Predictive Analysis of NewYork Motor Vehicle Collisions

Anusha Manjappa, Prathibha Gubbi Prakash, Srihitha Reddy Sivannagari, Jongwook Woo
Department of Information Systems, California State University
Los Angeles

E-mail: amanjap@calstatela.edu, pgubbip@calstatela.edu, ssivann@calstatela.edu, jwoo5@calstatela.edu

Abstract*:* In this paper, we have performed a predictive analysis of the New York motor vehicles collisions dataset to estimate the pattern of accidents based on suitable features using AzureML and SparkML. We have selected a dataset from the New York government website, 'New York open data' which contains detailed information about motor vehicle collisions. We collected nearly three years of data ranging from 2013-2016. The initial raw data size was 215MB.After cleaning the unwanted values, columns and rows with missing values, ambiguous data, etc., the data size reduced to 183MB.The data set we used has detailed information about not only the vehicle that caused the incident but, also the other vehicles that were involved in the accident,the contributing factors and the Geo-spacial location in which the accident happened. This helped us make more accurate predictions.

1. Introduction

The rapid growth in the technologies used by leading and yet budding companies is demanding a strategic approach to success. One such common emerging methodology is the predictive analysis. Applying predictive analytics across business functions, the enterprise achieves multiple strategic objectives. Predictive models generated from enterprise data are integrated with business units across the organization, including marketing, sales, fraud detection, the call center and core business capacity. An organization needs predictive analytics because strategic objectives can be attained to their full potential only by employing them [1].

2.Related Work

IBM Data science has a sample notebook which demonstrates visualizations of the New York motorcycle collisions. These are the normal visualizations done for data analysis using tools like Tableau, Excel, Power BI and so on. The existing paper has shown graphical representations for understanding the count of incidents across different borough, comparison of the number of accidents in different places, popular contributing factors leading to the accidents using Geo-spatial maps, bar graph, scatter plot. But this paper is missing predictive analysis over this dataset to derive some inference. As an extension to this, we carried out predictive analysis. Based on existing pattern we tried predicting the answer for other cases before they are explicitly observed.

3. Machine Learning in Spark Ml and Azure Ml

The Machine Learning field evolved from the broad field of Artificial Intelligence, which aims to mimic intelligent abilities of humans by machines. Important question that one considers in the field of machine learning is how to make machines able to "learn". Learning in this context is understood as inductive inference, where one observes examples that represent incomplete information about some "statistical phenomenon" [5].

3.1 Machine Learning in Spark

The scalable machine learning library of Spark is MLlib. It contains general learning utilities and algorithms, which include regression, collaborative filtering, classification, clustering, dimensionality reduction, and underlying optimization primitives. Spark also provides many language choices, including Scala, Java, Python, and R [3].

Cluster specifications:

- Apache Spark Version - Spark 2.1
- File System – DBFS (Data Bricks File System)
- 6GB Memory, 0.88 Cores, 1DBU

3.2 Machine Learning in Azure

Azure Machine Learning is a cloud predictive analytics service that makes it possible to quickly create and deploy predictive models as analytics solutions. You can work from a ready-to-use library of algorithms, use them to create models on an internet-connected PC, and deploy your predictive solution quickly [4].

Azure Specifications:

- Cloud platform
- 10GB Memory

4. Dataset Details

This dataset contains details of Motor Vehicle Collisions in New York City provided by the Police Department (NYPD).
URL: https://data.cityofnewyork.us/Public-Safety/NYPD-Motor-Vehicle-Collisions/h9gi-nx95

Data specifications:

- File Size:215MB.
- Number of Files: 1.
- File Format – CSV (Comma Separated Values).

5. Our Work

In this paper, we have done two types of predictive analysis. Firstly, we inferred the borough in which the accidents had occurred using features like vehicle type code, number of persons killed and the contributing factor of the accidents in azure ML. Second, we predicted the number of accidents using the borough in which the accidents had occurred in spark ML.

We conducted these predictive analysis experiments on two platforms, namely Azure ML and Spark ML.

5.1 Azure ML Analysis:

On observing our dataset which only had independent and non-continuous values, we decided on considering classification models for our predictive analysis. In supervised learning, we have labels associated with every entity considered. If the label is found to be discrete, then it is considered a classification problem – otherwise if it is a real valued label we consider it a regression problems. Accordingly,we considered three classification models for our predictive analysis in AzureML. The three models we have considered are multiclass decision forest classification,multiclass decision jungle classification and multiclass neural network classification.

5.1.1 Selection of Labels and Features

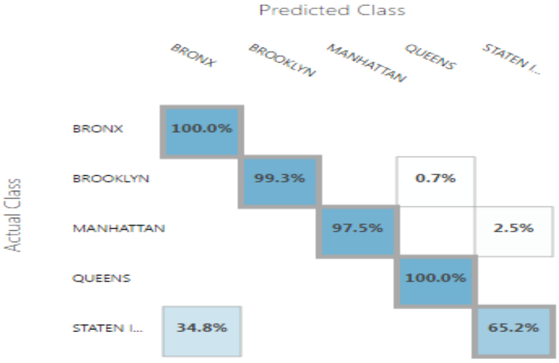
We selected a few columns which are most appropriate in predicting the labels. Our initial analysis on the dataset helped us decide of the columns which were best suited to be featured.We chose vehicle type code, number of persons killed and contributing factor to be the features in our experiment to predict our label being the borough in which the accident happened.

5.1.2 Model Construction Process

The dataset was loaded into azureML in the form of a .csv file. The columns that were not being used for our predictive analysis were elminated using the select column module. The filtered out columns are then sent to the split model module to split the dataset to a ratio of 7:3 where 70% of the data is sent to the train model and the rest 30% of the data is used up for testing. Then we train our selected label using three different classification model. Score models are included to check the predicted values.We then evaluate model using by comparing the predicted value against the actual value. Evaluate model also gives us the accuracy, precision and recall values after comparison.

5.1.3 Multiclass Decision Forest Classification Model

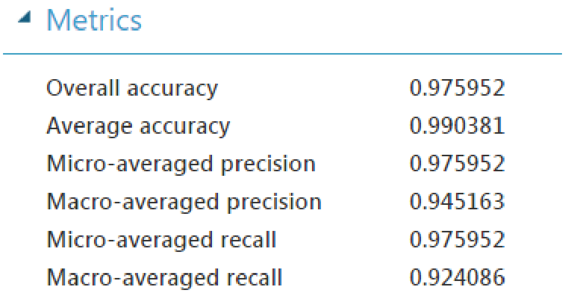
The multiclass decision forest algorithm is a learning method for classification. The algorithm works by building multiple decision trees and then voting on the most popular output class[2]. The confusion matrix derived on evaluating this model is shown below:



	BRONX	BROOKLYN	MANHATTAN	QUEENS	STATEN I...
BRONX	100.0%				
BROOKLYN		99.3%		0.7%	
MANHATTAN			97.5%		2.5%
QUEENS				100.0%	
STATEN I...	34.8%				65.2%

Figure 1: Confusion matrix for multiclass decision forest classification model

The accuracy, precision and recal values of this model are shown below:



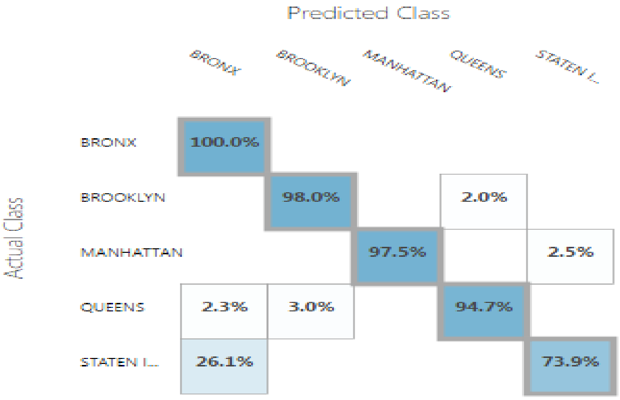
◢ Metrics

Overall accuracy	0.975952
Average accuracy	0.990381
Micro-averaged precision	0.975952
Macro-averaged precision	0.945163
Micro-averaged recall	0.975952
Macro-averaged recall	0.924086

Figure 2: Evaluation metrics for multiclass decision forest classification model

5.1.4 Multiclass Decision Jungle Classification Model

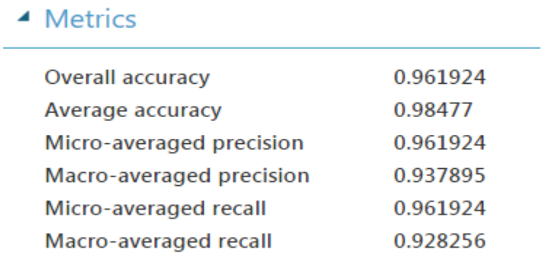
The Multiclass Decision Jungle module is used to create a machine learning model that is based on a supervised learning algorithm called the decision jungles. The model can be used to predict a target that has multiple values [2].The confusion matrix derived on evaluating this model is shown below:-



	BRONX	BROOKLYN	MANHATTAN	QUEENS	STATEN I...
BRONX	100.0%				
BROOKLYN		98.0%		2.0%	
MANHATTAN			97.5%		2.5%
QUEENS	2.3%	3.0%		94.7%	
STATEN I...	26.1%				73.9%

Figure 3: Confusion matrix for multiclass decision jungle classification model

The accuracy ,precision and recal values of this model are shown below:



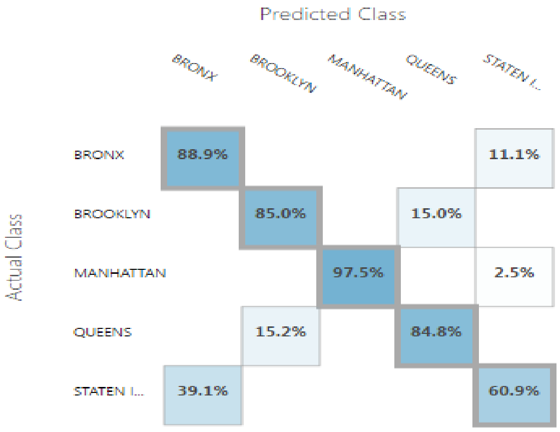
Metrics

Overall accuracy	0.961924
Average accuracy	0.98477
Micro-averaged precision	0.961924
Macro-averaged precision	0.937895
Micro-averaged recall	0.961924
Macro-averaged recall	0.928256

Figure 4: Evaluation metrics for multiclass decision jungle classification model

5.1.5 Multiclass Neural Network Classification

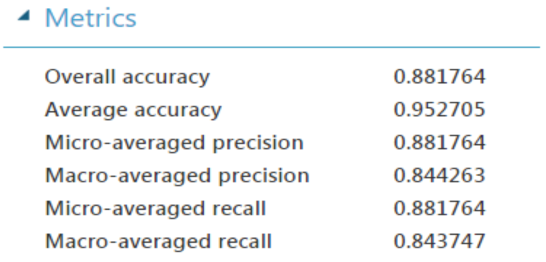
Multiclass Neural Network module is used to create a neural network model that can predict a target that has multiple values. Classification using neural networks is a supervised learning method, and therefore requires a tagged data set that includes a label column[2].The confusion matrix derived on evealuating this model is shown below:-



	BRONX	BROOKLYN	MANHATTAN	QUEENS	STATEN I...
BRONX	88.9%				11.1%
BROOKLYN		85.0%		15.0%	
MANHATTAN			97.5%		2.5%
QUEENS		15.2%		84.8%	
STATEN I...	39.1%				60.9%

Figure 5: Confusion matrix for multiclass neural network classification model

The accuracy ,precision and recal values of this model are shown below:-



Metrics

Overall accuracy	0.881764
Average accuracy	0.952705
Micro-averaged precision	0.881764
Macro-averaged precision	0.844263
Micro-averaged recall	0.881764
Macro-averaged recall	0.843747

Figure 6: Evaluation metrics for multiclass neural network classification model

5.1.6 Summary of all the classification models in AzureML

Multiclass decision forest classification model turned out to be the best model out of the three as it had the best accuracy of 0.975952, precision of 0.945163 and recall of 0.924086.

5.2 Spark ML Analysis:

The three classification algorithms that we have used in SparkML are Random forest classifier,Decision tree Classifier and Logistic regression.

5.2.1 Selection of Labels and Features

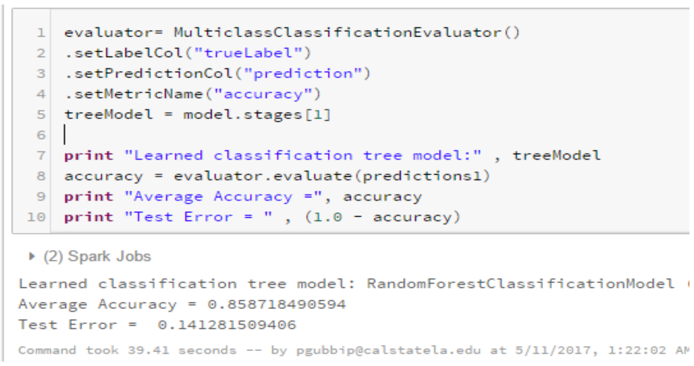
We have created a custom field Incident which has the count of the total number of people injured in each accident.We have tried to predict this custom column as a label having borough as a feature.

5.2.2 Model Construction Process

We loaded the source data into a table and prepared the data with features and labels.The data was split with a ratio of 7:3.The label was then trained using three classifications models to see which model predicted the number of incidents most accurately.

5.2.3 Random Forest Classifier

Multiclass classification evaluator has been used to evaluate this model and we have got a good accuracy of 0.85871.The error in prediction was just 0.14 as shown below.



```
evaluator= MulticlassClassificationEvaluator()
.setLabelCol("trueLabel")
.setPredictionCol("prediction")
.setMetricName("accuracy")
treeModel = model.stages[1]

print "Learned classification tree model:" , treeModel
accuracy = evaluator.evaluate(predictions1)
print "Average Accuracy =", accuracy
print "Test Error = " , (1.0 - accuracy)
```

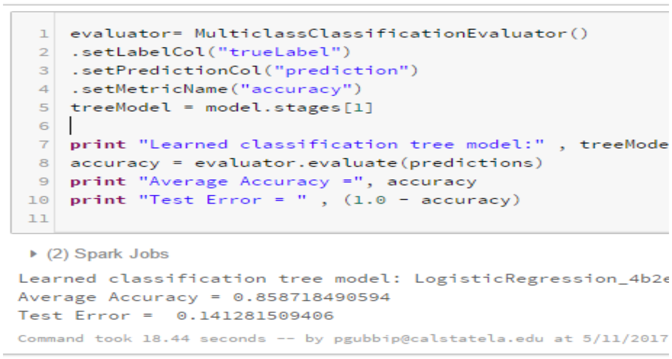
```
Learned classification tree model: RandomForestClassificationModel
Average Accuracy = 0.858718490594
Test Error =  0.141281509406
```

Command took 39.41 seconds -- by pgubbip@calstatela.edu at 5/11/2017, 1:22:02 AM

Figure 7: Evaluation metrics for random forest classifier model

5.2.4 Logistic Regression

Multiclass classification evaluator has been used to evaluate this model and we have got a good accuracy of 0.85871.The error in prediction was just 0.14 as shown below



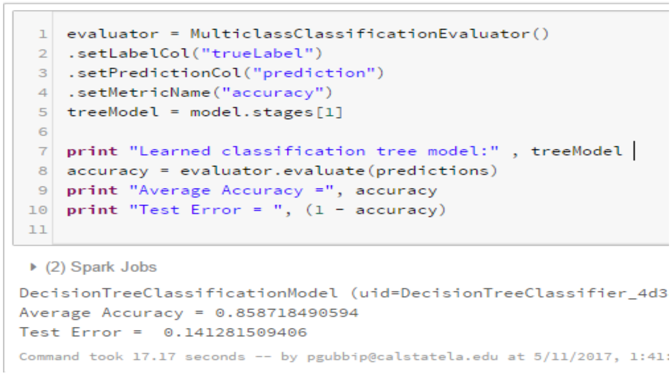
```
evaluator= MulticlassClassificationEvaluator()
.setLabelCol("trueLabel")
.setPredictionCol("prediction")
.setMetricName("accuracy")
treeModel = model.stages[1]

print "Learned classification tree model:" , treeMode
accuracy = evaluator.evaluate(predictions)
print "Average Accuracy =", accuracy
print "Test Error = " , (1.0 - accuracy)

```

```
Learned classification tree model: LogisticRegression_4b2e
Average Accuracy = 0.858718490594
Test Error =  0.141281509406
Command took 18.44 seconds -- by pgubbip@calstatela.edu at 5/11/2017
```

Figure 8: Evaluation metrics logistic regression classification model

5.2.5 Decision Forest Classifier

Multiclass classification evaluator has been used to evaluate this model and we have got a good accuracy of 0.85871.The error in prediction was just 0.14 as shown below



```
evaluator = MulticlassClassificationEvaluator()
.setLabelCol("trueLabel")
.setPredictionCol("prediction")
.setMetricName("accuracy")
treeModel = model.stages[1]

print "Learned classification tree model:" , treeModel
accuracy = evaluator.evaluate(predictions)
print "Average Accuracy =", accuracy
print "Test Error = ", (1 - accuracy)

```

```
DecisionTreeClassificationModel (uid=DecisionTreeClassifier_4d3
Average Accuracy = 0.858718490594
Test Error =  0.141281509406
Command took 17.17 seconds -- by pgubbip@calstatela.edu at 5/11/2017, 1:41:
```

Figure 9: Evaluation metrics for decision forest classifier model

6. Summary of the most accurate models in Azure ML and SparkML

Model Type	AzureML	SparkML
Classification	Multiclass decision forest Accuracy: 97.59	Random Forest Classifier Accuracy:85.9%

Table 1: Summary report of AzureML and SparkML

This table represents the resulting evaluation metrics of the most accurate classification models in Azure Machine Learning and Spark Machine Learning.

7. Conclusion

The predictions of motor vehicle accidents in New York using Classification models are compared in both Azure ML and Spark ML (Databricks). We could infer that Multiclass Decision Forest was the best model for our dataset in AzureML with an accuracy of 97% and Random forest classifier predicted the best values for our dataset in SparkML with an accuracy of 85.9%.These predictions will in future would definitely help the New York government to take respective measures to reduce motor vehicle accidents

7. References

[1].https://www01.ibm.com/common/ssi/cgibin/ssialias?htmlfid=YTW03080USEN

[2].https://msdn.microsoft.com/enus/library/azure/dn906015.aspx

[3].https://www.simplilearn.com/spark-ml-programming-tutorial-video

[4]. https://docs.microsoft.com/en-us/azure/machine-learning/machine-learning-what-is-machine-learning

[5]. https://events.ccc.de/congress/2004/fahrplan/files/105-machine-learning-paper.pdf

Dataset URL

https://data.cityofnewyork.us/Public-Safety/NYPD-Motor-Vehicle-Collisions/h9gi-nx95

GitHub URL

https://github.com/anushamanjappa/CIS5560-Big-Data-Analysis