MLDS 422 – Fall 2024
Final Python Project
Due: Wednesday, 12/4/24 at 11:59pm

Machine Learning Workflow Practice – 40 points:

Final Submission [3 points]

Your final submission on Canvas should be a link to a private Github repo with the following:

- **README file**: Clearly state the purpose of the project and what's in your repo
- **Code**: Juypter Notebook(s) / Python script(s) containing all your Python code
- **Data**: .csv file or .pkl file containing the data that your code needs to run
- **Requirements**: requirements.txt or .yml file with your environment details
- **Ignore**: .gitignore file (if necessary) to exclude any unwanted files, like .DS_Store, .ipynb_checkpoints, etc. from your repo
- **Permissions**: Give ***adashofdata*** and ***ja-nguyen*** access to your repo for grading

Tips:

- To receive full credit, please make sure your repo and code are ***clear***, ***well-documented*** and ***well-organized***
- If you're using notebooks, make sure all your cells are run so others can see the outputs in the browser
- Remember to remove your Postgres login credentials from your code on Github
- After the project is graded, you're welcome to make the repo public, if you want to use it as a portfolio project

1. Scope Project & Gather Data [1 point]

Goal: Create a model that will predict movie critic ratings

Data

- The movie data set is located within the NU MLDS 2024 server, in the everything2024 database, within the mlds422 schema
- Connect to the database using psycopg within Python to access the data

2. Data Cleaning & Exploratory Data Analysis [14 points]

Write Python code to do the following with the movies data:

1. Find the number of movies that were released in theatres each year.

- There are some erroneous values in the data
- Visualize the number of movies each year and describe what you see
- Explain what assumptions and fixes you plan to make in the data as you move forward with your analysis

2. Looking only at movies that were released in theatres before 2010:
   - Find the 5 highest rated movies by critics (critic_rating)
   - Find the 5 highest rated movies by the general audience (audience_rating)
   - Take a look at the movie titles – these top movies don't seem very popular

3. Create a new DataFrame containing only "popular" movies that were released in theatres before 2010.
   - Assume popular movies are those with more audience reviews than the average number of audience reviews of all movies before 2010
   - Find the 5 highest rated movies by critics (critic_rating)
   - Find the 5 highest rated movies by the general audience (audience_rating)
   - Take a look at the movie titles – these top movies should make more sense

4. Using your new DataFrame (popular movies released in theatres before 2010), answer the following questions about the *rating* column:
   - What percent of movies fall under each type of rating (R, PG-13, etc.)? What are your takeaways from the summary table?
   - Create a visualization that shows for each rating (R, PG-13, etc.), the average critic rating and the average audience rating. What are your takeaways from the visualization?

5. Create a pair plot of the new DataFrame. What are 3 insights you can take away from looking at the pair plot?

6. Using either pandas or data visualizations, find 3 more insights using any columns in the new DataFrame.

3. Feature Engineering [8 points]

Your goal is to create a predictive model that will predict the critic_rating. Using the full data set that you originally read into Python (with the erroneous years fixed):

7. Split the data into a training and test set, with the training data including movies released in theatres before 2010 and the test data including movies released in theatres in 2010 and after.

8. If your goal is to predict the critic_rating before the first critic or audience rating gets posted for a movie, which columns in the data should you NOT use to

create features? Update your training and test data sets to NOT include these columns.

9. Using only the training data, create a new DataFrame containing the following ID column and features:
    - movie_title
    - runtime_in_minutes
    - NEW: kid_friendly (1 if G or PG, 0 if other ratings)
    - NEW: dummy variable columns for each genre

10. Create 3 new features that you think will do a good job predicting the critic_rating. Each new feature should use various combinations of the columns from your training data.

4. Modeling [14 points]

11. Make sure you apply the same transformations on your X_test and y_test data sets that you applied on the X_train and y_train data sets.

12. Make sure that your X_train, y_train, X_test and y_test data sets only contain columns of numeric and non-null values. Explain and justify how you decide to deal with data issues.

13. Fit 3 linear regression models on the training data:
    - Model 1: Use only runtime_in_minutes
    - Model 2: Use runtime_in_minutes and kid_friendly
    - Model 3: Use runtime_in_minutes, kid_friendly and the dummy columns for the genres

14. Score the linear regression models on the test data by writing a function where you can input the y_test and y_pred values (y_pred = predicted values after you apply the fitted model to your X_test data), and it outputs the following metrics: R^2, MAE and RMSE. Apply the function to the three models that you've fit so far.

15. Which model performs the best so far? Which features seem to do a good job predicting the critic rating (hint: you can check p-values using statsmodels)?

16. Try fitting 3 more linear regression models on your own using a combination of the columns so far (runtime_in_minutes, kid-friendly and the dummy columns for genre) and your newly engineered features.

    - Each subsequent model should attempt to do a better job at prediction than the previous model (even if the metrics don't end up being better, your choices should make sense)

- With each new model, explain why you are making your decisions of which feature(s) to include / remove

17. Out of the 6 models you created, which model performs the best? Which features seem to do a good job predicting the critic rating?

18. List 3 other things you could to do at this point to try and improve your model.

Homework Policies for MLDS 422

Submissions

- Submit assignments through Canvas
- The output for each problem should be displayed in your submission so the TA doesn't have to run your code
- Code should run without errors
- Code should include comments and documentation

Grading

- ~60% of points for correctness (code runs and outputs correct results)
- ~30% of points for quality (well-structured and incorporates best practices)
- ~10% of points for documentation (commented and well-explained)
- Late policy: 10% of points deducted for each late day
- Keep in mind that there may be multiple correct approaches to each problem. As long as you explain your reasoning and your logic makes sense, you will receive full credit.

Resources

- Students should work individually on their assignments
- Students are encouraged to discuss questions with classmates, TAs and the instructor via the course Slack channel and office hours
- DO use ChatGPT to look up documentation and specific technical questions
- DON'T use ChatGPT to paste entire assignments without thought (remember that practice and struggle are essential to becoming a proficient coder)