Databowl Journal

Week of 2/10

Objectives:

Will

- Develop objectives and plan to carry out the project
- Create milestones for the semester
- Familiarize with the dataset and work on data cleaning
- Create some Github issues

Sam

- Set up Git Large File Storage
- Familiarize with the dataset

Tushar

- Set up Github page
- Familiarize with the dataset

Progress:

Will

- Created a document detailing our goals and plans for the semester
- Familiarized with the dataset
- Started working on data cleaning
- Created Github issues

Sam

- Set up repository
- Started EDA

Tushar

- Set up Github page
- Familiarize with the dataset

Week of 2/17

Objectives:

Will

- Examine Kaggle submissions to get a sense of how we can visualize the play-by-play data
- Do some more exploratory data analysis and work on feature engineering
- Try out some basic modelling techniques and see how they perform. Will work on some more later and set a more official baseline accuracy

Sam

- Feature engineering
- EDA

Tushar

- EDA
- Top 100 players and histogram of each player yards

Progress:

Will

- Most Kaggle submissions only included visualizations for their exploratory analysis, though some pictured the play-by-play data
- Potential of using heat map with contour lines to visualize defensive, offensive formations and rusher movement
- Worked with Sam to see if we can use player positions and orientations to create an accurate visualization of the play-by-play data

Sam

- Built out basic EDA
    - Investigating positional data

Tushar

- Did a lot of EDA and made some charts (including top 100 players histogram of yards).
- Basic feature engineering.
- Started writing a script to transform raw data into json as input for vis

Week of 2/24

Objectives:

Will

- Set baseline for our models using linear regression on our raw data
- Create presentation for full team meeting on Saturday 2/29

Sam

- Build basic play visualization
- Engineer any needed features for it

Tushar

- Build play visualization

Progress:

Will

- Set a baseline for our models using linear regression on our raw data
- Min-max accuracy considers the average between the minimum and the maximum yardage predictions
    - Baseline attained min-max accuracy score of approximately 0.35
- Correlation between the actual and predicted yardage values can be used as a form of accuracy measure (a higher correlation accuracy implies that the actual and predicted values have similar directional movement, i.e., when the actual values increase the predicted values also increase and vice-versa)
    - Baseline attained correlation of approximately 0.18
- AIC estimates out-of-sample prediction error and measures the relative quality of statistical models for a given set of data
    - Baseline attained AIC of approximately 147162
- Made slides and presented at full team meeting on Saturday 2/29 — it went well!

Sam

- Built play visualization in matplotlib

Tushar

- Worked on script to convert raw data into json for the vis - still having some issues between libraries
- Began building main vis in tkinter. Basic graphics that depict players moving dynamically in the window.

Week of 3/2

Objectives:

Will

- Improve our regression model using polynomial regression with interaction terms and do some basic feature engineering on the data

Sam

- Fix yard line in visualization
- Build basic models
- **Feature engineering**

Tushar

- Begin building physics engine in JavaScript

Progress:

Will

- Did some feature engineering to improve the workability of our model
- Ran backward selection using BIC criterion (similar to AIC, but prioritizes limiting the number of features used in the model)
    - Selected an 8-predictor model
        - **Player-level features:** *location, speed, acceleration, position played*
        - **Game-level features:** *yardline, side of field, yards till first down, number of defenders in the box*
- Added some appropriate polynomial and interaction terms to the features in our model
    - AIC value was lowered by over four orders of magnitude to approximately 19.41
    - Correlation between our actual and predicted yardage values improved to approximately 0.25
    - Min-max accuracy improved only a little
- Our model does a good job capturing the center of the data but fares poorly in attempting to identify big yardage gains or losses
    - Hypothesis: due to our assumption that yardage gain is approximately Gaussian in shape: in reality, the distribution of yards appears to have a higher peak and heavier tails compared to a Normal distribution

Sam

- Started FE notebook
- Fixed visualization--works 100% now

Tushar

- Began building physics engine shape (hitbox) classes in JavaScript
- Implemented collision detection between circles & circles, and rectangles & rectangles

Week of 3/9

Objectives:

Will

- Make our basic models more cohesive and clean up the code (was in a bit of a rush to get baseline model done in time for full team presentation)
- Look into machine learning packages to use in R
- Develop the framework for some of these more complicated regression models

Sam

- Continue feature engineering
- Understand how our data cleaning works

Tushar

- Continue building physics engine
- Circle rectangle collision detection

Progress:

Will

- Cleaned up the code for our basic linear and polynomial regression models
- Found the H2O package in R — offers parallelized implementations of many supervised and unsupervised machine learning algorithms such as GLMs, GBMs, Random Forests, Deep Neural Networks, etc.
- Started experimenting with how to develop some of these models

Sam

- Poked through the data cleaning script
- figured out how our positional data works

Tushar

- Implemented collision detection between circles & rectangles
- Built physics engine body classes.

Week of 3/16: Corona Boogaloo

Objectives:

Will

- coronavirus :(
- Try to plan rest of semester
- Choose which machine learning models to go ahead with

Sam

- Figure out goals & timeline for rest of semester

Tushar

- Goals, timeline, etc.
- Continue building physics engine
- Fix circle rectangle collision detection

Progress:

Will

- Updated our goals for the rest of the semester

    - Try to accomplish a visualization of plays and develop models with very good prediction accuracies
- Will go ahead with random forest, gradient boosting machine, and neural network regression models
- I will work on implementing the RF and GBM regression models using the H2O package in R and Sam will work on the neural network regression models using the Keras library in Python

Sam

- Made plans for remainder of semester
- Learned about tensorflow, built some practice models

Tushar

- Goals, timeline, etc.
- Fixed issues with collision detection between circles & rectangles
- Implemented dynamic bodies with acceleration, force, and impulse methods and attributes

Week of 3/23

Objectives:

Will

- Implement random forest regression model using clean data

Sam

- Further feature engineering
- build initial neural net models

Tushar

- Continue building physics engine
- Intersection manifolds
- Simulator class

Progress:

Will

- Implemented random forest regression model with our clean data using the H2O package in R
    - Attained a correlation accuracy between actual and predicted yardage values of approximately 0.30
    - Attained a min-max accuracy of approx. 0.41 before tuning hyperparameters, but this decreased to approximately 0.35 afterwards (look into this)

    - Compared to polynomial regression model using the same data including the same features, min-max and correlation accuracies improved by ~5%
    - Evaluated the MAE (average difference between yards predicted and the actual yards) and RMSE (square root of avg of squared differences between actual and predicted yardage values) of the random forest regression model
    - Note: since errors are squared before being averaged, RMSE gives a relatively high weight to large errors → RMSE more useful when large errors are particularly undesirable
    - Attained a MAE of approximately 3.41 yards and an RMSE of approximately 5.49 yards
- Evaluated MAE and RMSE of polynomial regression model to get a comparison
    - Found the MAE decreased by approximately 0.08 yards and the RMSE decreased by approximately 0.33 yards using random forest regression
- Next week, want to learn more about the scoring metric used to evaluate submissions in actual Kaggle competition

Sam

- Engineered basic features; related data cleaning
- Built neural net regression

Tushar

- Wrapped collision detection into a function that returns the intersection manifold of any two shapes
- Implemented a World class to track and simulate bodies

Week of 3/30: Spring Break

Week of 4/6

Objectives:

Will

- Look into the scoring metric used to evaluate submissions in the actual Kaggle competition
- Try implementing a random forest regression model with a log-transformed response and see if the accuracy improves

Sam

- Advanced feature engineering
    - Include more players
    - Use position/velocity data

- Tune existing neural net
- Neural net diagnostics

Tushar

- Continue building physics engine
- Impulse resolution

Progress:

Will

- CRPS (cumulative ranked probability score) is the scoring metric used in the competition
  - CRPS is a forecast accuracy measure which takes values between 0 and 1 and compares the empirical cdf of our predictions with that of the actual data
  - Directly related to MAE (i.e., CRPS = MAE/number of predicted values = MAE/199)
- Implemented the random forest regression model with a log-transformed response
  - MAE improved to approximately 3.30 yards and RMSE got slightly worse to approximately 5.52 yards
  - This corresponds to a CRPS of approximately 0.017
  - For perspective, the top kaggle submission got a CRPS of approx. 0.012 — still progress to be made; hopefully, with some more advanced models and feature engineering we can improve our score

Sam

- Tried to tune neural net
- Built some diagnostics for neural net
  - It doesn't beat the baseline :(

Tushar

- Implemented a Euler integrator to update dynamic body attributes
- Implemented impulse resolution to resolve collisions between bodies.

Week of 4/13

Discussed databowl evaluation metric (CRPS), need to revisit model design because our models are built as regressors.

Objectives:

Will

- Implement gradient boosting machine model using our clean data

- Try also implementing a gbm model with a log-transformed response and assess accuracy

Sam

- Figured out CRPS
- Build neural net classifier to generate CDFs
- Calculate CRPS scores for new model

Tushar

- Continue building physics engine
- Fix impulse resolution sinking
- learn how football works

Progress:

Will

- Implemented gbm model with our clean data using the H2O package in R
  - Attained a MAE of approximately 3.35 yards and an RMSE of approximately 5.66 yards
  - Similar to random forest regression model using log-transformed response
- GBM model with log-transformed response did not improve accuracy
- Can work on tuning paramters, but problem likely in our features — work on feature engineering next week

Sam

- Rebuilt neural net in modeling_2

Tushar

- Implemented Baumgarate Stabilization to improve impulse resolution
- Implemented an accumulator-based method of updating the world
- Naive implementation of broadphase and narrowphase collision detection in the world

Week of 4/20

Objectives:

Will

- Feature engineering

Sam

- feature engineering on additional player data
- Fix model generating non-monotonic CDFs
  - generate PDFs instead?

Tushar

- Continue building physics engine
- Collision tracking

Progress:

Will

- Engineered some cool features
    - Overall offense and defense features based on the average associated yardages of each configuration of personnel/formation
- High values of F statistic with very low p-values
    - Useful features
- High values of T statistic with very low p-values
    - Factor levels well-established

Sam

- Implemented PDF-based neural net
    - implemented CRPS loss function for tensorflow

Tushar

- Wrapped impulse resolution into a Collision class to both detect and resolve collisions.
- Added collisions as a trackable feature in bodies for feature engineering.

Week of 4/27

Objectives:

Will

- Re-run models with engineered features and get final accuracies
- Comment code and push everything to Github

Sam

- Any additional feature engineering
- Implement cross-validation if possible
- Create final presentation of model performance

Tushar

- Integration of physics engine with vis
- Game logic by applying forces

Progress:

Will

- Finished models and got our final accuracies for direct yardage prediction models
- Commented code and pushed to Github

Sam

- Impelemented basic CNN model
- Generated diagnostics for old models

Tushar

- Integrated physics engine with dataset by creating a new PlayerBody class that extends dynamic bodies.
- Implemented game logic by applying forces in the world on players to dictate their dynamics.

Week of 5/4

Objectives:

Will

- Prepare slides for final presentation
- Final presentation :)

Sam

- Touch up CNN model
- Run training on CDS server?

Tushar

- Simulation throttling
- Presentation

Progress:

Will

- Final presentation went well!!
- Currently finalizing everything for the end of the semester

Sam

- Set up CDS server environment, accelerated training, testing, model development
- Tuned CNN model; really good CRPS scores
- Verified final results and prepared presentation

Tushar

- Implemented automatic throttling during high simulation load
- Prepared presentation slides