Team Members:
Anshul Sharma (as10950)
Srishti Patel (sp4917)

Course:
CS-GY 6053
Fall 2018

Instructor:
Rumi Chunara, PhD

PUBG Game Analysis and Prediction

Motivation: PUBG is an online multiplayer battle game where 100 players participate in game. First prize goes to the player with top rank and second prize to player with most kills. Since, rank prediction is an active problem so, we wanted to analyse and predict the number of kills by a player in game along with the distance walked by player.

Problem: As, kills is a very important feature in the game, we want to predict it given player's rank percentiles, winPoints, kills and other numerous factors. Also, we figured, distance travelled is very important in the game, which is why we also want to predict walk distance as the second target in our project.

Targets: The two important features we're predicting are:

- Target1- **Kills**: Kills is the number of kills made by a player in the game and it is an integer value. We selected it as the target variable because **second prize** in the game goes to player having most number of kills. We will be finding the top features having highest correlation with our target variable

- Target 2 - **walkDistance**: walkDistance is the distance walked by a player in the game. We selected it as our second target as distance traveled has a **huge impact** on the game as the region keeps getting smaller and in order to win the game, you probably have high distance travelled value.

Learning the Background: Background learning is an important step before starting the project. We will learn the background through the following ways:

- By Participating in the game ourselves and learning more about the game.
- Going through the online PUBG description.
- Discussing with people their strategies while playing the game and their average kills and walkDistance in the game according to their ranks.

Data Source: https://www.kaggle.com/c/pubg-finish-placement-prediction/data

Data Description:

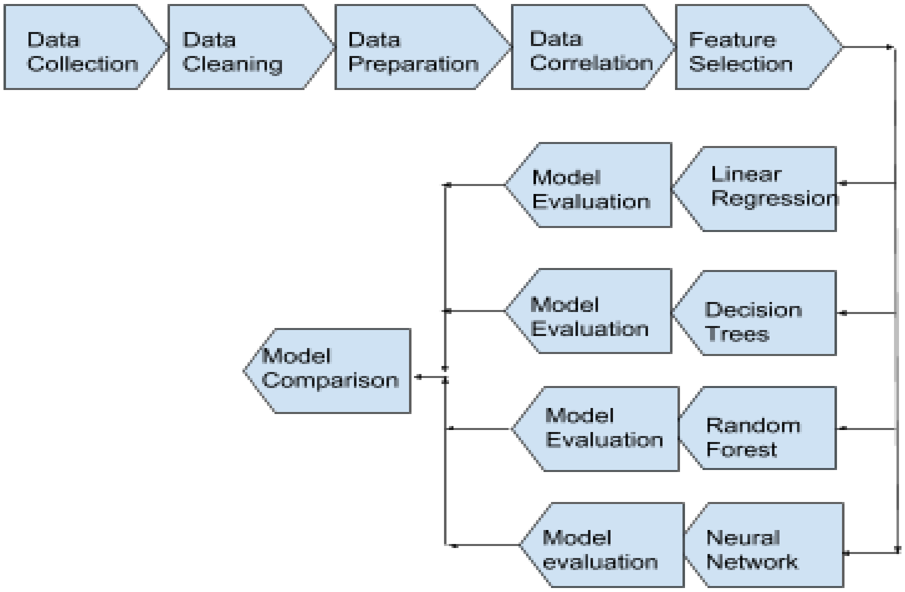
The PUBG dataset consists of 4.45m rows and 29 columns. Dataset if of following types dtypes: float64(6), int64(19), object(4). 'Id', 'groupId' and ' matchID' ar object types and will be droped, as they are of not much use in prediction. Match type is also an object and will be mapped to integer values from 0 to 15.

Data Type	Column	Type/Range
Object	Id,groupId,matchId,matchType	Categorical
int64	assists, boost, DBNOs', headshotKills, heals, killPlace, killPoints, kills, killStreaks, matchDuration, maxPlace, numGroups, rankPoints, revives, roadKills, swimDistance, teamKills, vehicleDestroys, weaponsAcquired, winPoints, winPlacePerc	1.Quantitative 2.Positive Integers(>0)
float64	damageDealt, longestKil, rideDistance, swimDistance, rideDistance, walkDistance	1.Quantitative 2.Positive float(>0)

Assumptions:

- No cheat codes were used by the players. (Apart from the ones we could identify and remove).

Analysis Approach:

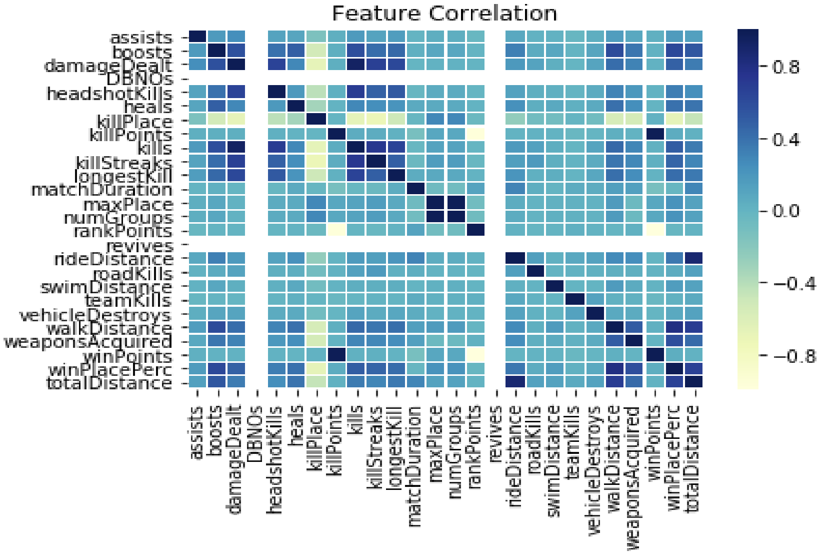


Steps Followed:

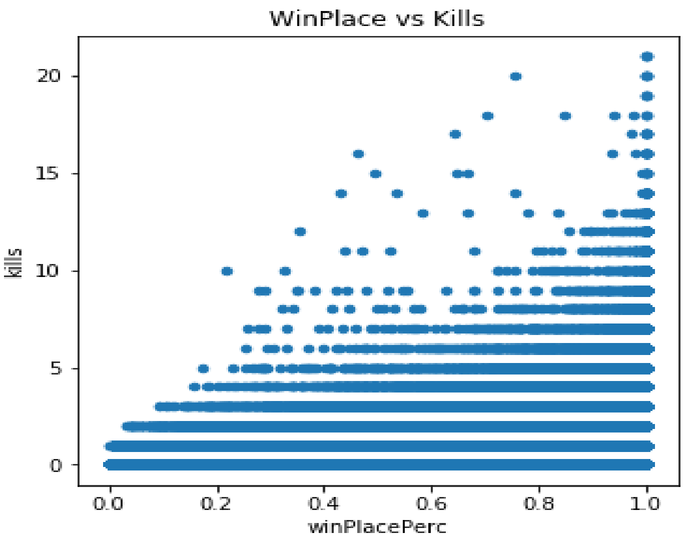
- **Data Collection:** We collected data from Kaggle data source having 29 features.
- **Data Cleaning:**
  - Remove Null Data: We had 1 null value in winPlacePerc, so we removed that row, as it was only one row.
  - Removing Outliers: We removed outliers such as:
    - If the total distance moved by a person is 0 but the kills for the same is very high, it can possibly be a cheater
    - It’s not possible to have made roadKills if the vehicleDistance is 0
    - A player cannot possibly walk more than 10000 metres so that’s an anomaly
    - Similarly, rideDistance greater than 20000 is also an anomaly and should be removed
    - longestKill is the longest distance from where a kill is made. A longestkill more than 1000m is very unlikely and can probably be done by a cheater.
- **Data Preprocessing and Preparation:**

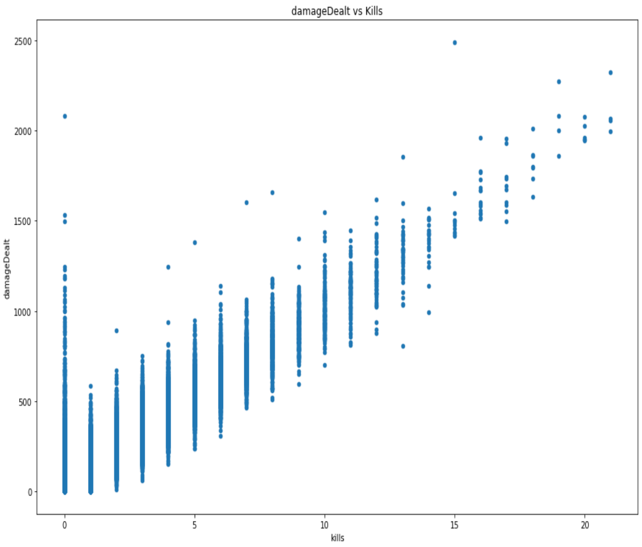
- Data Transformation: Created totalDistance column with summation of walkDistance, swimDistance and rideDistance.
- Label encoding the matchType column to integer type. 16 type of match types[solo,duo.. etc] were mapped to 0-15 values.
- Train Test Split: We divided our data into training and testing data by 80-20 ratio.

- **Data Correlation:**
  - Heatmap: Shows the correlation between all the 29 features of the dataset.



  - **Correlation with Target 1:**

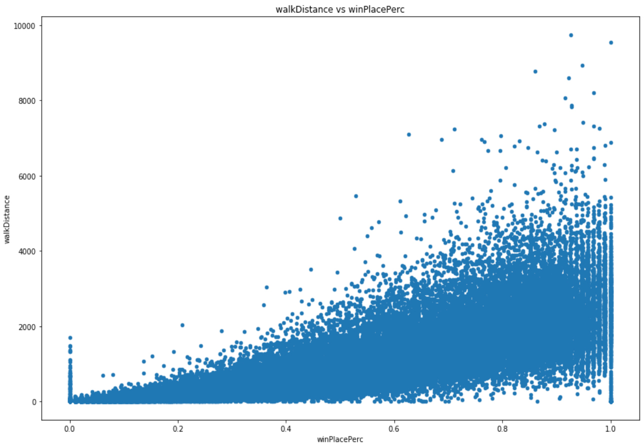


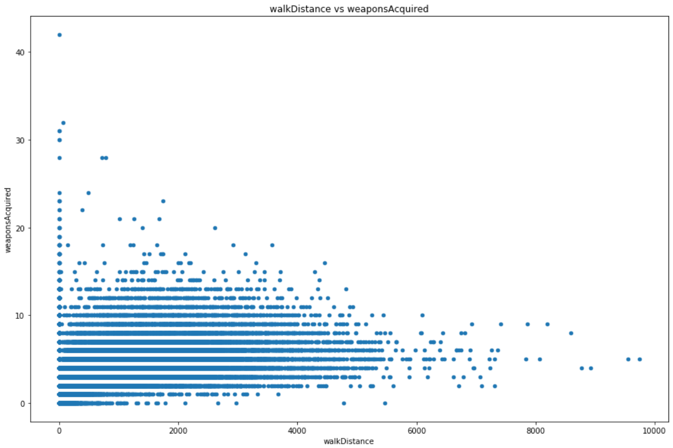


1.winPlace vs Kills: From the graph above, we can see that as the winPlacePerc of a player increases, the number of kills made by him also increases because obbviously he outkills the rest.

2.damageDealt vs kills : The plot shows that the kills has an increasing trend with the damageDealt. This is because, when a person kills someone, there’s high risk of getting shot by the opponent.

  - **Correlation with Target 2:**



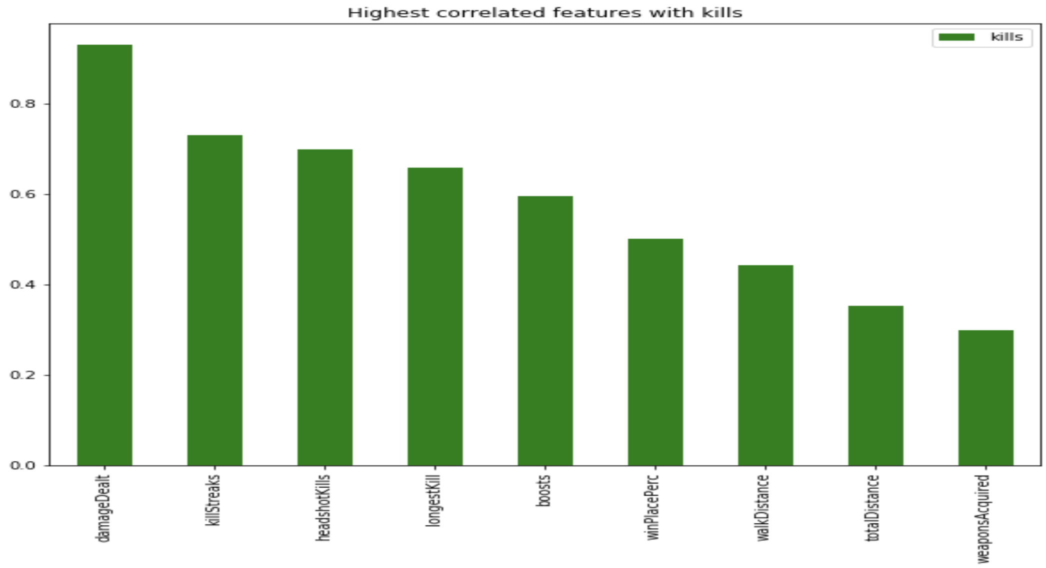


1.winPlacePerc vs walkDistance: walkDistance has high correlation with winPlacePerc and hence chosen as the second target variable.

2. walkDistance vs weaponsAcquired: walkDistance has high correlation with weapons because a player needs to move places to acquire the weapons as all the players are dropped on to the island without any. Also, most of the weapons are acquired initially in the game and it gradually decreases with walking once you get your desired weapons.

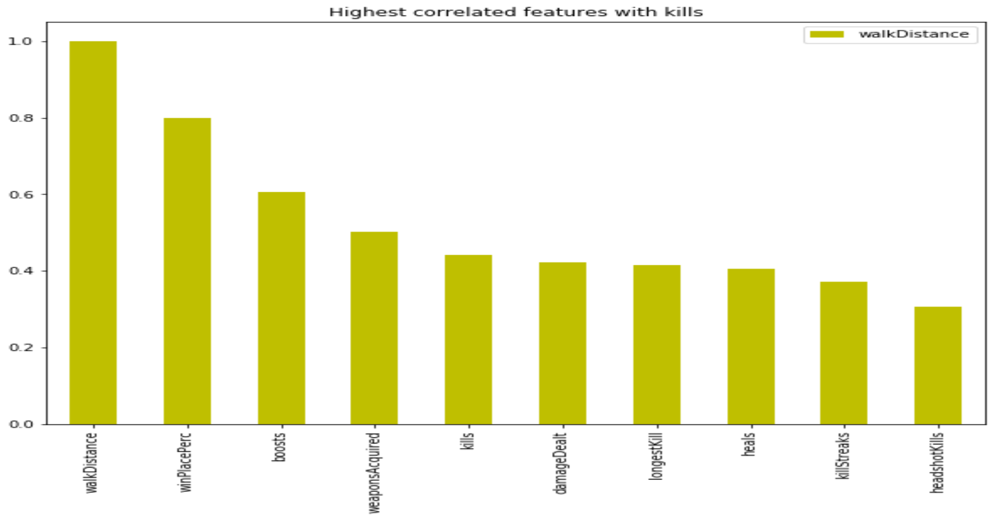
- **Feature Selection:**
    - Features with correlation > 0.5 for Target 1(Kills) were selected:

      The following plot shows the correlation of the target variable kills with other variables.



‘DamageDealt’ and *‘KillStreaks’* have the highest correlation with number of kills. KillStreaks correlation is high because the more you kill, more chance you have of kill streaks. Also, If you are killing more, you are taking more risks and that is why your damage dealt should be high. Overall there are many features that are highly correlated with kills.

    - Features with correlation > 0.5 for Target 2(WalkDistance) were selected:



      'winPlacePerc' and 'boosts' has the highest correlation with walkDistance, since you would have to move in order to have the top ranking, as region is continuosly contracted after some time. Also, boost having high correlation also makes sense, since more boosts means you are travelling more to collect the boost and can travel more.

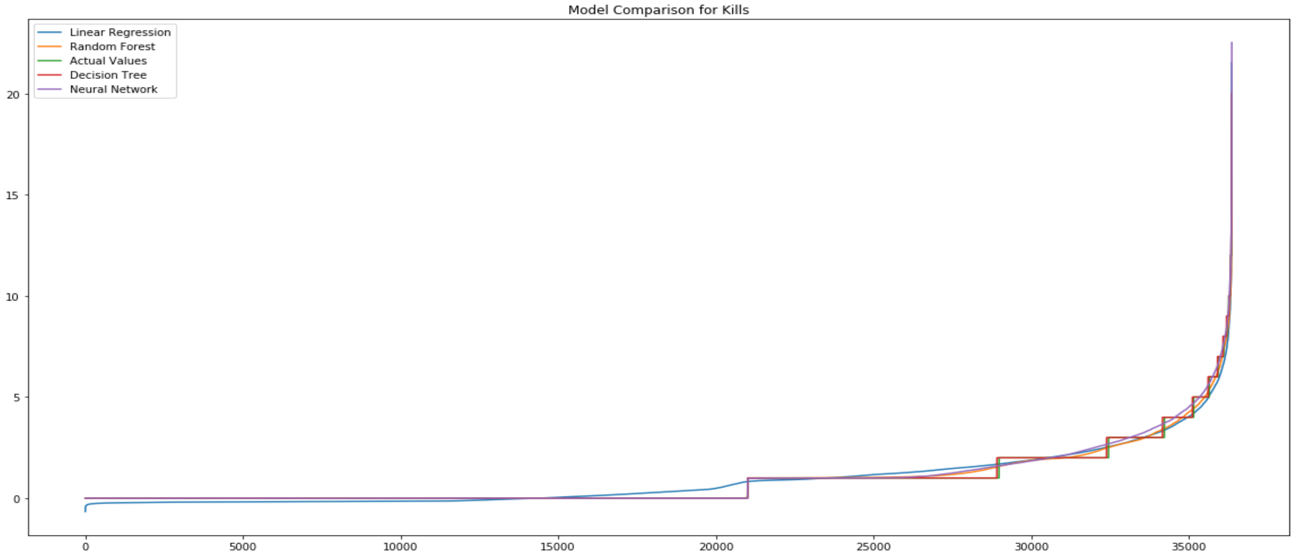
- **Model Selection**:
    - Linear Regression: As we wanted to predict the number of kills and distance walked, which are quantitative and continuous values, we chose regression model and we wanted to start with the simplest one: linear regression.
    - Decision Trees: As the results from linear regression were poor, we moved on to decision trees and chose entropy as the criteria so as to gain maximum information gain and improve the results.
    - Random Forest: To further improvise, we decided to run the random forest regressor model as it takes the aggregate result of many decision trees thus avoiding overfitting.
    - Neural Network: Since the results in the case of the target variable walkDistance were erroneous with the above models, we decided to train it with more complex model like Neutral Network
- **Model Evaluation**:
    - We used MAE because it would show mean absolute error and large errors are just as bad as small errors, unlike mse where they are treated proportionally to their magnitude, and hence, MAE would be a better choice here.

Models	MAE Metric (Kills)	MAE Metric (walkDistance)
Linear Regression	0.299	409
Decision Trees	0.167	357
Random Forests	0.131	353
Neural Networks	0.13	352.7

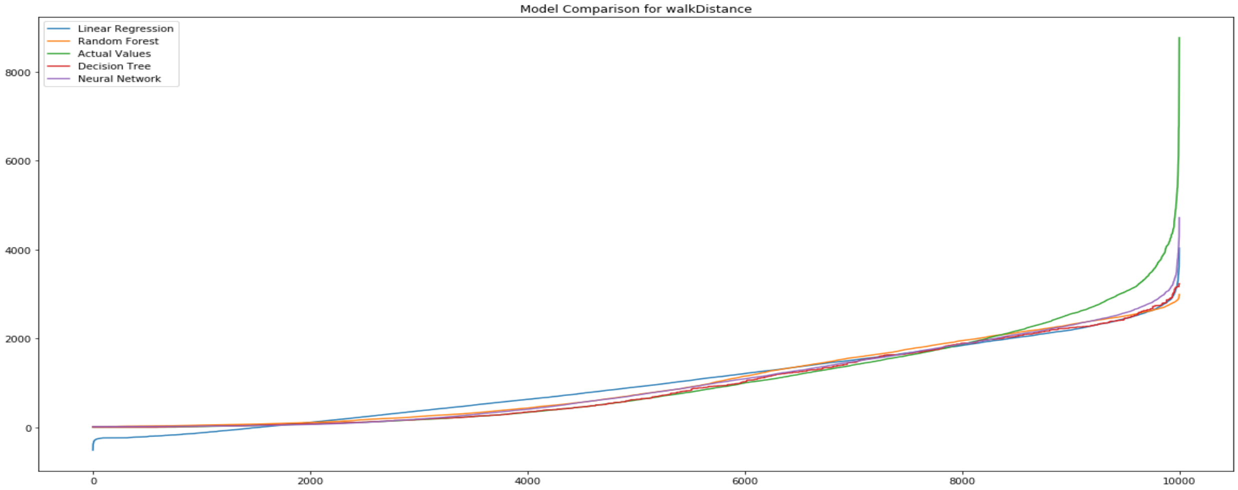
For Kills: We had MAE = 0.299 for linear regression and results improved when we used decision tree because data was not linear. And since, random forest is combination of decision trees, better results were expected here. Also, it deals with bias and variance better than previous two models. Also we had slightly better performance with Neural network after tweaking the performance metrics.

For walkDistance, we were having very poor results as the variance is too high and correlation with other features is very low. There are only two features three above 0.5 correlation (winPlacePerc, boosts, weaponsAcquired). We tried implementing random forest and neural network and best we got was 352.7 which is still very bad.

- **Model Comparison**:



The plot above shows that the results of Decision Tree Regressor and Neural Network are the closest to the actual values and hence display great performances.



From the plot above we can clearly conclude that random forest and neural network predict the closest values to actual values of walkDistance in the dataset but are still not close to it.

Updates Made From Proposal:

1. Updated Target variable - The target variable chosen in the proposal was matchDuration, but since it is exactly the same as predicting the winPlacePerc, which is already being solved, we decided to go with kills and walkDistance instead.
2. Data description - The data description was not written in detail mentioning the data types of all the features, we made sure to include it in the report.
3. Neural network model added - Because the other regression models like decision tree, random forest and linear regression delivered poor results, reason being, the correlation of the walkdistance feature is very less with other variables. Only 3 features : winPlacePerc, boosts and weaponsAcquired, had correlation >0.5 with it, so we trained neural network so that more weight would be given to these features.

Code Improvements:

We are getting great results for Kills prediction for all the models, we used Linear regression and Decision trees and we were able to further improve our results to MAE = 0.13 using Random Forest.

Predicting WalkDistance was more tough because of low correlation with other variables and MAE was too large with Linear regression, decision tree and Random Forest. To further improve the results, we tweaked the performance of models, like changing the min_samples_split in decision tree to 100 and min_samples_leaf to 12 and thus improving the error from 500 to 357.

Similarly, the performance of random forest was tweaked by changing min_samples_split in decision tree to 50 and min_samples_leaf to 20 and also used Neural Network to reduce MAE value. But the error values are still very high because of the high variance in walkDistance with other Features.

Conclusion: As the values were continuous, choosing the regression model was right choice. And as values were not linear, Random forest and Neural Network performed better than Linear Regression. We were able to predict kills accurately with all models, but predicting walkDistance is tough because it seems to be random, because it has high correlation with only two features.