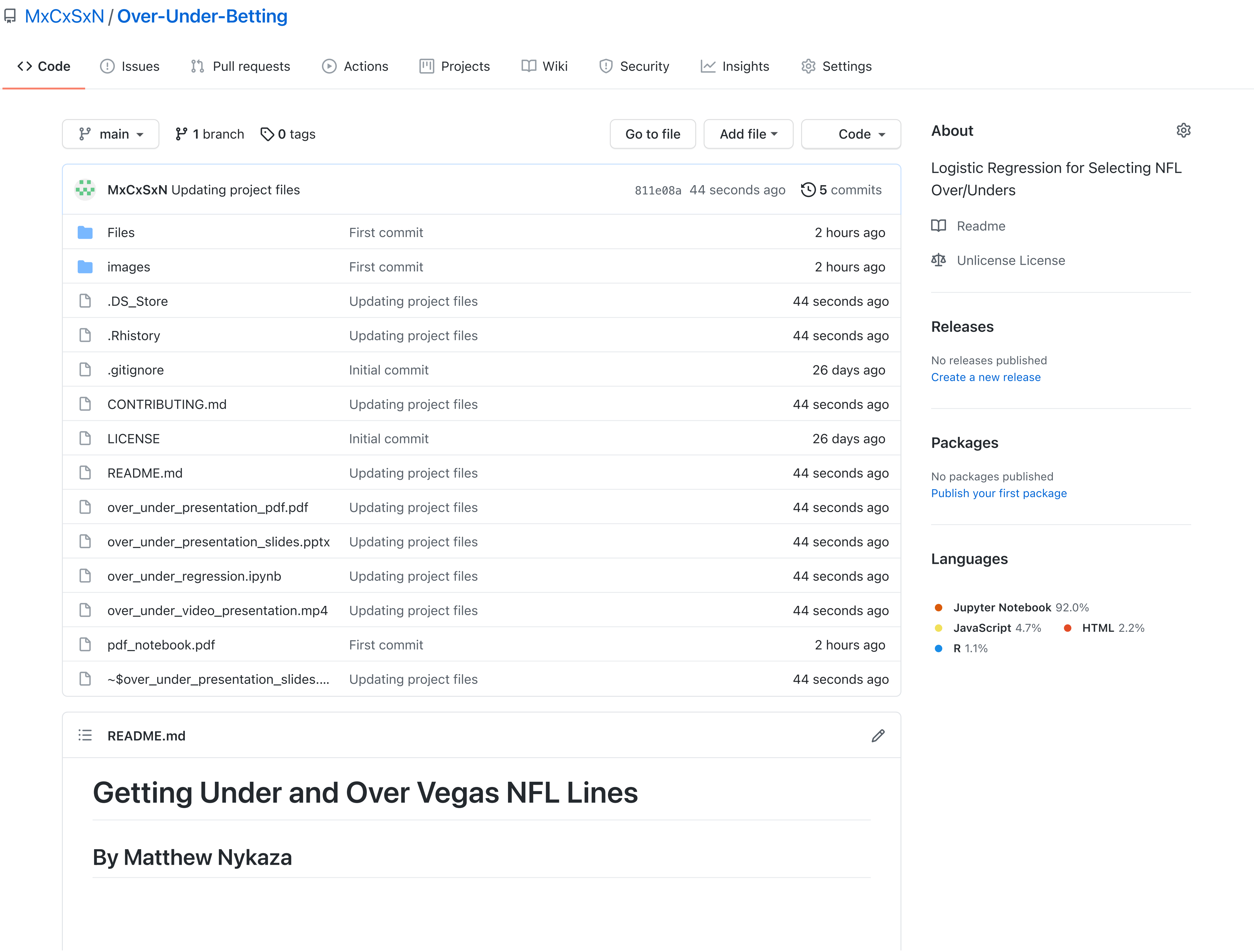

MxCxSxN / Over-Under-Betting

MxCxSxN Updating project files `811e08a` 44 seconds ago 5 commits

Files	First commit	2 hours ago
images	First commit	2 hours ago
.DS_Store	Updating project files	44 seconds ago
.Rhistory	Updating project files	44 seconds ago
.gitignore	Initial commit	26 days ago
CONTRIBUTING.md	Updating project files	44 seconds ago
LICENSE	Initial commit	26 days ago
README.md	Updating project files	44 seconds ago
over_under_presentation_pdf.pdf	Updating project files	44 seconds ago
over_under_presentation_slides.pptx	Updating project files	44 seconds ago
over_under_regression.ipynb	Updating project files	44 seconds ago
over_under_video_presentation.mp4	Updating project files	44 seconds ago
pdf_notebook.pdf	First commit	2 hours ago
~$over_under_presentation_slides....	Updating project files	44 seconds ago

README.md

Getting Under and Over Vegas NFL Lines

By Matthew Nykaza

About

Logistic Regression for Selecting NFL Over/Unders

Readme

Unlicense License

Releases

No releases published
Create a new release

Packages

No packages published
Publish your first package

Languages

- Jupyter Notebook 92.0%
- JavaScript 4.7%
- HTML 2.2%
- R 1.1%

Overview

For this project I sourced NFL betting, stadiums and teams data from the data website Kaggle. This data includes data going back to the 1978 NFL seasons through this last season. I am tasked with creating a classification model that will assist sports bettors with selecting the over or under in an NFL game. The Over/Under line is the predicted combined score between two teams in a game, to go Over the two teams must combine for greater than that total, and to go Under they must combine for less than that total. I began this project by preprocessing the data, a step that included taking out unnecesary variables, engineering relevant ones from the given data, and performing intelligent decisions as what to do with missing/incomplete data. Once the processing was done I begain to train and edit models using that data, and Sklearn's pipeline function. After some early attempts with base a Logistic Regression model, it was determined that using a tree-based model will likely be the best to get the best scores. The final model used was a Random Forest model, with some hyperparameter tuning, that allowed me to achieve roughly 53% accuracy on my test data. In the future I hope to acquire more information, and continue to tune the model to achieve optimal results.

Business Problem

Vegas has been making money hand over fist from the average person for generations. In today's Vegas they use advanced models that take every bit of information possible in order to create their betting lines. For the average bettor, beating Vegas can feel like an impossible task, but with this project my aim to to make that a reality for more people. The goal will be for anyone to input information about any NFL game and be able to get a accurate prediction over whether to take the Over or Under in that specific matchup.

Datasets

Data can be found in the `files` section of this GitHub repository.

For this project I had three datasets

- NFL betting information which included dates, week of schedule, teams, scores, betting lines, weather, stadium names, and playoff information
- NFL stadium data which had more detailed information about the various stadiums that NFL teams have played at since 1978
- NFL team data which included information about individual teams such as nicknames, conference and division information The main data that I used was te NFL betting, but I used the stadium information to dig into greater detail about individual stadiums, and I used the NFL teams data to compare conference/divisional matchups, as well as help setup the average scores for each individual team's last 5 games.

The main data that I used was te NFL betting, but I used the stadium information to dig into greater detail about individual stadiums, and I used the NFL teams data to compare conference/divisional matchups, as well as help setup the average scores for each individual team's last 5 games.

Analysis

For this project I needed to create the "target" variable, which was whether or not the game will be Over or Under. To do this I first preprocessed and cleaned all the data to make it more digestible for a model. This included steps such as handling NaN data, reviewing outliers, handling erroneous data, and engineering new features to assist with the future modeling.

Modeling

For the modeling process two main models were used, then improved with the help of Sklearn's pipline, GridSearchCV and other methodologies. The first model was a reletively basic Logistic Regression model that had roughly 51% Accuracy, ROC-AUC Score (basically the likelyhood of the model ranking a random positive example, in this case an over, correctly) of 52%, and an F1 score (a combination of Recall and accuracy) of 44%. While these results were note enough to be considered anywhere near accurate they were a good starting point. After much training, the final model selected was a Random Forest model with a good deal of hyperparameter tuning. The reasoning for selecting a Random Forest model was because there was a good deal of multicolinearity in the data, and this type of model would not be effected by that. In that final model we were able to improve the Accuracy score to 54%, the ROC-AUC to 53% and the F1 Score to 45% on the testing data. This still is not a model that can be implemented, and more work needed to be completed on the data.

Conclusion

- This model was able to perform the best out of all previous models, and I really think that finding RancomizerCV as a method of getting a good starting point with the hyperparameters

- All metrics rose by around .01 - .015 points, which may not seem like a lot, but this was the greatest increase in the data seen to date.
- Overall what this shows is the need to complete more data cleaning, and get more data.
- I believe that one major issue is that I am trying to beat Las Vegas, which creates these point spreads using models much more advanced and practiced than this, I do believe that with more time this could be a viable product
- It may be that I have too much past data, when they game was very different, this could be a hinderence as well

Further Work

- Need to get more information about each individual game, this could include more data mining and more feature engineering of feature variables
- More tests of Hyperparameters
    - Utilize graphing techniques to help determine some of these features
- Try boost models
    - They tend to be more powerful, and may be able to achieve better scores