**Kapil Bastola  
02/21/2017  
DS-670  
Assignment 5**

**Method (Algorithm Overview and Implementation in Zeppelin)**

**Flowchart****Method**

**Data Retrieval and Combining**

Above is the suggested flow chart of the process I am going to employ for the capstone project. The dataset I was assigned for the capstone class is the Road Traffic data from CityPulse dataset collection. This dataset consists of data from traffic in a city called Aarhus in Denmark. This dataset is collection of traffic data between two points for certain duration of time. There are four different datasets available in the CityPulse website for different durations. The first dataset is from February 2014 to June 2014, second dataset is from August 2014 to September 2014, the third dataset is from October 2014 to November 2014 and the fourth dataset is from July 2015 to October 2015. The datasets are available in two formats. One is a zipped file with collection of csv files while the other format is semantically annotated format.

I will download all 4 collections of CSV files from the website and combine the datasets into one data frame or similar structure using Spark through a zeppelin notebook. There are 449 observation points in total. So we have 449 files for each time frame and we are combining 1796 different raw files into one table.

**Data Exploration and Descriptive Statistics**

Datasets consists of 9 different columns with a mix of numerical, categorical and date time variables. With SQL and R in zeppelin, I will look at the summary of each variable. For categorical variables I will look at the number of factors or levels in the variables. For the numerical variables I will look at the distribution and variation of each variable along with measures of central tendencies like mean and median. For the numerical variables I will also look at any correlation that might exist between two variables. This will help me in sorting out the variables. I will look at histograms for distribution and scatterplots to evaluate correlations between numerical variables.

**Labeling of Dataset**

In order to perform supervised learning algorithms, we need to train the dataset using labels. In this dataset there are no labels present. Average speed column in the dataset can be used to label the dataset as speed represents how fast the cars are going and this tells us how congested the road is. If we want to predict traffic congestion, velocity of cars in the road can be considered the label we want to predict. Traffic congestion can be broken down in 3 different levels, heavy, mild, and low. For our algorithm, we will assign average speed of less than 30 kilometers per hour (kph) to be heavy traffic congestion, average speed of between 31 and 60 kph to be mild traffic congestion and average speed of 60 kph or higher to be low traffic congestion. Using these labels we can now train our models and predict for any new data.

**Splitting Dataset (Training and Test Sets)**

For building any supervised learning models we can break up datasets into a training set and a testing set. Only training set is used to train the classification model. We train and tune our model using the training set and test how well the model can generalize using the test set with data that the model has never seen. One common way of splitting the dataset is by using 10% of dataset for testing and 90% of dataset for training. This may lead to some bias in the classification result and the model we built may not necessarily be generalizable. In order to prohibit that we can use another well accepted method called N-Fold cross validation, in which you randomize the dataset and create N number of almost equal size partitions. Then we can choose the Nth partition for testing and remaining partitions for training the classifier. Within the training set we can further employ another K-fold cross validation to create a validation set and find the best parameters. And repeat this process N time to get an average of the metric. Since we want to get rid of classifier bias we repeat this above process certain number of times by randomizing data and splitting into N fold and take average of the metric. This will result in a non-biased classification model that we can generalize.

**Machine Learning Algorithms/ Predictive Models**

Machine learning algorithms can be generalized into two different types, supervised learning and unsupervised learning. In supervised learning, we train a model for each input with a corresponding target and later predict target for any new input. If the targets are in distinct classes we call it a classification model and if the target is continuous we call it a regression model. Where as in unsupervised learning there are no targets. We evaluate the relationship between different inputs and their structure in unsupervised learning. One of the most important unsupervised learning methods is clustering, where we group input data based on the inherent structure of those inputs and build a model to place a new input data in one of the groups created.

For the purpose of the study we will look into following machine learning algorithms:

**Decision Trees**

Decision tree is an algorithm used for building classification models whose output looks like a tree structure. Decision tree consists of root node, test node and decision nodes (leaf node). A decision tree is a flowchart-like structure in which each internal node represents a test on an attribute, each branch represents the outcome of the test and each leaf node represents a class label, which is the decision taken after computing all attributes. The paths from root to leaf represent classification rules. In decision analysis a decision tree and the closely related influence diagram are used as a visual and analytical decision support tool, where the expected values of competing alternatives are calculated. Data is classified by a decision tree by following the route from root node to the decision node according the set attributes or criteria. Once the decision tree is build we start at the root node to apply the testing scenario and follow the branch that fits the scenario, ending up at one of the leaf nodes which are one of the classes of the classification model. We can use Rpart package in R through zeppelin to perform decision trees algorithms.

**Support Vector Machines**

Support Vector Machines (SVM) is another widely used algorithm for supervised learning as can be used for both classification and regression. The basic idea of SVM is to map the data into a high-dimensional feature space via a nonlinear mapping. A linear learning machine in a kernel induced feature space learns a non-linear function while the capacity of the system is controlled by a parameter that does not depend on the dimensionality of the space. With a given training data set that are marked with labels, SVM training algorithm builds a model that assigns new data point into a specific category. We can use e1071 package in R through zeppelin to build SVM model.

**Random Forest**

Random Forest is a machine learning algorithm that uses decision trees. Random forest does a collective classification using decisions from different decision tress. We don’t make decision based on just one decision tree, but by an almost unanimous prediction made by a set number of decision trees. A random forest consists of thousands of individual trees that are trained on different part of a training set. Random forest provides very good accuracy compared to other classification algorithms and it can be very efficient for large data sets, which makes this algorithm very scalable. Another benefit of random forest algorithm is that it can handle many variables at one time thus allowing the model to be built for high dimensional data. We can use randomForest package in R through zeppelin to build a Random Forest model.

**Time Series Analysis: Autoregressive Integrated Moving Average (ARIMA)**

Since the dataset we are looking at is time based we can conduct time series analysis to predict future points in the series. We can also look into seasonality as we may guess that the traffic data changes during the course of the day and the week. We can use the velocity data to predict future velocity or number of vehicles at a given time using time series models like Auto-Regressive Integrated Moving Average (ARIMA) model. The auto correlation function (ACF) and partial auto correlation function (PACF) can be used to determine which order of auto regression and moving average is required for the model. By predicting velocity at a certain time we can make suggestions as to what time are busy and when to be careful when using the roads. If the ACF shows any significance at any of the lags we use that order of lag for the moving average and if PACF shows any significant spikes at any of the lags we use that order of lags for the auto regression. After we set the order for the model we can generate the model using R through Zeppelin. We can then predict next series of data points based on the model. In order to predict the accuracy of the model we can use mean average predicted error (MAPE).

**Evaluation of Model**

When we are done with building of the different algorithms we need to decide which algorithm gives us the best result. In other words, we need to evaluate our models. We use the model on the test dataset to get the prediction from the model and compare them against the expected labels that we had in hand. There are many performance metrics that help us evaluate the performance of our models.

For binary classification, we can look at the number of true positives (tp), which are the correct affirmative predictions, true negatives (tn), which are the correct negative predictions, false positives (fp), which are the incorrect affirmative predictions, and false negatives (fn), which are the incorrect negative predictions. Using these values we can calculate different measures of model performance like accuracy (tp + tn)/ (p + n), precision (tp / (tp + fp)), recall (tp / (tp + fn)), specificity (tn / (fp + tn)), fall-out (fp / (fp + tn)), F1 score (2 \* tp / (2\*tp + fp + fn)), etc.

We can also look at a receiver operating characteristic curve or ROC curve to evaluate model performance. ROC curve can be plotted for different models to visualize which model is performing the best. The x-axis of the ROC curve is false positive rate while the y-axis is true positive rate. The ROC curve is hence the recall as a function of fall out. We look at area under the curve (AUC) to calculate the probability that a model will rank a randomly chosen positive instance higher than a randomly chosen negative instance.

Using these measures we can evaluatewhich models performs best on the test dataset to come up with a final model to be used as the prediction model.

**Prediction**

This is the final model we pick based on the evaluation of different models that we build. For any new dataset this model will predict the class based on the inputs provided.

**Pseudo Code**

Let’s summarize the method with use of pseudo code as below:

1. Download all datasets available from the website http://iot.ee.surrey.ac.uk:8080/datasets.html
2. Using Spark, load and combine all the datasets
3. Using SQL and R, look at the descriptive statistics of the dataset. Create a correlation matrix between different numerical variables and create histograms, scatter plots and line graphs to better understand the data.
4. Label the dataset into different levels of congestion based on the average speed
5. Split the dataset into training and test data sets for cross validation and evaluation of the models to be created
6. Build different supervised learning models based on training datasets using different packages in R
   1. Decision Trees
   2. Support Vector Machines
   3. Random Forests
   4. Time Series Analysis (e.g., ARIMA)
7. Evaluate the models with different performance metrics to pick the best model that can be used for prediction
8. Select the final model