* Introduction
  + Sports analytics evolved tremendously in the past few years and coaches, analysts and players are trying to use new technology every year to try to squeeze out every bit of advantage they can. With the introduction of Next Gen Stats and a partnership with AWS the NFL and its teams are collecting more and more data. The NFL does this to augment the viewing experience with interesting statistics and the teams do it to improve their performance on and off the field. Unfortunately all this data is not made available to the public as it has tremendous value and existing research is mostly done by the teams who have an interest in keeping their findings secret. In this project we wanted to make an attempt on the Holy Grail of analytics in the NFL by creating a model that predicts the yardage gained on an upcoming play. Mainly we wanted to see how far publicly available data can take us and if we can make a model based on this data that performs reasonably better than a semi-random guess.
* Previous Solutions
  + Due to reasons mentioned in the introduction there are very few publicly documented attempts similar to ours. Both a [study](https://arxiv.org/pdf/1601.00574v1.pdf) made by students at UMass in 2016 and [another](http://cs230.stanford.edu/projects_winter_2020/reports/32263160.pdf) made at Stanford in 2020 concluded that the data they used was not enough to make a model that predicts yardage with an acceptable error. We tried to improve on these solutions by gathering data from multiple different sources and from a longer time period as discussed in the Dataset section of this documentation.
* Dataset
  + For our dataset, we used a publicly available play-by-play NFL dataset from Kaggle from the years 2009-2018. We also gathered publicly available passing statistics on Kaggle for each quarterback in this same time period. Finally, using data scraping techniques through the selenium and beautifulsoup packages, we collected publicly available rushing statistics for each running back, also only in this time period. After this and several other data processing/cleaning steps, we combined the data into one dataframe, and then chose the features which we deemed important to the possible prediction of the yards gained on any given play. This included game\_seconds\_remaining (from 3600 downwards), ydstogo (until a first down), down (the number, 1 to 4), shotgun (binary, yes or no), play\_type (run or pass), qb\_kneel or qb\_spike (binary, yes or no), yardline\_100 (where you are on the field), posteam\_score and defteam\_score (possessing and defensive), rushing average yards per game and per carry for that year (rushing only), and quarterback age, yards/game, QBR, and rating for that year (passing only). Next, we separated the passing plays from the rushing plays, so we could make separate models for these and hopefully get more accurate predictions. Finally, after shuffling the dataframe, for the classification problem, we grouped the yards gained target variable into four groups: 0 = less than 3 yards, 1 = between 3 and 6 yards, 2 = between 6 and 10 yards, 3 = more than 10 yards (traditionally, a first down).
* Proposed Method
  + In our first method, we initially only developed one deep learning model with several layers. However, through the necessary guidance, it was suggested that we first conduct baseline tests and then move on to more complex models to see if those would help the model in any way. In this portion, the first step for both regression and classification was to see what our mean squared error and accuracy would be if we simply guessed the mean yards gained and majority “yards gained” categorical variable, respectively. This would give us the bare bones baseline to which we would compare any other model used. From here, we would conduct baseline machine learning model tests with linear regression and multiple logistic regression. Finally, we would move on to our deep learning models, beginning with a very simple fully connected network, with only one layer. We would then use a more complex fully connected network, to be followed by a couple of networks with complex residual connections. In these models, we planned on manually tuning some parameters, such as batch size and learning rate, to see if there would be any meaningful impact on the efficacy of the model.
* Evaluation Method
  + To evaluate our neural networks, we first used the standard method of model.evaluate(X\_test, Y\_test), giving us a mean squared error and accuracy for our regression and classification models, respectively. To understand the results, we then generated regression plots and confusion matrices to understand where our model made its errors. For the final portion of the evaluation, we had to compare our results to the very baseline models, beginning with our guessing of the mean and majority and then to our simple machine learning models. If our model achieved a significantly better result than these baseline methods, then it would be reasonable to evaluate the significance of features using their respective SHAP values.
* Results and Discussion
  + We made a regression model that tried to predict exactly how many yards a play would get if called, and a classification model that just tried to make a rough guess on the length of the play. The results we obtained turned out to be very far from predictions that would be acceptable to use in an actual play calling scenario. All of the models used only did marginally better than just simply guessing the mean of the values in case of the regression model and guessing the majority “yards gained” category in case of the classification. We tried several model configurations and different sets of hyperparameters but they had close to zero influence on the results. From this we conclude that the amount of data we used is not enough to achieve a good result. It would be interesting to see how these models perform with the data NFL teams have access to which include the position and velocity of every player on the field. Despite making close to no progress regarding our predictions we consider this project to be a great learning experience where we got to play around with various concepts in deep learning while trying to implement, train and evaluate different models.