# Project 4

Heather, Jazmin, Amy, Khalia

## Final Grade: 92/100 Well done!

**Data Model Implementation (25 points)**

* A Python script initializes, trains, and evaluates a model (10 points)

10/10 Well done. I’ve never seen/used countplot before. Neat.

One thing you should probably include with your project is a requirements.txt file so that it is easy for someone to install the required libraries.

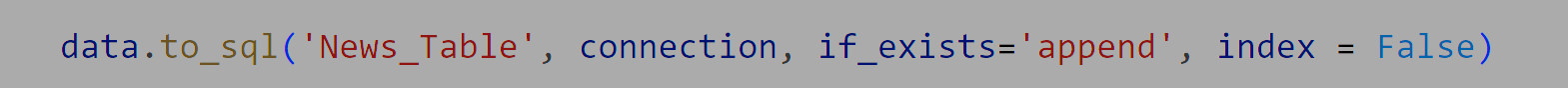
OK. So I tried to run some of the code. That was when I realized that the database and CSV files were considered Large files by GitHub, in which case you need to use git lfs to upload them. It would be useful to learn how to use git lfs because it is a PITA, as is most of git.

Good use of tqdm too. I’m learning a lot grading this project.

The one thing I would like to see is the word frequency bar chart with the stopwords removed. The word ‘the’ doesn’t need to be there.

* The data is cleaned, normalized, and standardized prior to modeling (5 points)  
  5/5 This is well done in the ETL notebook.   
  This code:  
  A screen shot of a computer code

  Description automatically generated

Could have been one line. Pandas already knows how to write to SQLite.  


Pandas has a to\_sql method that should work well with SQLite.

* The model utilizes data retrieved from SQL or Spark (5 points)

5/5 Yes it does.

* The model demonstrates meaningful predictive power at least 75% classification accuracy or 0.80 R-squared. (5 points)

5/5 Your model seems to be nearly perfect in differentiating between fake news and real news. Perhaps too perfect?

**Data Model Optimization (25 points)**

It’s a tiny thing, but the convention is to use capitals for X, and lowercase for y.

x\_train, x\_test

should be

X\_train, X\_test

Just an observation. This does not impact the grade.

* The model optimization and evaluation process showing iterative changes made to the model and the resulting changes in model performance is documented in either a CSV/Excel table or in the Python script itself (15 points)

8/15

Your analysis and charts are very good. However, I don’t see any iterations in your Fake\_News\_Detection\_using\_Machine\_Learning.ipynb or your Fake\_News\_Optimized.ipynb.

I needed to see some method of selecting Feature Importance (i.e. words and word patterns (sentences) that matter). Or a method of optimizing the Hyperparameters. Such as scikit-learn’s GridSearchCV function.

Your analysis earned you 8 points, the missing optimization strategy cost you 7.

* Overall model performance is printed or displayed at the end of the script (10 points)

10/10 In this chart, you could use better X and Y labels. Instead of ‘True label’, which is confusing, you could say “Article is Actually Fake News” vs. “Article Predicted to be Fake News”.

**GitHub Documentation (25 points)**

* GitHub repository is free of unnecessary files and folders and has an appropriate .gitignore in use (10 points)

10/10 It is and does.

* The README is customized as a polished presentation of the content of the project (15 points)

14/15 Your readme is great. Just the right amount of technical and explanation. It was engaging and covered all the bases. I think your conclusion is a little weak. Your model and analysis seem to show that you can predict, with a high degree of accuracy, whether an article is real or fake, but in your conclusion you don’t say that. Instead, being a lot more reserved and throwing your analysis into doubt.

**Group Presentation (25 points)**

* All group members speak during the presentation. (5 points)

5/5 Well done.

* Content, transitions, and conclusions flow smoothly within any time restrictions. (5 points)

5/5 Excellent presentation.

* The content is relevant to the project. (10 points)

10/10 I learned that fake news can be detected nearly all the time with the provided source data and analysis technique.

* The presentation maintains audience interest. (5 points)

5/5 It did indeed. Thanks.