WEBCAMS, PREDICTIONS, AND WEATHER

By Manshant Singh Kohli

Problem

Collecting weather related data can be tedious and time-consuming process if it is needed to be recorded manually. Goal of this project is to reduce the human effort required to complete this task, by generating the weather details based on images taken from a webcam and some other sensor data like temperature and relative humidity.

Acquiring Data

Two data sets were used to answer this problem:

- Webcam images were provided by Kat Kam images. These images were taken every hour from downtown Vancouver.
- Sensor data and Weather observations data set from Vancouver Airport weather station was obtained. This data was collected by the Canadian government. This data contains weather observations as well as sensor information like temperature, relative humidity, wind speed and air pressure.

Cleaning Data

The following steps were taken to clean the data:

- The sensor data and weather observations data set was provided through a series of CSV files. All these CSVs were combined into a single pandas DataFrame. Any columns that were not needed or were missing values were removed. The result was a single DataFrame – df.
- The webcam images were provided as a series of jpg images. The image filenames contained the date and time of when these images were taken. A DataFrame was created whose columns included the filenames of these images and the data/time column that presented date and time in the same format contained in the ‘df’ DataFrame mentioned above.
- An inner join was performed on ‘df’ with the DataFrame that contained the image file names. This provided us the image file names that had matching weather observations and sensor data.

- The weather observations did not follow a fixed pattern for observation names. It included observations like “Mainly Clear”, “Clear”, “Rain”, “Rain Showers”, “Moderate Rain Showers” etc. These observations were categorized into 5 distinct groups: ‘clear’, ‘cloudy’, ‘rain’, ‘fog’ and ‘snow’. Some of the weather observations also contained more than one of these categories like “Rain,Fog” and therefore the new cleaned weather observations were also allowed to have multiple groups for an observation entry.
- These new cleaned data was then written to a CSV file to be used for machine learning later.

Data Transformation

The following steps were taken to transform the data:

- The image files were read and cached as a collection of pixels. The images had 192x256 pixels and each pixel and 3 values (RGB). These 3-dimensional arrays for each image were reshaped into a single dimensional array. These 1-dimensional arrays were then transformed using PCA to have shrink the number of columns for images
- These transformed image data was then joined with additional sensor data like temperature, hour of the day and relative humidity.
- MinMaxScaler transformation was applied to this data and cached to be used for sample input data for multiple machine learning techniques.
- The multiple grouped weather observation records were transformed using MultiLabelBinarizer to support output as weather observation that could support more than 1 observation groups.

Data Analysis

The transformed data (mentioned above) was passed to multiple machine learning models:

- KNeighbors
- MLPClassifier (Neural Nets) with default solver and activation functions
- DecisionTrees
- ExtraTreeClassifier
- ExtraTreesClassifier
- RandomForestClassifer

These above models were used as they supported MultiLabel outputs. Also, different parameters of these models were tried to see what results they yield.

Results

While using the default parameters the following scores were obtained:

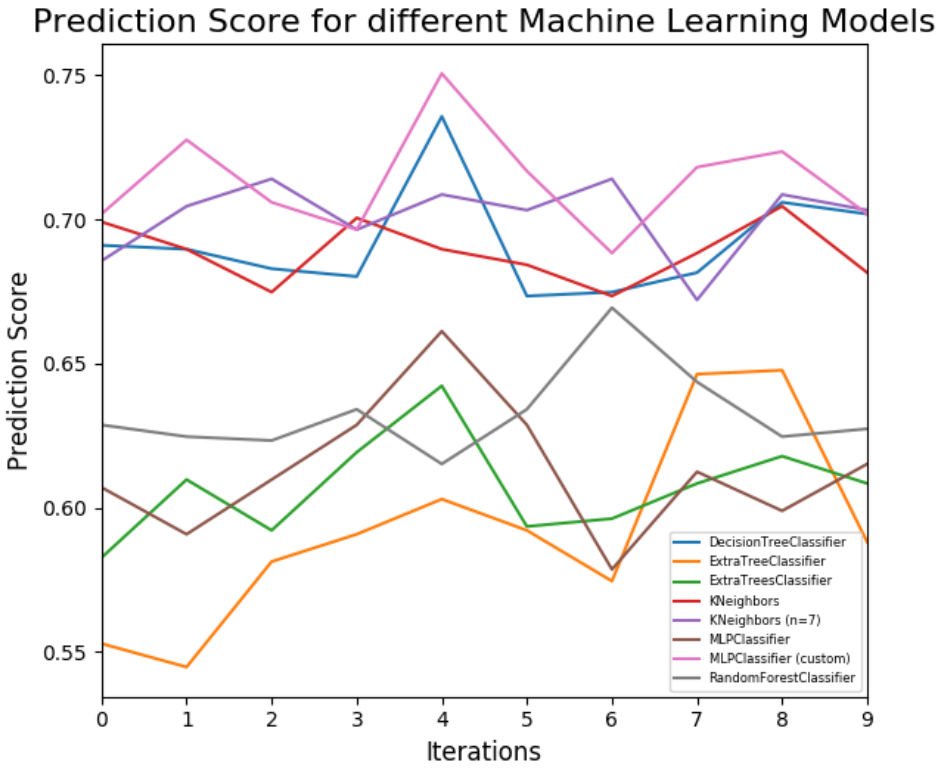
- KNeighbors: 0.682926829268
- MLPClassifier: 0.60433604336
- DecisionTreeClassifier score: 0.724932249322
- ExtraTreeClassifier: 0.579945799458
- ExtraTreesClassifier: 0.631436314363
- RandomForestClassifier: 0.636856368564

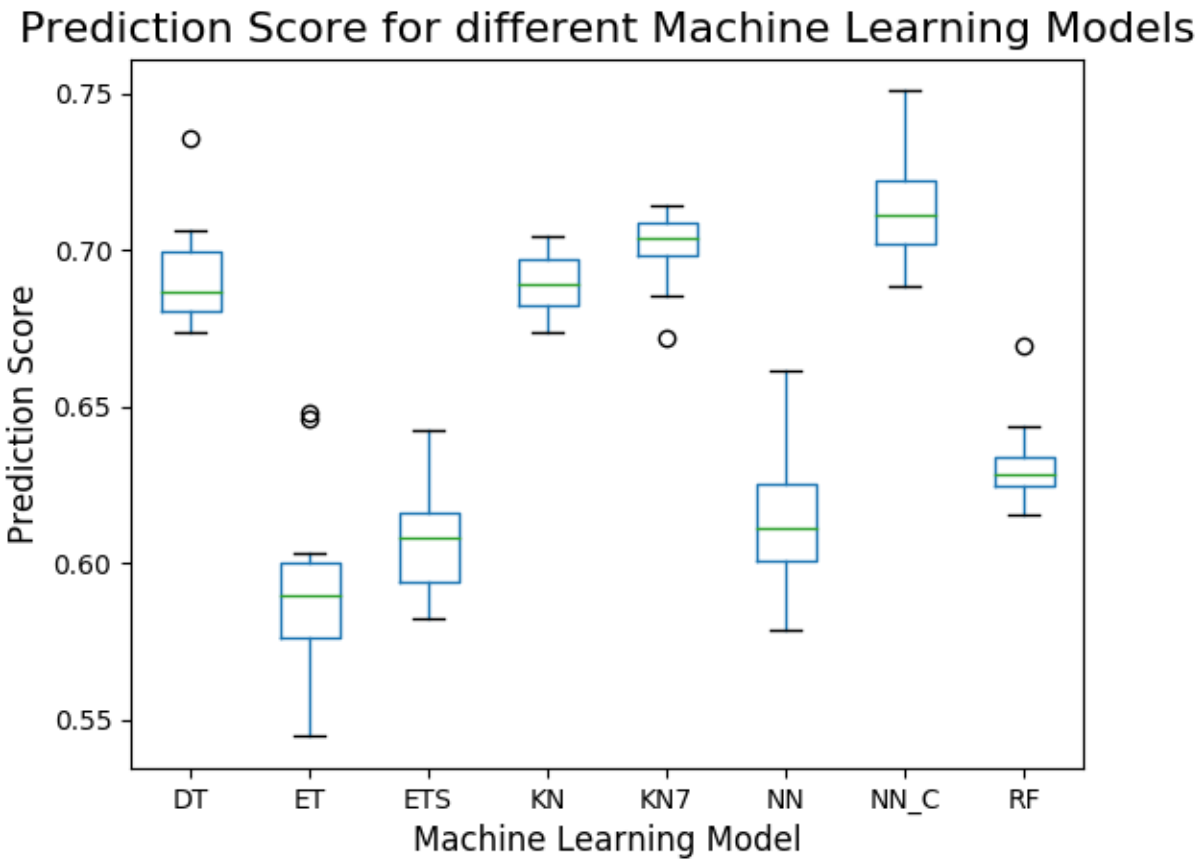
After tweaking the parameters of some of these models, better scores were obtained:

- KNeighbors (n=7): 0.70460704607
- MLPClassifier (custom): 0.7466124661246613

'MLPClassifier (Neural Nets) custom' used the lbfgs solver, logistic activation function and 64 hidden layers. It was noticed that these models yielded different scores (due to randomness in splitting training data and/or the model used) and the 'MLPClassifier (neural nets) custom' went as high as **0.7642276422764228**.

These models were run multiple times and their results are shown in the images below.





According to these results 'MLPClassifier (Neural Nets) custom' (also labeled 'NN_C' in Figure 2) appears to be yield the best overall results.

Limitations

- The observations were categorized into 5 distinct groups. This means that the accuracy of the models is checked after converting the observations to our groups. Therefore, the models don't predict the weather in the original weather observation labels.
- It takes time to try different parameters of the models to see which combination yields the best results. If given more time, better refined combinations of parameters for each model could be found.
- Currently there are 57 input columns (50 for image data after PCA transformation + 7 sensor data inputs). Maybe there is a way to weigh the columns for all the machine models.