MIDPOINT REVIEW

--- YOUTUBE SUBSCRIBER PREDICTION

Yiwei Zhang

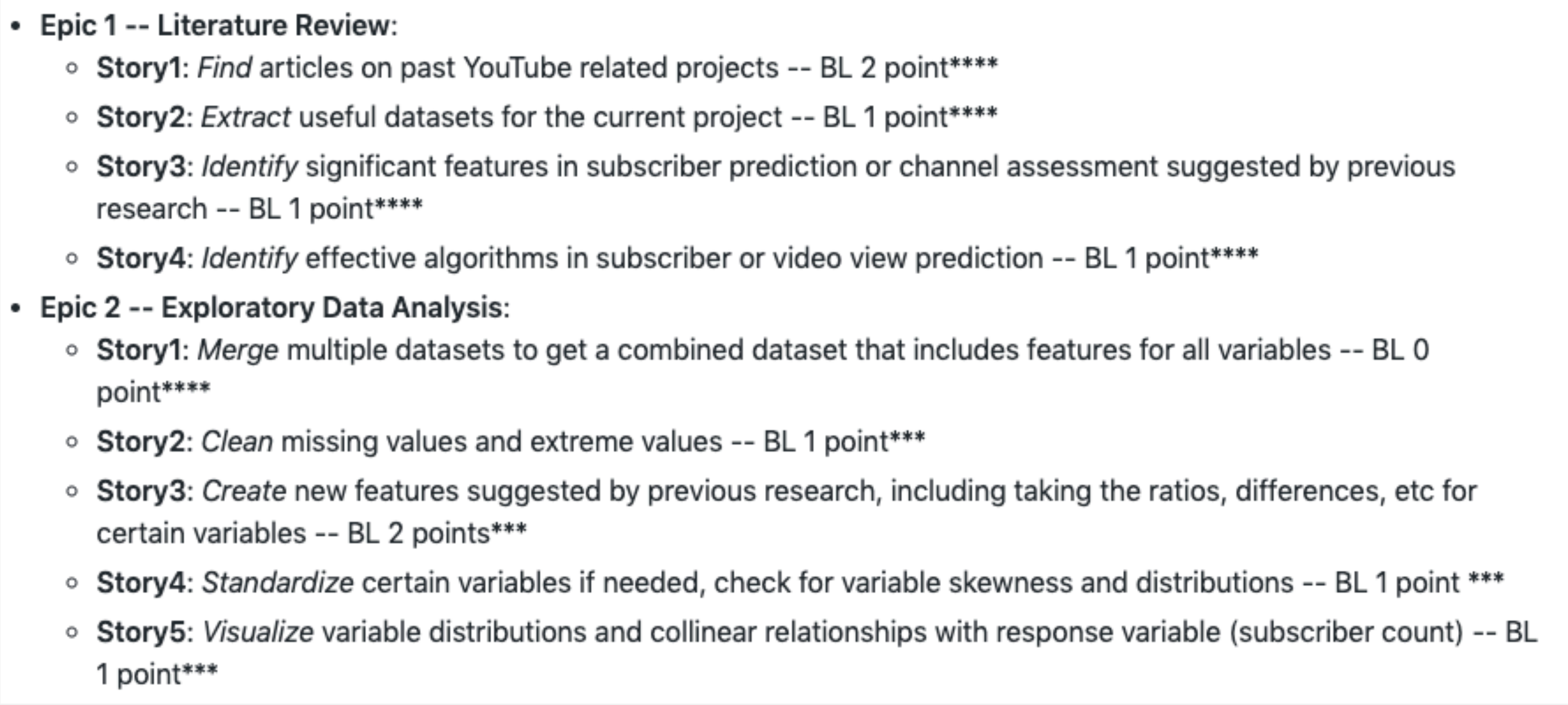
05/14/2019

HIGHLIGHTS

- Finished data cleaning and exploratory analysis. Created new features based on exploratory literature review on YouTube subscriber predictions
- Finished feature engineering and variable selection. Created a Gradient Boosting model to select important features to use for the distance matrix calculation for the KNN process later.
- Built the first draft of models with results met the success criterion. Created 4 models to predict the subscriber amount of a YouTube channel in the next 3, 5, 6.5 and 8 years. Results of all the models met the success criteria.

PROGRESS REVIEW

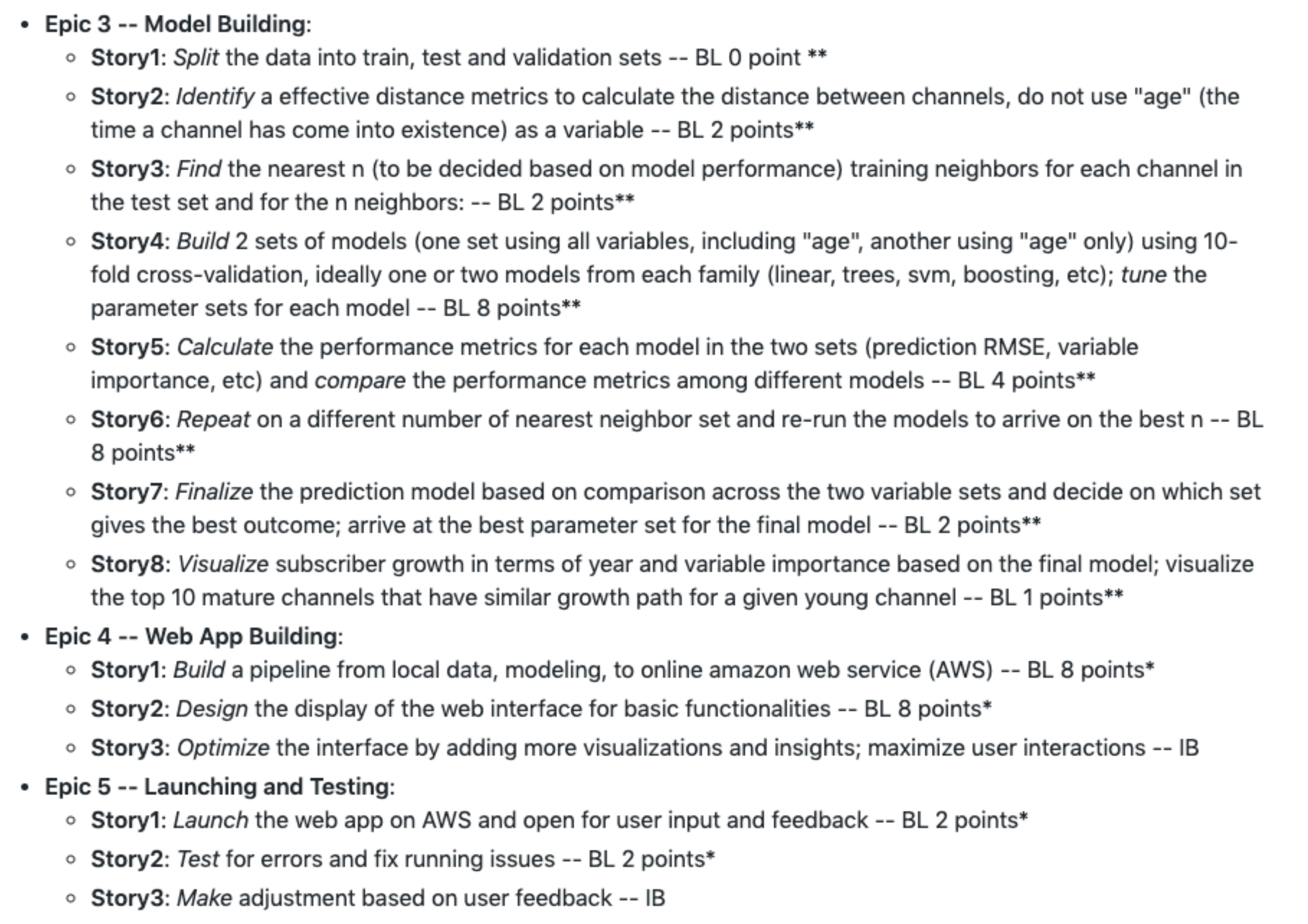
- Finished Story 1-4 in Epic 1: Literature Review
- Finished Story 1-5 in Epic 2: Exploratory Data Analysis



- **Epic 1 -- Literature Review**:
  - **Story1**: *Find* articles on past YouTube related projects -- BL 2 point****
  - **Story2**: *Extract* useful datasets for the current project -- BL 1 point****
  - **Story3**: *Identify* significant features in subscriber prediction or channel assessment suggested by previous research -- BL 1 point****
  - **Story4**: *Identify* effective algorithms in subscriber or video view prediction -- BL 1 point****
- **Epic 2 -- Exploratory Data Analysis**:
  - **Story1**: *Merge* multiple datasets to get a combined dataset that includes features for all variables -- BL 0 point****
  - **Story2**: *Clean* missing values and extreme values -- BL 1 point***
  - **Story3**: *Create* new features suggested by previous research, including taking the ratios, differences, etc for certain variables -- BL 2 points***
  - **Story4**: *Standardize* certain variables if needed, check for variable skewness and distributions -- BL 1 point ***
  - **Story5**: *Visualize* variable distributions and collinear relationships with response variable (subscriber count) -- BL 1 point***

PROGRESS REVIEW

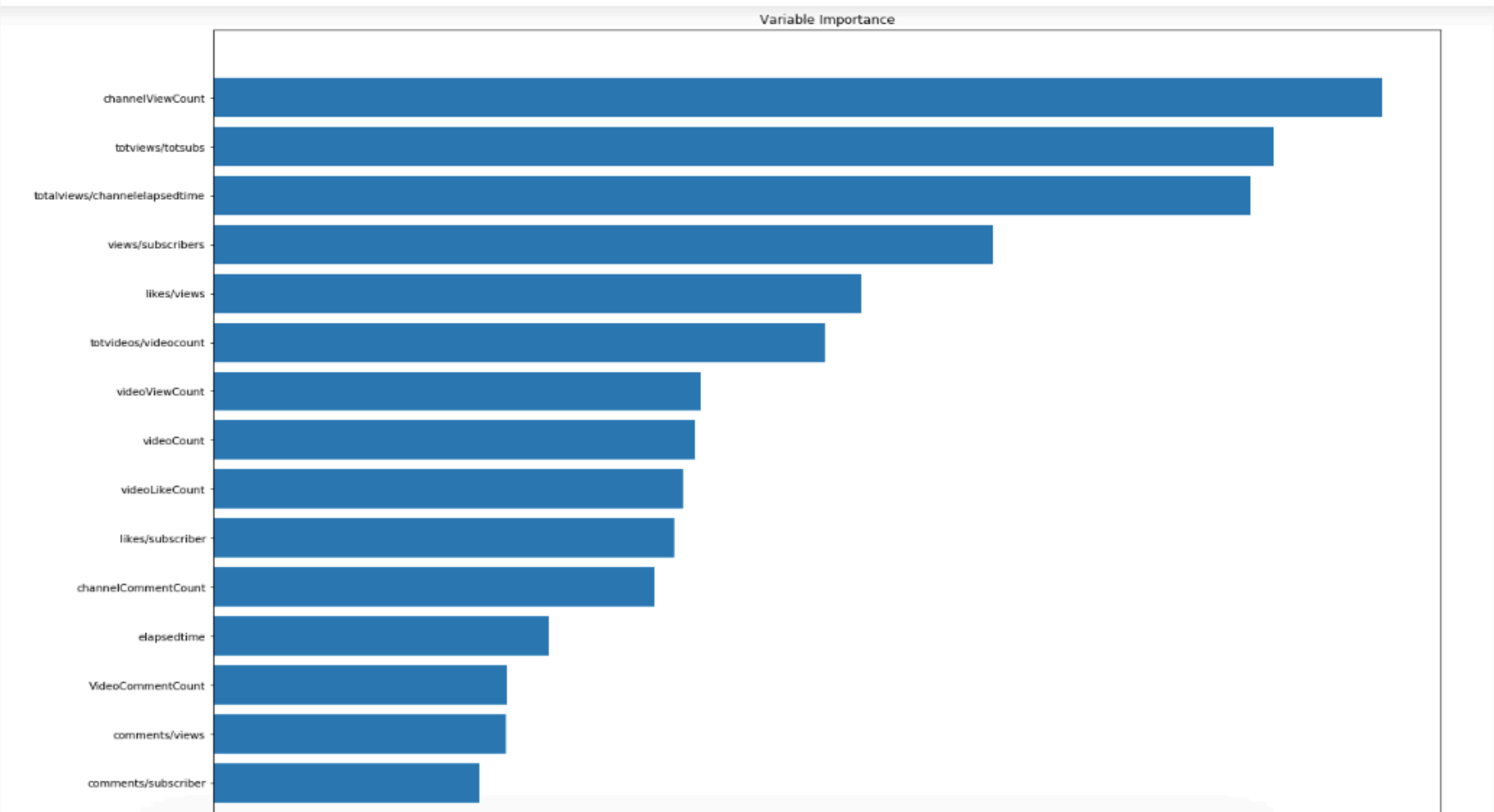
- Finished Story 1-7 in Epic 3: Model Building
- Finished Story 1 in Epic 4: Web App Building



- **Epic 3 -- Model Building**:
  - **Story1**: *Split* the data into train, test and validation sets -- BL 0 point **
  - **Story2**: *Identify* a effective distance metrics to calculate the distance between channels, do not use "age" (the time a channel has come into existence) as a variable -- BL 2 points**
  - **Story3**: *Find* the nearest n (to be decided based on model performance) training neighbors for each channel in the test set and for the n neighbors: -- BL 2 points**
  - **Story4**: *Build* 2 sets of models (one set using all variables, including "age", another using "age" only) using 10-fold cross-validation, ideally one or two models from each family (linear, trees, svm, boosting, etc); *tune* the parameter sets for each model -- BL 8 points**
  - **Story5**: *Calculate* the performance metrics for each model in the two sets (prediction RMSE, variable importance, etc) and *compare* the performance metrics among different models -- BL 4 points**
  - **Story6**: *Repeat* on a different number of nearest neighbor set and re-run the models to arrive on the best n -- BL 8 points**
  - **Story7**: *Finalize* the prediction model based on comparison across the two variable sets and decide on which set gives the best outcome; arrive at the best parameter set for the final model -- BL 2 points**
  - **Story8**: *Visualize* subscriber growth in terms of year and variable importance based on the final model; visualize the top 10 mature channels that have similar growth path for a given young channel -- BL 1 points**
- **Epic 4 -- Web App Building**:
  - **Story1**: *Build* a pipeline from local data, modeling, to online amazon web service (AWS) -- BL 8 points*
  - **Story2**: *Design* the display of the web interface for basic functionalities -- BL 8 points*
  - **Story3**: *Optimize* the interface by adding more visualizations and insights; maximize user interactions -- IB
- **Epic 5 -- Launching and Testing**:
  - **Story1**: *Launch* the web app on AWS and open for user input and feedback -- BL 2 points*
  - **Story2**: *Test* for errors and fix running issues -- BL 2 points*
  - **Story3**: *Make* adjustment based on user feedback -- IB

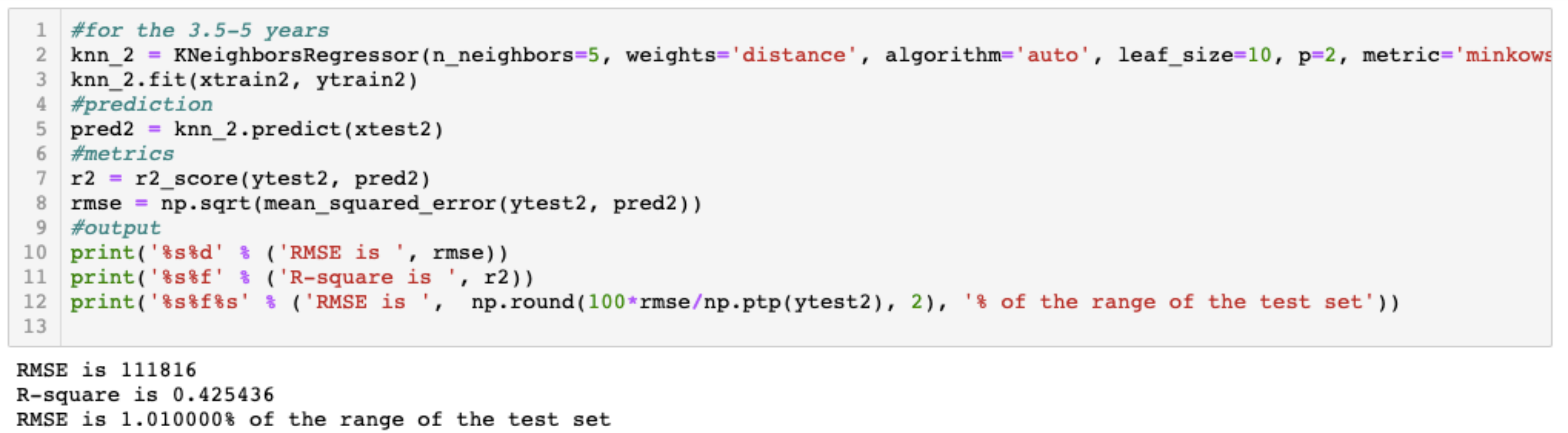
DEMO

- Important features selected by the Gradient Boosting Model. Variables were selected based on this result to fit the prediction model



DEMO

- Code snippet, as well as the prediction, r-squared, and root mean square error (rmse) for my prediction for the subscribers of YouTube Channels in the next 5 years. The RMSE met the success criteria in the project proposal. Another 3 models were fit to predict the subscribers in different time frames in the future, but only this one is shown here to reduce redundancy.



```
#for the 3.5-5 years
knn_2 = KNeighborsRegressor(n_neighbors=5, weights='distance', algorithm='auto', leaf_size=10, p=2, metric='minkows
knn_2.fit(xtrain2, ytrain2)
#prediction
pred2 = knn_2.predict(xtest2)
#metrics
r2 = r2_score(ytest2, pred2)
rmse = np.sqrt(mean_squared_error(ytest2, pred2))
#output
print('%s%d' % ('RMSE is ', rmse))
print('%s%f' % ('R-square is ', r2))
print('%s%f%s' % ('RMSE is ',  np.round(100*rmse/np.ptp(ytest2), 2), '% of the range of the test set'))

```

```
RMSE is 111816
R-square is 0.425436
RMSE is 1.010000% of the range of the test set
```

LESSONS LEARNED

- I learned how to build a pipeline to connect online data source (S3) to scripts of models, and to other online databases (RDS). This is essential for the structure of any web app and makes big data computation possible (since regular data source cannot host large datasets). I really appreciate this learning process and learning the concept while doing the project definitely deepened my understanding of how it works.
- Also learned about how to use API to acquire data and populate to online databases for internal use. Although I did not use API for my project, learning about its use was still informative for future projects.

RECOMMENDATIONS

- In the following sprint, I aim to visualize the result of the model (story 8, epic 3) for web interface and populate data to the RDS database.
- I also plan to start to work on and optimize the design of the web interface (story 2, epic 4).
- By the end of the next spring, the first draft of the web app should be done and ready for user testing and optimizations.