**Predictive Analysis and Data Mining on**

**Counter Strike-GO Matches**

PORNIMA BANSODE

Dept. of Information Systems, Fall 2017

Northeastern University, Boston, MA

bansode.p@husky.neu.edu

**Abstract**:

Based on the CSGO stats, approx. 546197 players play matches 24 hours peak, and it’s 850486 for the all-time peak. Another statistic shows The *Global Offensive* professional scene consists tournaments hosted by third-party organisations and Valve-organised as majors. These majors have huge prize pools, originally it was set at $250,000, the prize pools for majors have risen, with [MLG Columbus 2016](https://en.wikipedia.org/wiki/MLG_Columbus_2016) having the first pool of $1,000,000. It becomes important for a professional player to look forward to winning the match in competitions which can be achieved by proper analysis and prediction. In this project we desire to analyse the Average Damage per round, Kills and Deaths of each player that would further build a simple predicting model for the players in the matches.

**Introduction:**

We aim at building different predictive models with high accuracy which will enable us to predict the most useful player in the match. Consider an example of a random players playing as a team in a match, and they lose as they have no knowledge of other team members who are the beginners. This becomes difficult for the professional players to win the match also this can affect the overall rating of a player. In this project we will try to predict the number of Kills by a player and the ADR (Average Damage per Round) which will help us to know how useful the player is for considering in a team.

**Related Work:**

Decision diagrams and other classifiers like genetic algorithms, Bayesian and neural networks have been used for a very long time for simple classification and decision support

One of the most important aspects of building a decision tree is to determine the best split attribute-value pair for a certain data-set. Thus, at every stage in the formation of the decision tree, the attribute-value pair that gives rise to the best split of the current data-set must be determined. The importance of estimating the best split is that otherwise, the resultant tree could be longer and wider. In the worst case, the resultant tree could be a skewed one, with only one non-leaf child per node and a smaller number of records being classified at each step.

**Approach:**

Initially we will apply the existed model to our dataset and analyse which model gives the best precision and prediction. The dataset consists of various float values, also the dependent variables are related to each other. The plotting shows there is a significant linear relationship between the variables. Hence, we consider the linear model here. Further, to analyse which model fits the best, we will try to use different predictive models to get the highest accuracy score. Our aim is to get the optimum model for the present dataset.

**Models used for training:**

Linear Regression:

Simple linear regression is a statistical method that allows us to summarize and study relationships between two continuous (quantitative) variables. The purpose of regression is to evaluate if a set of predictor variables does a good job in predicting the outcome. The linear regression estimates are useful in explaining the relationship between one dependent variable and one or more independent variables.

CART- Decision Tree Classification:

Decision tree is a predictive model which is used in machine learning, statistics and data mining. The decision trees are non-parametric supervised learning method which are used in classification and regression. Its aim is to create a model that can predict the value of target variable by decision making from the dataset.

K-Nearest Neighbour

k-NN is a type of instance-based learning, or lazy learning, where the function is only approximated locally, and all computation is deferred until classification. The k-NN algorithm is among the simplest of all machine learning algorithms. In k-NN classification, the output is a class membership. An object is classified by a majority vote of its neighbours, with the object being assigned to the class most common among its k nearest neighbour (k is a positive integer, typically small). If k = 1, then the object is simply assigned to the class of that single nearest neighbour.

Lasso Regression

Lasso (least absolute shrinkage and selection operator) is a regression analysis method used in machine learning and statistics that performs both regularization and variable selection to improve the accuracy of predictions.

Ridge Regression:

It is a technique to analyse multiple regression data that has multicollinearity in it. When the data consist of multicollinearity, the variance is too large. We can reduce the variance by adding a degree of bias to the regression which will reduce the standard errors.

Support Vector Machine

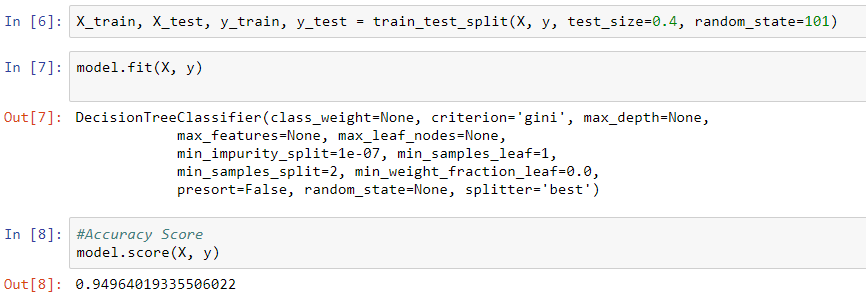
Support Vector Machine (SVM) is a supervised machine learning algorithm which can be used for both classification or regression challenges. It is mostly used in classification problems. In this algorithm, we plot each data item as a point in n-dimensional space (where n is number of features you have) with the value of each feature being the value of a coordinate.

Naïve Bayes Classification

In machine learning, naive Bayes classifiers are a family of simple probabilistic classifiers based on applying Bayes' theorem with strong (naive) independence assumptions between the features.

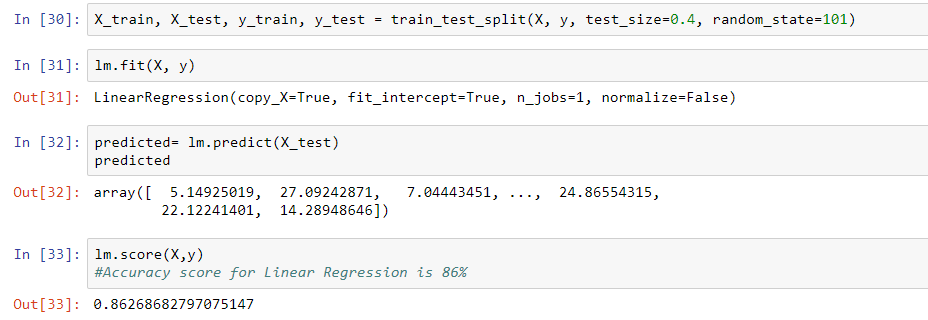
**Model Selection and Evaluation**

Decision Trees



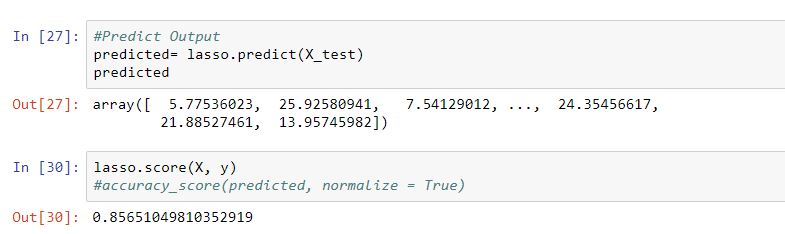
The Decision Trees gave the accuracy of 94%.

Linear Regression



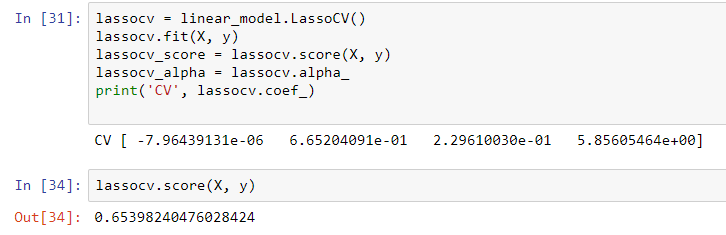
Linear Regression gave an accuracy of 86%.

Lasso Regression



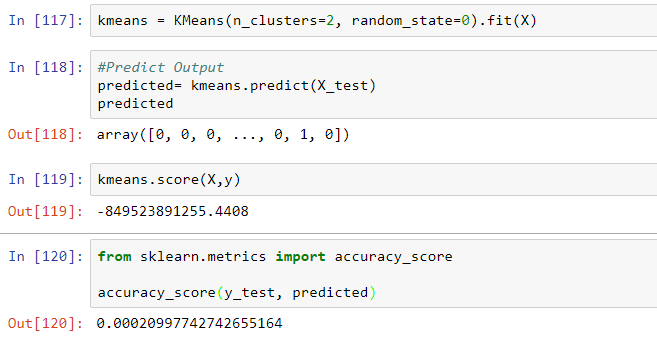
Gave an accuracy of 85%.

Lasso Regression with cross validation

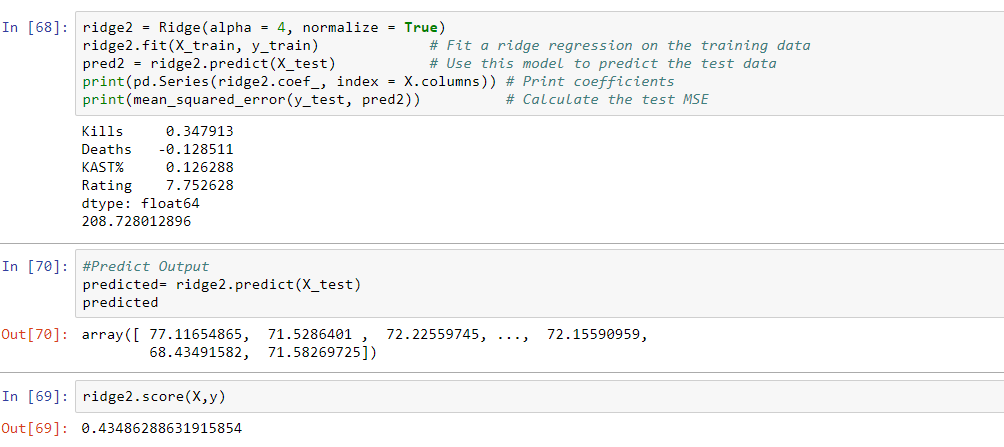


Accuracy score is 65%.

Clustering with K-means

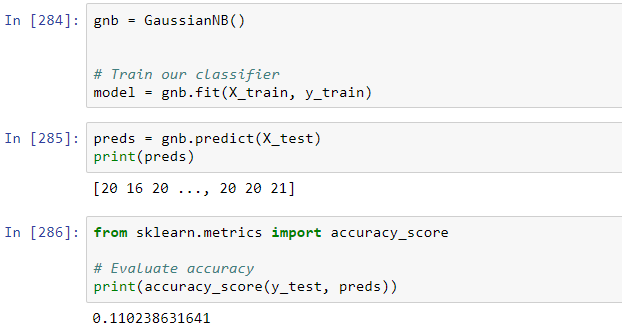


Ridge Regression



Accuracy of 43%

Naïve Bayes Classification



Accuracy of 11%

**Conclusion:**

From the training results, we analysed that after applying decision tree classification the accuracy score was 0.9496 which proves to be the optimum model. The next model is the Linear Regression model with accuracy score of 0.8626. Also, Lasso Regression model can be considered as a good model with the accuracy of 0.85651.

**Data Source:**

<https://www.kaggle.com/reagentx/HLTVData/data>

**Acknowledgment**

I would like to express my appreciation to Professor Nicholas Bear Brown, who has helped me to make things simpler and easier to understand. With his guidance, we had a proliferated knowledge on how Machine Learning algorithms are being used.

He has always motivated us in the project and gave a feedback on it to improve our adroitness.

# References

*Building Random Forest*. (n.d.). Retrieved from http://dataaspirant.com/2017/06/26/random-forest-classifier-python-scikit-learn/

*Cross Validation*. (n.d.). Retrieved from http://ogrisel.github.io/scikit-learn.org/sklearn-tutorial/modules/cross\_validation.html

*Decision Trees*. (n.d.). Retrieved from https://www.packtpub.com/books/content/predicting-sports-winners-decision-trees-and-pandas

*GAUSSIAN NAIVE BAYES CLASSIFIER IMPLEMENTATION IN PYTHONGAUSSIAN NAIVE BAYES CLASSIFIER IMPLEMENTATION IN PYTHON*. (n.d.). Retrieved from http://dataaspirant.com/2017/02/20/gaussian-naive-bayes-classifier-implementation-python/

*In Dept Naive Bayes Classification*. (n.d.). Retrieved from https://jakevdp.github.io/PythonDataScienceHandbook/05.05-naive-bayes.html

*Kmeans*. (n.d.). Retrieved from https://datasciencelab.wordpress.com/2013/12/12/clustering-with-k-means-in-python/

*Lasso Regression*. (n.d.). Retrieved from https://chrisalbon.com/machine-learning/lasso\_regression.html

*Linear Regression implementation*. (n.d.). Retrieved from http://dataconomy.com/2015/02/linear-regression-implementation-in-python/

*Ridge*. (n.d.). Retrieved from https://github.com/akaashagarwal/Ridge-Regression/blob/master/Ridge%20Regression.ipynb

*Ridge & Lasso*. (n.d.). Retrieved from https://www.analyticsvidhya.com/blog/2017/06/a-comprehensive-guide-for-linear-ridge-and-lasso-regression/

*Simple and Multiple Linear Regression*. (n.d.). Retrieved from https://towardsdatascience.com/simple-and-multiple-linear-regression-in-python-c928425168f9

*Support Vector Machine*. (n.d.). Retrieved from https://www.analyticsvidhya.com/blog/2017/09/understaing-support-vector-machine-example-code/