Week 8 Meeting Notes

1. Call to Order

A meeting was held at 33 Charles St East on 28 October 2022.

2. Attendees

1. Regan
2. Riley
3. Jared

3. Open Issues for Week 8

1. Acceptance testing - consider the pros and cons of jenkins and github actions and why we choose Github actions over the other.
2. Rationale for machine learning model used in the project
3. Functional requirements (what inputs are required, and what format)

1. Acceptance testing
   During the discussion, Jared led the discussion on why we should choose to use Jenkins over Github actions for the purpose of creating workflows that automatically build, test, publish, release and deploy code. This was important because we will save time by automating our future deployments and reduce human errors when deploying manually. We debated between the pros and cons of each tool, and came to a conclusion of Github actions based on the following reasons:
   1. Plugins in Jenkins have to be kept up to date.
   2. A single Jenkins server build is costing money even if I don't run any builds.
   4. Not consistent on concurrent builds.
   5. Dependent on several plugins, which come with updates that we will need to deal with from time to time.
   6. On the other hand, GitHub actions has a tight integration with GitHub
   7. Higher security as we do not need to hand over access to the source code and sensitive information to a third-party provider

2. Rationale for machine learning model used in the project
   a. Decision Trees learn simple decision rules from the features and determine the best split by choosing the feature that allows for greater information gain or lower node impurity. However, if there is no stopping criteria, the tree will continue splitting until all children nodes are pure. This causes the tree to overfit the training set and it becomes more complex than necessary.
      - Due to overfitting, the tree was not able to perform well for the first 3 variations (baseline, outlier removal, feature selection). Only when a limit was placed to the depth of the tree during hyperparameter tuning, the tree was forced to terminate early and its performance improved significantly.

b. Why is Gradient Boosting the better tree-based model in theory
    i. The tree-based models used are Decision Tree, Random Forest, and Gradient Boosting.
    ii. Random Forest is an ensemble learning technique - it constructs a set of base models from the training set and make predictions by aggregating the predictions made by each base model. For Random Forests, the base models are Decision Trees and each tree is generated using a random subset of training set and feature set. The forest then averages the predictions from all trees.
    iii. Random Forest is more accurate than a single Decision Tree as it incorporates more diversity. It is also more efficient as searching among a subset of the feature set when determining the best split is much faster too. However, it does not perform as well as Gradient Boosting.
    iv. Gradient Boosting uses boosting on top of ensemble learning. It boosts a set of weak learners to a strong learner such that misclassifications made by weak learners are made important. In general, the distribution of training set is changed adaptively so that weak learners will focus more on errors made by previous learners.
    v. For Gradient Boosting, Decision Trees are used as the weak learners and are added one at a time, and fitted to correct the errors made by previous trees. The errors made can be captured by a cost function, which is to be minimized using the gradient descent algorithm (hence the name gradient boosting). Thus, Gradient Boosting is the better tree-based model
c. Best model to predict gross revenue for Top IMDB Movies dataset
    - From a theoretical standpoint, Gradient Boosting should perform better than Decision Tree and Random Forest due to reasons made in earlier discussions. It should also perform better than the basic linear regressor. Indeed, from our results, Gradient Boosting performed the best among the 4 models for all variations (baseline, outliers removal, feature selection, hyperparameter tuning).
    - From our results, it is safe to say that Gradient Boosting is the best tree-based model to predict the gross revenue of top IMDB movies from this dataset. However, as there are other strong models that were not covered in the scope of this project (eg. Multi-Layer Perceptron, Support Vector Machine), we cannot be certain that Gradient Boosting is the best model in predicting gross revenue for top IMDB movies.

3. Functional requirements
   We discussed the number of features that users can input so as to output the desired results from our machine learning model. Regan has completed his feature importance testing using his machine learning algorithms and has narrowed down to 5 most important features which were ranked by their Mean Decrease in Impurity scores (MDI scores). The top 5 features which we have narrowed down are:
    1. Number of users who voted
    2. Budget

3. Number of critic reviews
4. Family genre
5. IMDB score

It was also important to know the format in which users input into our web application. For points 1,2,3,5 inputting a float or integer will be necessary. Our algorithms will automatically convert these 4 inputs into floats. With regards to 4, it will be a boolean of 0 and 1. 1 will represent the genre of ‘Family’.

After the values have been inputted, we have to ensure that the output gross revenue from our model will result in a plot that will be observed in the main plot on the web application