**Assignment 10**

Wanyi Su

Student# 301445656

1. What is your best guess for the slope and intercept of the streaming points being produced?

When stream.awaitTermination = 600 and xy=100 (get 100 messages per second), slope tends to converge to around 6.1125, and intercept tends to converge to value around -9.64 to -9.72.

1. Is your streaming program's estimate of the slope and intercept getting better as the program runs? (That is: is the program aggregating all of the data from the start of time, or only those that have arrived since the last output?)

Yes, the streaming program's estimate of the slope and intercept getting better as the program runs. We use Spark DataFrame-based structured streaming in this program. As the input streams in, the unbounded table grows in real time. Statistically, more data means more information about the whole dataset, which contributes to more accurate results.

1. In the colour classification question, what were your validation scores for the RGB and LAB pipelines?

Validation scores:

For RGB pipeline: around 0.60 to 0.69

For LAB pipeline: around 0.75

1. When predicting the tmax values, did you over-fit the training data (and for which training/validation sets)?

According to the results on three tmax datasets, there’s no overfitting.

(W/o “yesterday\_tmax” feature)

**tmax-1 dataset:**

*(on training dataset:*

R2 = 0.9048928630294217

Rmse = 3.7702558891922373

*(on validation dataset:*

R2 = 0.8849420484670949

Rmse = 4.2743340031127195

**tmax-2 dataset:**

*(on training dataset:*

R2 = 0.8383962166525962

Rmse = 5.185701175581239

*(on validation dataset:*

R2 = 0.8368316730925512

Rmse = 5.234652607296786

**tmax-3 dataset:**

*(on training dataset:*

R2 = 0.8342987132246416

rmse = 5.265049671599341

*(on validation dataset:*

R2 = 0.8332683935570384

rmse = 5.287582349017721

1. What were your testing scores for your model with and without the “yesterday's temperature” feature?

(on tmax-2 dataset)

w/o the feature:

r2 = 0.8117343692667439

rmse = 5.628047226593437

With the feature:

r2 = 0.9128955950216859

rmse = 3.8156841486043827

1. If you're using a tree-based model, you'll find a .featureImportances property that describes the relative importance of each feature (code commented out in weather\_test.py; if not, skip this question). Have a look with and without the “yesterday's temperature” feature: do the results make sense and suggest that your model is making decisions reasonably? With “yesterday's temperature”, is it just predicting “same as yesterday”?

The results make sense.

With the “yesterday’s temperature” feature added, which has an importance score of about 0.8622, RMSE score has makes a lower value. And the importance scores imply that the new prediction model depends largely on yesterday’s temperature.

Without the “yesterday’s temperature” feature, the former two features have little larger weights in predicting tomorrow’s temperature, while the last two features have relatively lower weights in prediction.

With “yesterday’s temperature”, since it has a high importance score of 0.8622 compared to other features’ scores, the model prediction value depends heavily on “yesterday's temperature” feature. Therefore, it’s predicting almost “same” as yesterday’s (which means prediction model value varies mainly upon this “yesterday’s temperature” feature). (we got prediction result of 9.336 here with tmax-2 dataset)

With the feature:

(5,[0,1,2,3,4],[0.048708440512891346,0.04253587766145797,0.02763139525418963,0.018877473698435378,0.8622468128730258])

w/o the feature:

(4,[0,1,2,3],[0.39572324174699286,0.35709584614432616,0.12662642182746855,0.12055449028121237])