SENG 550 Final Project Report

*Develop a classifier that can predict whether a given day will be ”good” or ”bad” in terms of rush hour commute on Deerfoot trail

1st Rui Guan
Schulich School of Engieering
University of Calgary
Calgary, Canada
rui.guan@ucalgary.ca

2nd Xiangyu Liu
Schulich School of Engieering
University of Calgary
Calgary, Canada
xiangyu.liu1@ucalgary.ca

3rd Zheng Chen
Schulich School of Engieering
University of Calgary
Calgary, Canada
zheng.chen1@ucalgary.ca

***Abstract*—This report is supposed to show our final project, which is about developing a classifier that can predict whether a given day will be ”good” or ”bad” in terms of rush hour commute on the Deerfoot trail. It was developed by analyzing various factors that are known to affect the rush hour commute on Deerfoot Trail, including the number of accidents that occurred on the trail every hour on a given day, the mean temperature, the total rain and total snow, and the thickness of any snow on the ground. The professor provided the data with three spreadsheets which respectively include the commute time of the Deerfoot trail from 4 a.m. to 10 p.m. from September 21, 2013, to April 10, 2014, and all the weather conditions of 2013 and 2014. The classifier was trained using a large dataset of historical data and was tested using a separate dataset to evaluate its accuracy. The results showed that the classifier could predict the ”good” or ”bad” nature of a given day with high accuracy, demonstrating its potential usefulness in helping commuters plan their rush hour travel on Deerfoot Trail.**

***Index Terms*—machine learning, classifier, commute time analysis, SVM, logistic regression, decision tree**

I. Preamble

A three-member group did this project, and everyone contributed to the project's completion. All members did their job together, so everyone contributed almost the same percentage (around 33% everyone). The link is below:
https://github.com/LAREINA-JO/SENG550Final

The declaration of all group members is shown below:

We, the members of the SENG550 Group, hereby declare that the following statement of contributions and estimate of total contribution is true and accurate.

Rui Guan 30072848,
Zheng Chen 30091064,
Xiangyu Liu 30067071

We confirm that the statement of contributions and estimate of total contribution accurately reflect the contributions made by each group member to the project.

We make this declaration willingly and under penalty of perjury.

Date: December 15, 2022
Signature:

Fig. 1. Signature from every member

II. Introduction

The following report presents the results of a study to develop a classifier that can predict whether a given day will be ”good” or ”bad” for a rush hour commute on the Deerfoot Trail. This is a significant problem that needs to be solved because the Deerfoot Trail is a major road in Calgary, and congestion on the Deerfoot Trail during peak hours can significantly impact people and cause significant delays and inconvenience to commuters. Developing a classifier that can accurately predict congestion on the Deerfoot Trail can help people plan their commute and avoid congestion. Previously, many mapping software developed some features, such as navigation tools like Google Maps or Waze, that show users how long the trip will take, calculate their estimated arrival time, and create the best route based on road conditions and predicted traffic[1]. Many logistics-related operations rely heavily on the accuracy of these calculations. Furthermore, there is strong government support for developing such software so that users can be aware of current road conditions and adjust road infrastructure to improve transportation efficiency and safety. However, there are some problems with the analysis of this mapping software; for example, most of them cannot analyze the current road conditions in conjunction with the current weather and the number of accidents on the road but analyze and provide results by road congestion. This is very one-sided, especially in cities with extreme weather conditions like Calgary, and the impact of weather on travel is very significant. So, support vector machines (SVM), logistic regression, and decision tree

algorithms will be used to develop a classifier developed by analyzing various factors known to affect commuting on the Deerfoot Trail during peak hours, including the number of accidents on the trail on a given day, mean temperature, total rain, and total snow, and snow cover thickness. Any snow accumulation on the ground.

III. Background and Related Work

A. Technical background helpful for understanding the report:

- Apache Spark[3]
- Resilient Distributed Dataset (RDD)
- Machine Learning Algorithm
- Machine Learning Evaluation Metrics

B. Review of existing work pertinent to the project:

- Project progress report: Until the progress report, the data has been understood and familiarized. A basic distribution visualization of the data is done. The Databricks platform is set up to work on the environment and try to run example code.
- Deerfoot Trail commute time and accident numbers: The Deerfoot Trail commute time and accident numbers are given from Sep 21st, 2013, to April 10th, 2014, in the CSV file.
- Deerfoot Trail weather conditions: The Deerfoot Trail weather conditions (temperature, snow, rain) are given from Jan 1st, 2013, to Dec 31st, 2014, in two separate CSV files.
- SENG550 Spark RDD tutorial: The Spark, RDD, and Machine Learning Models tutorials are provided from SENG 550 to demonstrate data preprocessing, basic model training, and evaluation examples.

IV. Methodology

A. Experiment setup

To set up an experiment to develop this classifier, collecting data on the relevant features and labels and then splitting the data into training and testing sets are necessary. The training set would be used to train the classifier, and the testing set would be used to evaluate the final performance of the classifier. Once the data are split into the appropriate sets, the machine learning algorithms will be selected and implemented: support vector machines (SVM), logistic regression, and decision trees. The classifier will be trained by the training data. Finally, using the testing data to evaluate the final performance of the classifier and determine its accuracy in predicting ”good” or ”bad” rush hour commutes.

B. Experimentation factors

- The type of machine learning algorithm used: It can have a significant impact on the performance of the classifier. Different algorithms have different advantages and disadvantages and may be better suited to the type of data and the classification task. For example, some algorithms, such as support vector machines (SVMs) and logistic regression, are well-suited for binary classification tasks. In contrast, others, such as decision trees, are better suited for multi-class classification tasks. Choosing the right algorithm for a given task can help improve the performance of a classifier while choosing the wrong algorithm can lead to poor performance.
- Training and testing datasets: As mentioned earlier, splitting the data into appropriate datasets is important to properly evaluate the classifier's performance. The training set is used to train the classifier, and the test set is used to evaluate the final performance of the classifier.
- Threshold: To evaluate the performance of a classifier, it is important to choose the appropriate threshold. Common metrics may include accuracy, precision, recall, and F1 score for such a binary classification task. These thresholds will be calculated using the test set's results to determine the classifier's overall performance.

V. Experiment process

A. Collect and prepare the data

Data were collected on the relevant features and labels to develop the classifier. Data on commute times and weather conditions were used and merged. It is important to ensure that the data is clean and consistent and that any missing or incorrect values are properly handled.

- First, all the data about the commute time of the Deerfoot trail from September 21, 2013, to April 10, 2014, were provided. Based on this dataset, the number of accidents was decided as the feature, and the diagram by commute time on Monday and the number of accidents (Figures 1 and 2) were drawn. Then the data distribution and change were observed from this diagram.
- Then, the new data about the weather condition in 2013 and 2014 were provided. After looking through the new dataset, the mean temperature, the total rain and snow, and the thickness of any snow on the ground were added to the features.
- Select useful columns, modify the date format of the weather conditions table, select the dates that overlap with the table of commute time, and union the two tables using the join method.

B. Split the data into training and testing sets

Once the data were collected and prepared, they were split into training and testing sets that used 70% of the data for training and 30% for testing.

- The model design was considered, as our data did not have a label (unsupervised), so using a cluster was considered at first.
- However, it was difficult to classify the cluster by good and bad. So, the next step was to generate a label by the

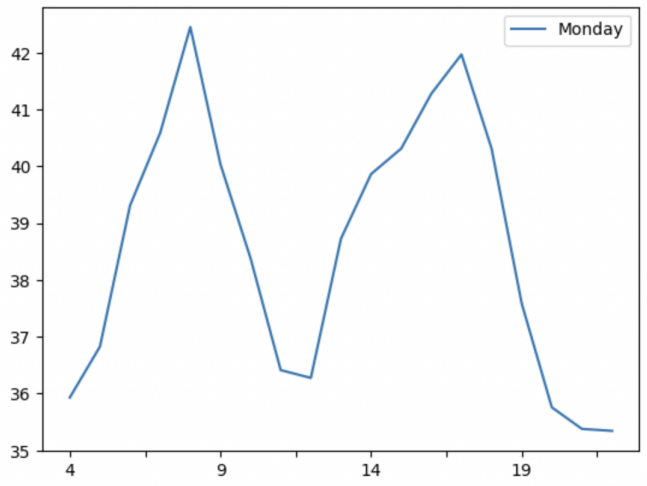

Fig. 2. Diagram of the average commute time of all Mondays from 4 am to 10 pm

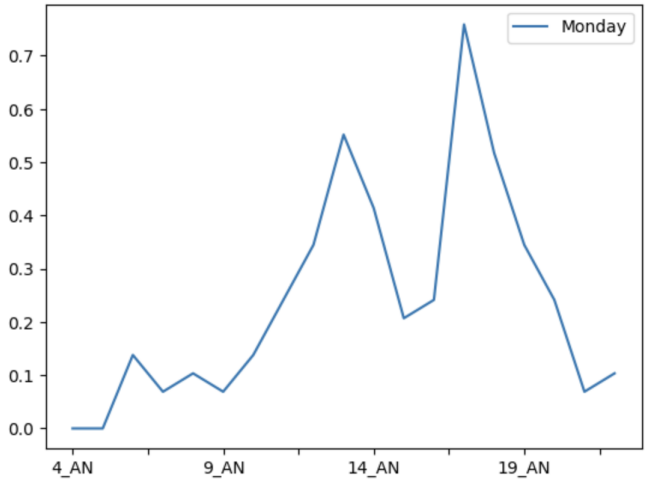

Fig. 3. Diagram of the average number of accidents all Monday from 4 am to 10 pm

average of all commute times, 37.811, and classify it by this label. If the commute time is greater than the label, it is bad (represented by 0), and the less one is good (represented by 1)).the data scaling didn't be considered at this time.

C. Select and implement the machine learning algorithms

Support vector machines (SVM), logistic regression, and decision tree algorithms were used to develop the classifier. First, run them under the default mode. The result models by these three algorithms are shown below:

- SVM:
  Confusion Matrix: [[29. 3.], [24. 11.]]
  RMSE score = 0.6348110542727384
  Accuracy scores = 0.5970149253731343
  Precision = 0.6717825964517037
  Recall = 0.5970149253731343
  F1 Score = 0.5604414900288472

- Logistic Regression:
  Confusion Matrix: [[30. 2.], [28. 7.]]
  RMSE score = 0. 6691496051182058
  Accuracy scores = 0. 5522388059701493
  Precision = 0. 653342482987362
  Recall = 0. 5522388059701493
  F1 Score = 0. 4846223428312981

- Decision Tree:
  Confusion Matrix: [[25. 7.], [11. 24.]]
  RMSE score = 0. 5183210553488161
  Accuracy scores = 0. 7313432835820896
  Precision = 0. 7361044241159793
  Recall = 0. 7313432835820896
  F1 Score = 0. 7311038390933035

D. Use normalization as the method of data scaling

The data of both SVM and logistic regression are relatively lower, especially the data of accuracy. Therefore, we started to consider data scaling. Normalizing our data is a good practice to follow when training machine learning models because normalization scales the data to a common range, which can make the optimization process more stable.[2]

- Because there is a considerable number of features, these features are measured in different units and in different ranges, which can affect the model's performance. Normalizing the data will bring all the features to the same scale, which can help the model learn and weigh features more effectively. The models' results after data normalization by three algorithms are shown below:
- SVM:
  Confusion Matrix: [[22. 10.], [8. 27.]]
  RMSE score = 0. 5183210553488161
  Accuracy scores = 0. 7313432835820896
  Precision = 0. 7314508538389135
  Recall = 0. 7313432835820896
  F1 Score = 0. 7307414540202215

- Logistic Regression:
  Confusion Matrix: [[22. 10.], [7. 28.]]
  RMSE score = 0. 5183210553488161
  Accuracy scores = 0. 746268656716418
  Precision = 0. 7472438172115827
  Recall = 0. 7462686567164178
  F1 Score = 0. 7452430191284762

- Decision Tree:
  Confusion Matrix: [[25. 7.], [11. 24.]]
  RMSE score = 0.5183210553488161
  Accuracy scores = 0.7313432835820896
  Precision = 0.7361044241159793
  Recall = 0.7313432835820896
  F1 Score = 0.7311038390933035

VI. Result and Improvement

TABLE I
THE SCORES FOR THREE MODELS BEFORE DATA PREPROCESS

Model	RMSE	Accuracy	Precision	Recall	F1 Score
SVM	0.6348	0.5970	0.6718	0.5970	0.5604
LG	0.5183	0.5522	0.6533	0.5522	0.4846
DT	0.5183	0.7313	0.7361	0.7313	0.7311

TABLE II
THE SCORES FOR THREE MODELS AFTER DATA PREPROCESS

Model	RMSE	Accuracy	Precision	Recall	F1 Score
SVM	0.5183	0.7313	0.7315	0.7313	0.7307
LG	0.5183	0.7463	0.7472	0.7463	0.7452
DT	0.5183	0.7313	0.7361	0.7313	0.7311

Compared with the scores for these three models before normalization, it can be seen that the scores for SVM and LG is relatively low, but the scores for the decision tree are high. It demonstrated that the distribution of the data is not neat. SVM and LG algorithms require data preprocessing. It also proved that the Decision tree works well without data preprocessing.

Compared with the scores for the three models after normalization, it can be found that the LG performs the best. LG has the highest accuracy and F1 score among the three models. It may be because that logistic regression is a useful tool for classification tasks when the response variable is binary, and it is particularly well-suited for working with numerical data.

Therefore, the Logistic Regression model performs the best work. Then the next step should be how to improve this model.

To improve this model, the best tuning parameter for a logistic regression model should be found. An algorithm was created that will iterate the values of the important parameters for us. The parameters should be chosen to be regParam (from 0, 0.1, 0.01, 0.001), regType (from l1, l2, None), intercept (from True, False), validateData (from True, False). After the 48 iterations, the best accuracy score of 0.7463 was received. The parameter for it is: regParam=0.1, regType=l1, intercept=False, validateData=True.

TABLE III
THE SCORES FOR THREE MODELS AFTER DATA PREPROCESS

Model	RMSE	Accuracy	Precision	Recall	F1 Score
LGRraw	0.5183	0.5522	0.6533	0.5522	0.4846
LGImproved	0.5183	0.7463	0.7472	0.7463	0.7452

From the table above, it can be seen that all the scores improved a lot. However, the main improvement is caused by data scaling. The parameters do not affect the improvement. The other parameter is also tried. For example, accident numbers and snow on the ground looks will affect the commute time more. However, there is no effect on the accuracy even if the weight of these parameters is increased.

The figure above demonstrates the predicted and true values after improving the logistic regression model by tuning

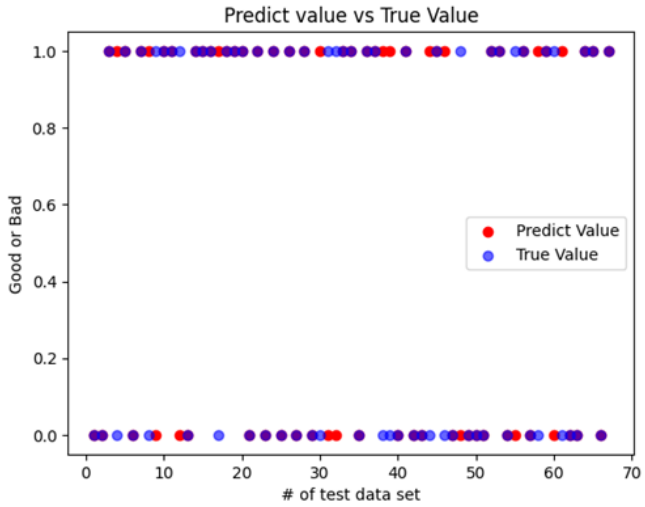

Fig. 4. Diagram of Predict Values vs. True Values from Logistic Regression with Tuning Selected Parameters and Normalized Dataset

selected parameters and normalized datasets. The legend for predicted values and true values are correspondingly in red and blue. Points in purple color mean that the predicted value matches the true value, i.e., true positive.

VII. Conclusion and Future

A. Conclusion

In this project, we created a classifier that predicts whether the day is good or bad in terms of rush hour commute on the Deerfoot trail. We combine the Deerfoot commute dataset with Calgary weather. There are 5 features we take from the dataset. They are total accident number, min temperature, total rain, total snow, and snow on the ground. We do not have any labels. However, since unsupervised learning is not suitable for our topic. We decide to generate a label by using the commute time values. Since the variance for the commute time is not big, we decide to calculate the total min value of commute time, which is 37.811. Then, we set the commute number that is bigger than 37.811 to bad (we encode it to 0). We set the commute number bigger than 37.811 to good (we encode it to 1). Then we choose three algorithms that are good to use in this situation. There are SVM, Logistic Regression and Decision Trees. Under the default mode, only the decision tree has high accuracy, and the other two are not accurate enough. Obviously, the preprocessing of the data can have a potential impact on these models. Therefore, we do normalization to all three models' data. From the comparison between the data before normalization, we can find that normalizing the data is important for improving the performance of SVM and logistic regression. Meanwhile, the decision tree model is not sensitive to normalizing because its results didn't change. These results suggest that logistic regression is the most effective model for predicting the binary outcome based on the normalized features, with an

average accuracy of 0.7463 and an F1 score of 0.7452. The SVM model had an average accuracy of 0.7313 and an F1 score of 0.7452, while the decision tree model had an unchanged average accuracy of 0.7313 and an F1 score of 0.7311. However, it is worth noting that the other two models may still have value in certain contexts, particularly if some requirements or constraints make one of these models more suitable. In this experiment, the final chosen model, Logistic Regression, was minimally affected by the parameters. After analysis and discussion, the group gave the following reasons. Firstly, the model is already well-tuned. If the model is already performing well, then adjusting the parameters might not significantly impact its performance. Secondly, the model is not sufficiently complex. If the model is too simple, it may not have enough capacity to learn the underlying pattern in the data. Third, since the labels for this data were set by the group itself. This label lacks reasonableness. The criterion for distinguishing good and bad traffic times is 37.811. So, when the potential time corresponding to the data is far from the criterion (e.g., it is 34), changing the parameter at this time cannot affect the label of this data set from good to bad even if it will affect the potential time (e.g., it grows to 37). Fourth, there may be a lack of correlation between feature and label. labels are not sensitive to changes in features.

B. Future Work

This project still has considerable potential and, meanwhile, has some concerning problems. Firstly, a more detailed model can consider traffic at an hour of the day than a one-day forecast and analysis. The commuting time analysis for a certain period is more valuable and relevant. People usually plan their routes through mapping software before traveling. Secondly, the label of the current project is decided based on the data set's average. This will have poor compatibility and adaptation behavior for other roads or larger datasets. The future labels can be relative to seasons, rush hour, weekdays, and time intervals of the day. Thirdly, the dataset used in this project is now limited. The current dataset does not cover a full annual period, which results in missing data for a particular season. This has a potential impact on temperature, rainfall, and people's travel habits. Fourthly, only three machine learning models were used in this project. With more data and a more accurate way of pre-processing, there are more models to train and analyze.

REFERENCES

[1] Editor, "Traffic prediction: How machine learning helps forecast congestions and plan optimal routes," AltexSoft, 27-Jan-2022. [Online]. Available: https://www.altexsoft.com/blog/traffic-prediction/. [Accessed: 15-Dec-2022].

[2] Simplilearn, "What is normalization of data in database?: Simplilearn," Simplilearn.com, 12-Dec-2022. [Online]. Available: https://www.simplilearn.com/automated-recruiting-in-companies-article. [Accessed: 15-Dec-2022].

[3] "Linear Methods-RDD-based API," Linear Methods RDD-based API-Spark 3.3.1 Documentation, 14-Jul-2014. [Online]. Available: https://spark.apache.org/docs/latest/mllib-linear-methods.html#classification. [Accessed: 15-Dec-2022].