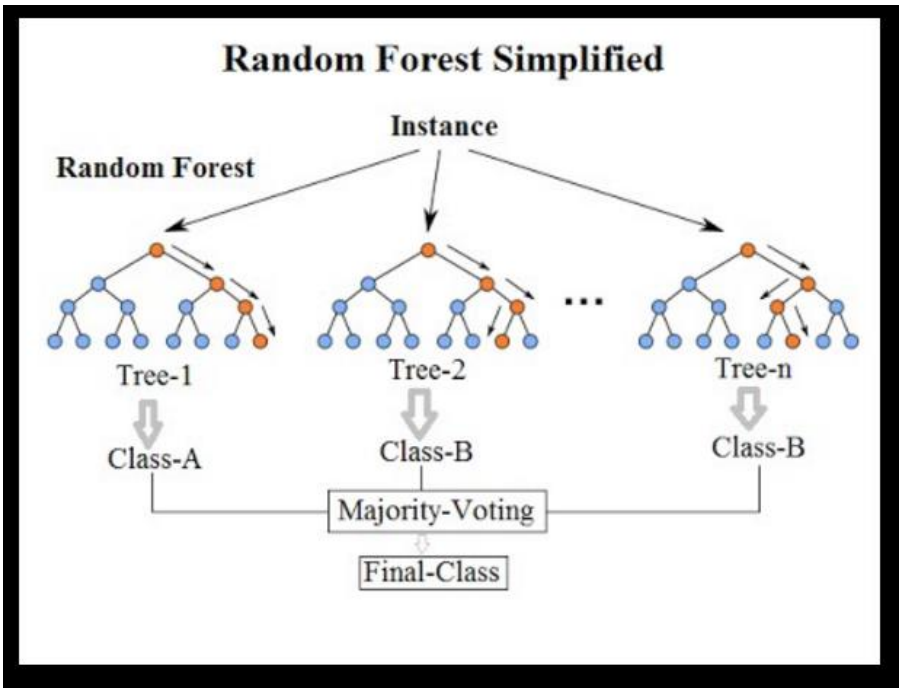

Figure 10: Random Forest

MODEL BUILDING

In this stage, we will implement all the decisions we made at the model planning phase, we will train the selected model using the labeled dataset as the input, we will also evaluate the model and make any necessary adjustment. In most cases the data labeled data is split into training and testing data where we have 80% of the data for training and 20% for testing, the test part of the data is not provided to the machine as this is what we require the algorithm to predict so that we can compare the accuracy with the actual dataset.

FEATURE SELECTION AND FEATURE ENGINEERING

We start off by selecting the target column which in our case is the road configuration and storing it as the y variable (figure 11);

```
In [11]: #seperate target col from other features to see feature importance
         X = df.drop(['C_RCFG', 'C_CASE'], axis=1)  #independent columns
         y = df['C_RCFG']    #target column i.e price range
```

Figure 11: Separating the target column

The dataset contains 23 columns as I have mentioned earlier but as we do not need all 23 columns we identify the important columns and remove unnecessary ones through the process referred to as **feature engineering.** After we figured out the important features we used scikit learn to extract the top 10 features that we could need and removed the unnecessary ones. Figure 12 shows this process;

```
#feature importance


model = ExtraTreesClassifier()
model.fit(X,y)
print(model.feature_importances_) #use inbuilt class feature_importances of tree based classifiers

#plot graph of feature importances for better visualization
feat_importances = pd.Series(model.feature_importances_, index=X.columns)
feat_importances.nlargest(15).plot(kind='barh')
plt.show()

plt.savefig("out.png")
```

```
C:\Users\muhelal\AppData\Local\Continuum\anaconda3\lib\site-packages\sklearn\ensemble\forest.py:245: FutureWarning: The default value of
  "10 in version 0.20 to 100 in 0.22.", FutureWarning)
```

```
[0.05424463 0.05778085 0.04724343 0.06583709 0.00218617 0.02225569
 0.12264569 0.01619719 0.01578356 0.01772732 0.36314093 0.01416949
 0.00985532 0.07100111 0.00977882 0.0061516  0.07564453 0.01046039
 0.00784163 0.00605418 0.00400035]
```

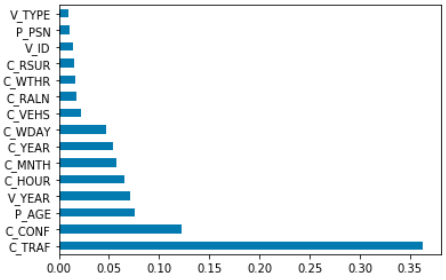

Figure 12: Feature Engineering

```
from sklearn.feature_selection import SelectKBest
from sklearn.feature_selection import chi2


#apply SelectKBest class to extract top 10 best features
bestfeatures = SelectKBest(score_func=chi2, k=20)
fit = bestfeatures.fit(X,y)
dfscores = pd.DataFrame(fit.scores_)
dfcolumns = pd.DataFrame(X.columns)

#concat two dataframes for better visualization
featureScores = pd.concat([dfcolumns,dfscores],axis=1)
featureScores.columns = ['Specs','Score']  #naming the dataframe columns
print(featureScores.nlargest(20,'Score'))  #print 10 best features
```

Figure 13: Select best columns

```
#drop cols that are not important after feature engineering
def drop_cols(df,*arg):
    columns = list(arg)
    df = df.drop(columns, axis=1)
    return df

df = drop_cols(df,'V_ID','P_PSN','C_CASE','P_SAFE','C_RSUR','C_RALN','P_USER','P_SEX','V_TYPE','C_SEV','P_ISEV','P_ID')
```

Figure 14: Drop unnecessary columns

As shown in figure 15, the road configuration variables appear to be unbalanced and to avoid a biased prediction from our model, we continued the process of building this model by balancing the road configuration column by up sampling the data.

```
In [12]: #looks like the class values are heavily imbalanced
         df['C_RCFG'].value_counts()

Out[12]: 1    2014779
         0    1523700
         2     218917
         4      35001
         3      11838
         7       7185
         5       4279
         8       1299
         6        364
         9        251
         Name: C_RCFG, dtype: int64
```

Figure 15: Unbalanced data

```
In [13]: #upsample data to balance
         def balance_dataset(dataset, column, frequency_val):
             count = dataset[column].value_counts()
             classes = [key for key,val in count.items()]
             df_majority = dataset.loc[dataset[column] == classes[0]]

             all_minor_dfs = []
             for val in classes[1:]:
                 all_minor_dfs.append(dataset.loc[dataset[column] == val])

             # Upsample minority class
             all_minor_dfs_upsampled = []
             for val in all_minor_dfs:
                 df_minority_upsampled = resample(val,
                                                  replace=True,     # sample with replacement
                                                  n_samples=frequency_val,    # to match majority class
                                                  random_state=123) # reproducible results
                 all_minor_dfs_upsampled.append(df_minority_upsampled)

             #concatenate majority class with matching upsampled classes
             #for df in all_minor_dfs_upsampled:
             df_balanced = pd.concat([df_majority, all_minor_dfs_upsampled[0]])
             for df in all_minor_dfs_upsampled[1:]:
                 df_balanced = pd.concat([df_balanced, df])

             df_balanced = shuffle(df_balanced)
             return df_balanced

         df_balanced = balance_dataset(df, 'C_RCFG',2014779)

In [14]: df_balanced['C_RCFG'].value_counts()

Out[14]: 9    2014779
         8    2014779
         7    2014779
         6    2014779
         5    2014779
         4    2014779
         3    2014779
         2    2014779
         1    2014779
         0    2014779
         Name: C_RCFG, dtype: int64
```

Figure 16: Up sampling to balance the data

MODEL TRAINING

Now we proceed to crucial part of our implementation which involves training the data with the model which we have developed. The first step we executed was to split the dataset into the train and test data. This process involved taking a sample of the dataset and separating or dividing them into train and test data. We provided 80% of the dataset for training while the remaining 20% is used for testing. Using random forest, we then proceeded to train the data and at the end of the training session we used the test data to evaluate the accuracy of the results of the model and the plot below the comparison between what our model predicted and the actual data, we also used k-fold validation to analyze the data.

```
#split into train and test
def split_train_test(dataframe, target):
    train_set, test_set = train_test_split(dataframe, test_size=0.2, random_state=42)
    X_train, X_test = train_set.drop(target, axis=1), test_set.drop(target, axis=1)
    y_train, y_test = train_set[target], test_set[target]

    return X_train, X_test, y_train, y_test

X_train, X_test, y_train, y_test = split_train_test(df_balanced, 'C_RCFG')
```

Figure 17: Splitting the dataset

```
def fit_predict(X_train, X_test, y_train, y_test):
    #algorithms = [KNeighborsClassifier(), DecisionTreeClassifier(),RandomForestClassifier(), GaussianNB()]
    model = RandomForestClassifier()
    model.fit(X_train, y_train)
    y_test_pred = model.predict(X_test)
    y_tain_pred = model.predict(X_train)

    return y_tain_pred,y_test_pred, model

y_train_pred, y_test_pred, model = fit_predict(X_train, X_test, y_train, y_test)
```

Figure 18: Training the model

PREDICTION MODEL STAGE

The trained model is evaluated to determine its robustness and see how it holds up when the test data is introduced. We want to ascertain that the model makes the prediction to some certain level of accuracy. We will ensure that any case of overfitting or under fitting is eliminated. A model is termed as under fitted if the model when it does not correctly predict the test data set while it over fitted if the accuracy of the training data is good but does not accurately predict the test data, these scenarios are corrected by optimizing the model. We used a simple plot to compare the results of the what was predicted by the model and the actual values in the test data.

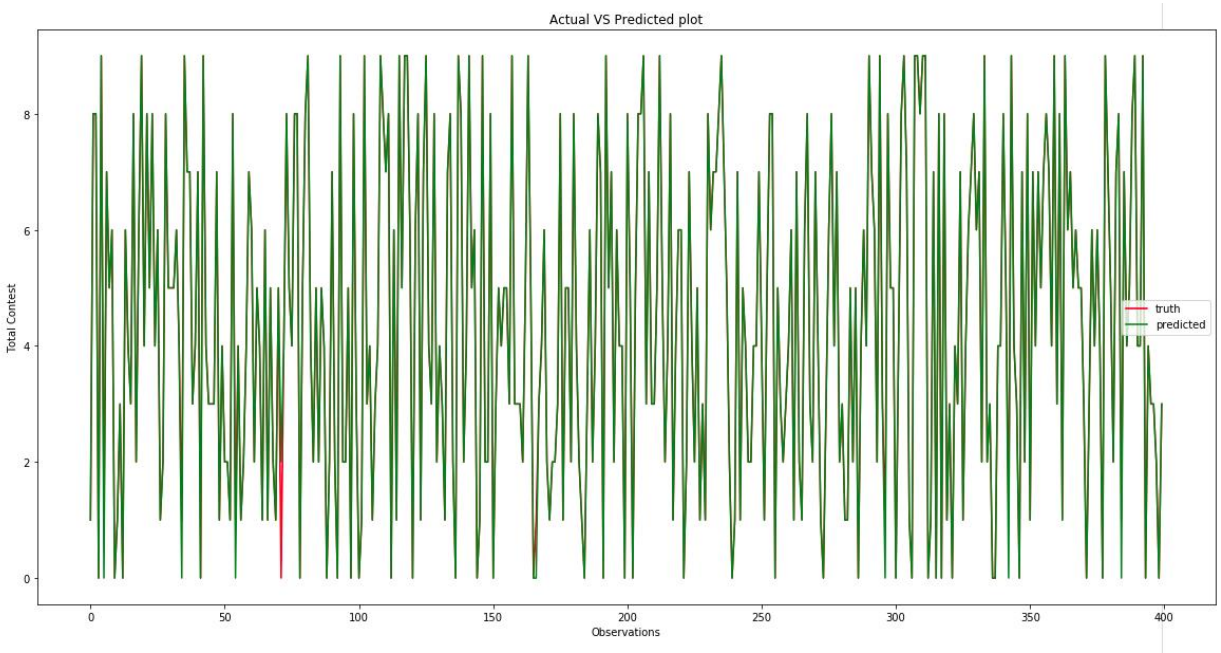

Figure 19: Actual Values Vs Predicted Values

MODEL EVALUATION

Now we evaluate our model to determine its performance using the test dataset. There are different metrics that can be used to evaluate the model but for our purpose we employed the confusion matrix.

The confusion matrix or error matrix describes the performance of a classification model using the test data. The matrix checks the ability of the model to adequately predict the collision hotspots based on the trained datasets. The confusion matrix is presented in figure 20. To further

analyze and evaluate the model we also perform k-fold cross validation. We ran about 10 folds that also yielded good accuracy (figure 21)

```
#model stat
def model_accuracy_stat(y_test, y_test_pred, y_train, y_train_pred):
    print('test accuracy: ', accuracy_score(y_test,y_test_pred))
    print('train accuracy: ', accuracy_score(y_train, y_train_pred))
    #print(classification_report(y_test, y_test_pred))
    print('confusion matrix: \n', confusion_matrix(y_test, y_test_pred))

model_accuracy_stat(y_test, y_test_pred, y_train, y_train_pred)
```

```
test accuracy:  0.9838515787587622
train accuracy:  0.9988805844214179
confusion matrix:
 [[386899  11965   3198     63    311     35      0     28      7      0]
 [ 40283 353043   8585    106    194     18      0     70      8      1]
 [   119     63 402410      5      9      3      0      0      0      0]
 [     0      0      0 402672      0      0      0      0      0      0]
 [     0      0      0      0 402400      0      0      0      0      0]
 [     0      0      0      0      0 403064      0      0      0      0]
 [     0      0      0      0      0      0 403458      0      0      0]
 [     0      0      0      0      0      0      0 403409      0      0]
 [     0      0      0      0      0      0      0      0 403588      0]
 [     0      0      0      0      0      0      0      0      0 403544]]
```

Figure 20: Confusion Matrix

```
#K-fold cross validation to check overfitting
def k_fold_cross_validation(num_of_folds, X_train, y_train):
    # check the same model with k-fold cross validation to make sure model is not overfitting
    model = RandomForestClassifier()
    cv_score = cross_val_score(model, X_train, y_train, scoring='r2', cv=num_of_folds)

    print(cv_score)

k_fold_cross_validation(10, X_train, y_train)
```

```
[0.99715213 0.99716062 0.99714679 0.99712423 0.99707414 0.99720198
 0.99716603 0.99719964 0.99714459 0.99713105]
```

Figure 21: K-fold cross validation

COMMUNICATE RESULT

This stage involves reaching out to our stakeholders and detailing the results of findings from undertaking this project. The results of this research will be communicated taking into consideration the different audiences that might be interested in our results. We will consider business executives who could include the different government agencies that can drive the deployment of our solution, we will provide this group of stakeholders a very high level results of our model and the predictions which they can quickly take actions upon. The other group of people that we will consider are IT analysts who have a strong technical background and would be interested the different processes we took to achieve the results and the different specifications to implement the codes in a development environment. These group of stakeholders will be provided with details of our code and the different rationale behind the different techniques we employed.

To adequately achieve this stage within our project, we prepared this well documented report that details the step by step process of completing this project, we provided two different presentations slides for business executives and analyst, we prepared a high level video presentation of our process and the results, we created a user guide that can help users interact effectively with the developed web application and we also provided the codes used for developing this project.

OPERATIONALIZE

The efforts we have made will be useless if the results of our analysis is not made available to stakeholders and user who need information about road accidents in Canada and as such we developed a web application that is user friendly and can be easily accessed by all users at all times. The application was developed using HTML and JavaScript, and we had our MONGODB as our database that interacted with the machine learning model developed in python. The images below show some screenshot of the web application.

At this stage the model and application is ready for deployment and we will continually monitor its performance and the user's performance while interaction with the application. Different users will be trained to use the application and the codes and the technical documents will be provided.

Fig 24: Launch the Web application

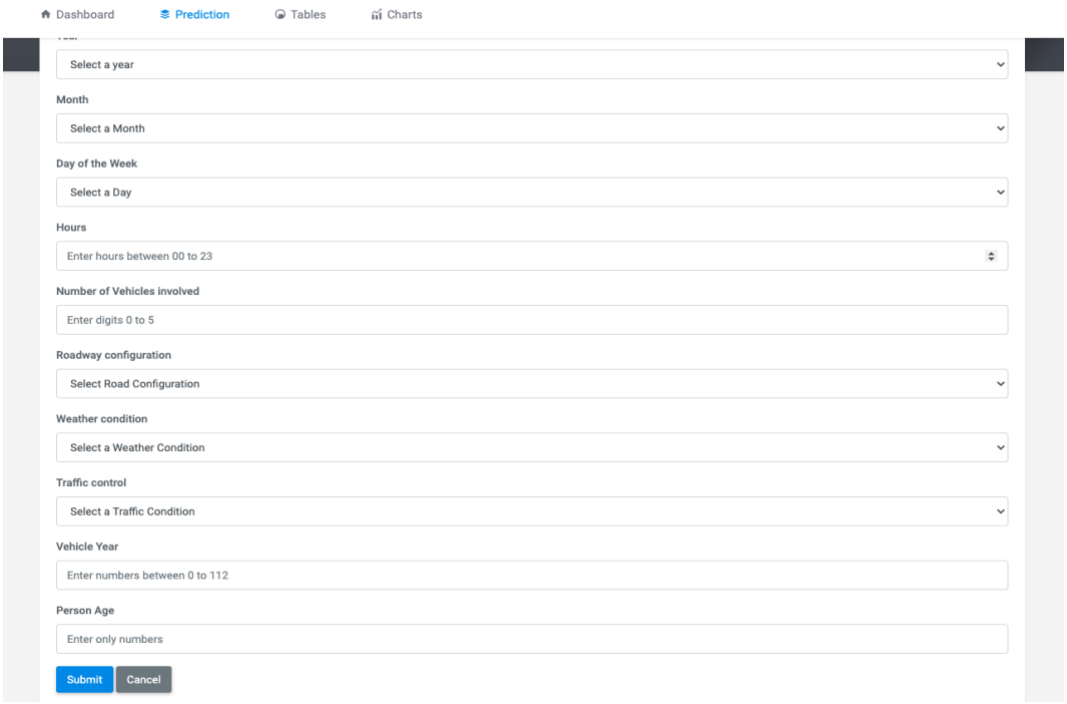

Fig. 25: Prediction Collision Hotspot

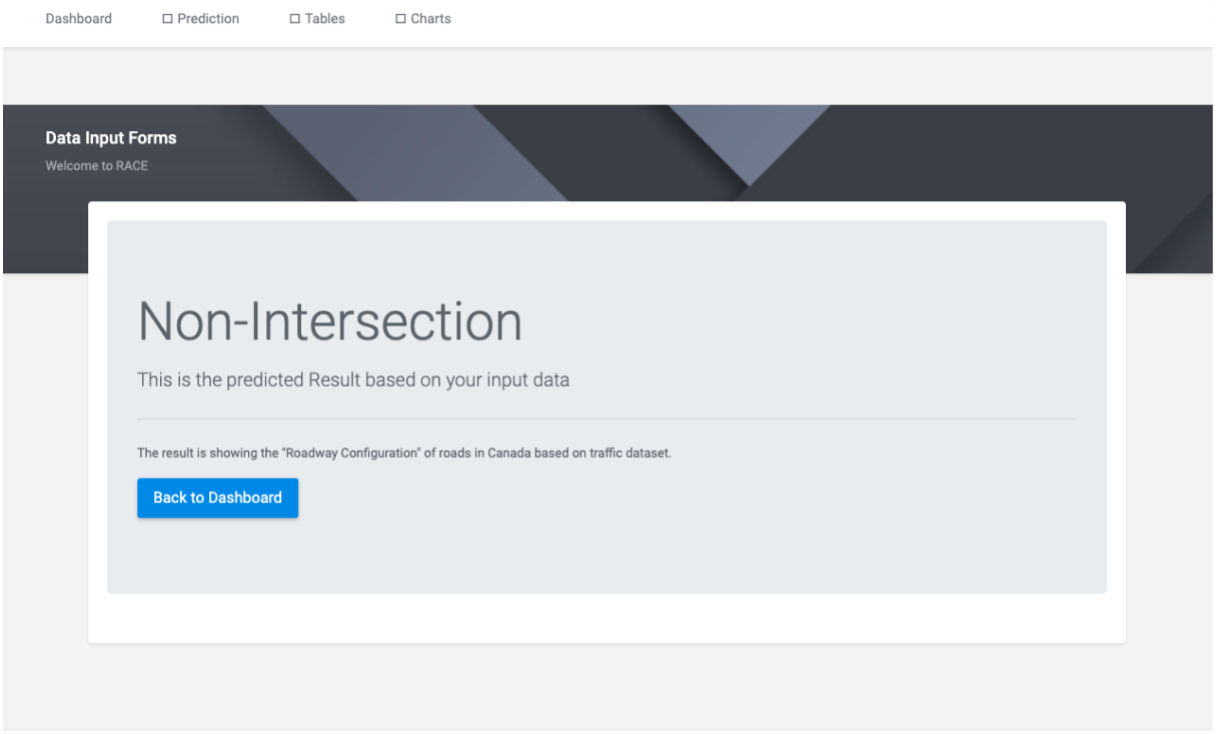

Fig. 26: Prediction Result

As a part of operationalization effort, we will train the technical analyst and different users on the step by step of how the project works. We will specify the factors considered for choosing the learning algorithm used and how the machine learning algorithm interacts with the web application. The technical team will have access to a setup guide which can help them to replicate the different processes or make new modification to the system. The user guide will also be available to train everyday users on the operation of the system.

LIMITATIONS

While we were to largely achieve the goals we set out at the beginning of the project, we encounter some challenges which are listed below;

- Large dataset, the selected data was large and storing and accessing it for both the python analysis and web application development was a great concern
- Due to the pandemic, during the execution of the project, we could not physically reach out to stakeholders to hold the necessary meetings and gather important requirements from them

- Lack of high end machines made our development process slower than expected building machine learning model and execute it in web application.

- The application stack (Flask, data analysis, mongodb) was completely new for us and we had been through a great learning curve during this project.

CONCLUSION

In this report we have highlighted the process within the data analytic lifecycle that we employed to execute the process of predicting road accidents at different identified hotspots within Canada. Our dataset was gotten from the government of Canada website and with this dataset we built a machine learning model using the random forest algorithm that can make prediction of the collision hotspots within Canada. We ultimately built a website where users can interact with different features of the dataset and the predictive model. All of our project files, codes and guides can be found the github repository provided here.

References

1. CCTMA, 2016. [online] Available at: <https://roadsafetystrategy.ca/en/strategy> [Accessed 4 July 2020].
2. Thecanadianencyclopedia.ca. 2018. *Humboldt Broncos Bus Crash | The Canadian Encyclopedia*. [online] Available at: <https://www.thecanadianencyclopedia.ca/en/article/humboldt-broncos-bus-crash> [Accessed 4 July 2020].
3. CBC News, 2018. *Scene Of Broncos Crash Haunted Sask. Truckers For Decades | CBC News*. [online] Available at: <https://www.cbc.ca/news/canada/saskatoon/scene-of-broncos-crash-haunted-sask-truckers-for-decades-1.4612253> [Accessed 5 July 2020].
4. Markhammitsubishi.ca. n.d. *Markham Mitsubishi | Most Dangerous Intersections In Canada*. [online] Available at: <https://www.markhammitsubishi.ca/en/news/view/most-dangerous-intersections-in-canada/62936> [Accessed 5 July 2020].
5. Kaya, N., Ayas, S., Ponnambalam, C. and Donmez, B., 2020. Visual Attention Failures during Turns at Intersections: An On-road Study. *CARSP Conference*.
6. Espinosa, M., 2015. SAFETY EVALUATION OF SIGNALIZED INTERSECTIONS WITH AUTOMATED VEHICLES AT VARIOUS PENETRATION LEVELS BASED ON CONFLICT ANALYSIS OF SIMULATED TRAFFIC. *CARSO Conference*.

7. Roos, J., 2016. Etiology of Motor Vehicle Collision Fatalities in Urban and Rural Canada. *CARSP Conference*.
8. Arason, N., 2019. The problem of cross over highway crashes and what can be done about them. *CARSP Conference*,.
9. Medium. 2020. *An Implementation And Explanation Of The Random Forest In Python*. [online] Available at: <https://towardsdatascience.com/an-implementation-and-explanation-of-the-random-forest-in-python-77bf308a9b76> [Accessed 18 August 2020].