Machine Learning Approach to Predicting Performance of PGA Tour Golfers

Michael Lambert, Pierce Saly, Khris Wilhelm

**Abstract**

The PGA Tour is the largest golf organization in the world and keeps detailed statistics on how golfers perform at various tournaments. However, many of these statistics may not be strongly correlated with success; a golfer that can drive the ball 320 yards may not effective if he is a bad putter. Most PGA Tour tournaments consist of four rounds of golf and begin with 120-156 players. However, only the top 70 players (and ties) after two rounds make the cut and continue play in the third and fourth rounds; players outside the top 70 miss the cut and do not receive any prize money payouts. To succeed in golf and consistently make cuts on the PGA Tour, a golfer needs a combination of skills. We seek to develop a quantitative model that can assist in predicting whether a golfer will make the cut, given recent tournament performance. Our original goal was to show applications of the model to online sports gambling, to capitalize on recent changes in legislation surrounding the industry; though our current model is unlikely to be highly profitable, we believe it is a good starting point.

**Introduction**

Recent legislation changes in the United States have repealed a two-decade-old ruling known as PASPA (Professional and Amateur Sports Protection Act), which prevented sports gambling in most U.S. states [1]. Previously intended to attempt to uphold the integrity of both professional and amateur sports, it has been argued that the attempt to regulate state specific gambling legislation with federal laws is unconstitutional. This has created the potential for growth in the online sports gambling industry, which previously did not exist. Prior to the overturning of the legislation, the sports gambling industry was estimated to be worth roughly $2 Billion by 2023, and with the overturning of the legislation this figure is estimated to be between $3.1 Billion and $5.2 Billion [2].

Many traditional performance models look for directional bias in the odds setting market to identify predictable market trends when considering the field of golfers [3]. With the advent of large-scale data collection techniques as well as the development of novel statistical methods to derive more accurate performance metrics [4], it is possible to develop and train predictive models using a depth and quality of data that has never been possible before. We leverage this newly created data source to develop a predictive model for PGA Tour golfer performance, specifically in the context of whether the golfer will make the cut in each tournament. In the future, we wish to improve this model to capitalize on the expanding industry of online sports betting and create not only an accurate model, but also one that is highly profitable.

**Description of Available Data**

Data was obtained from the PGA Tour’s Shotlink toolset. This data has been supplied to academics and not-for-profit foundations since 2005, with the purpose of providing novel opportunities for learning [5]. The dataset contains performance metrics for all golfers participating in PGA Tour tournaments from 2005-2018. It includes both individual golfer statistics – such as driving distance and putting ability – and field-level averages for all golfers playing a tournament round on a given day. These performance metrics quantify varying aspects of the players’ ability and differ from purely demographic or physical characteristics of the player, which are excluded from this dataset.

The base data is structured in nature and can be accessed via queries on a secure web portal. Data tables are aggregated based on specific categories, such as course statistics, overall player performance, and tournament-level player data. Sample data can be accessed at [5]. We use tournament-level data to map players at each tournament over multiple seasons and consider how historical tournament performance can be used to assess the chance of a player making the cut at subsequent tournaments. Tournament performance can be analyzed shot-by-shot, round-by-round, or by entire tournaments. We use 10 years of data between the years of 2008-2018, aggregated by entire tournaments; we believe round-by-round and shot-by-shot data is too granular for our purposes, as it can accentuate potential outliers in performance specific to a bad hole or round.

The raw tournament data contains over 120 features, many of which are still very granular (for example, average performance on approach shots from 125-150 yards away from the hole – a player may only have 2 or 3 of these shots in an entire tournament, which is quite a small sample size). Rather than develop our own cumulative metrics to aggregate these features into larger buckets, we use a statistic called strokes-gained, developed by Mark Broadie [4]. Strokes-gained quantifies a player’s performance relative to the average performance of all PGA Tour golfers playing the same course on a given day. The foundation of the statistic is Total Strokes-Gained, which measures how much a player out-performs the average score in a tournament round (for example, if Rory McIlroy shoots 68 on a day where the average score is 71.5, McIlroy’s total strokes-gained is 3.5). Using shot-by-shot data, total strokes-gained is broken down into four individual metrics: Strokes-Gained Off-The-Tee (OTT), Approach-the-Green (APP), Around-the-Green (ARG), and Putting. The sum of these four categories is equal to the total strokes-gained. Strokes-gained statistics are useful because they provide an adjusted measure of a golfer’s performance relative to the baseline performance level for PGA Tour golfers that day, accounting for factors that impact the raw scores, such as weather conditions and the inherent difficulty of the golf course.

Since our dataset contains the four strokes-gained metrics described above, to reduce noise, we removed the redundant shot location variables from our dataset (as they are accounted for in the strokes-gained data). We also derived additional features from the dataset to quantify both long-term and short-term performance of the player. For long-term performance, we developed a yearly rolling average of strokes-gained in each of the four categories detailed above. To identify short-term performance, we computed each player’s average strokes-gained over his last 3 tournaments. We also computed the number of days since a player was last cut from a tournament, to see if recent missed cuts could impact the psychological performance of a player. Similarly, we included the number of days since a player successfully made the cut in a tournament, as we realized that players who play in fewer tournaments (because they are less talented) would have misleadingly large values for Days Since Cut.

Unfortunately, not all PGA Tour tournaments collect the shot-by-shot data required to compute strokes-gained metrics; because of this, there is missing data in our dataset. Created features that required missing strokes-gained information were imputed with a weighted mean of the available short-term performance metric and the long-term performance metric for the player. For example, if strokes-gained data were only available for 2 of a player’s last 3 tournaments, the 3-tournament value would be imputed as 2/3 of the player’s (available) 2-tournament average + 1/3 of the player’s yearly average. If there was insufficient data for this step to be performed, then a value of 0 was imputed. We chose to input a value of 0 because a strokes-gained measure of 0 in any category is the PGA Tour average and as such does not provide meaningful information for either positive or negative confirmation. Missing data for fields unrelated to strokes-gainedmetrics were imputed with the column mean. The GBA algorithm did not require as much imputation, as GBA trees are inherently able to account for missing data. In-spite of this built in error handling, strokes-gained metrics were still imputed for GBA; we believe an accurate strokes-gained estimate based on long-term and short-term averages is preferable to a missing value.

Some specialized PGA Tour tournaments (such as World Golf Championship events, team events, and match play events) do not have cuts, so these tournaments were removed from the dataset. Players who withdrew during an event were also removed from the dataset, as it is unclear whether they would have made the cut had they not withdrawn. After removing redundant shot location features (already contained in cumulative strokes-gained data) adding our created variables, and removing no-cut events and withdrawals, the final dataset used for training and testing contains 45,857 entries and 28 features, with an 80/20 split used for training and testing respectively. The dataset was approximately balanced, with 26,824 entries for “Made Cut” vs 19,023 entries for “Missed Cut”.

**Methods**

Three classification models were tested to solve this problem: a Gradient Boosting Algorithm (GBA); K-Nearest Neighbours (KNN); and a Support Vector Machine with a radial basis function kernel (SVM). The GBA was fitted using XGBoost with a tree-based algorithm to attempt to classify players at a tournament as “Made Cut” (1) or “Missed Cut” (0) using only historical performance of the player. The main benefit of XGBoost versus standard gradient boosted machines (GBM) is that XGBoost includes a regularization term Ω, in addition to the loss function **L**, to control for overfitting by balancing the bias variance trade-off [6]. The training algorithm takes on the simplified form of:

where is the objective function that attempts to minimize the loss that occurs during training and is a transformed vector of our included features. Our loss function **L** is defined here using cross entropy loss:

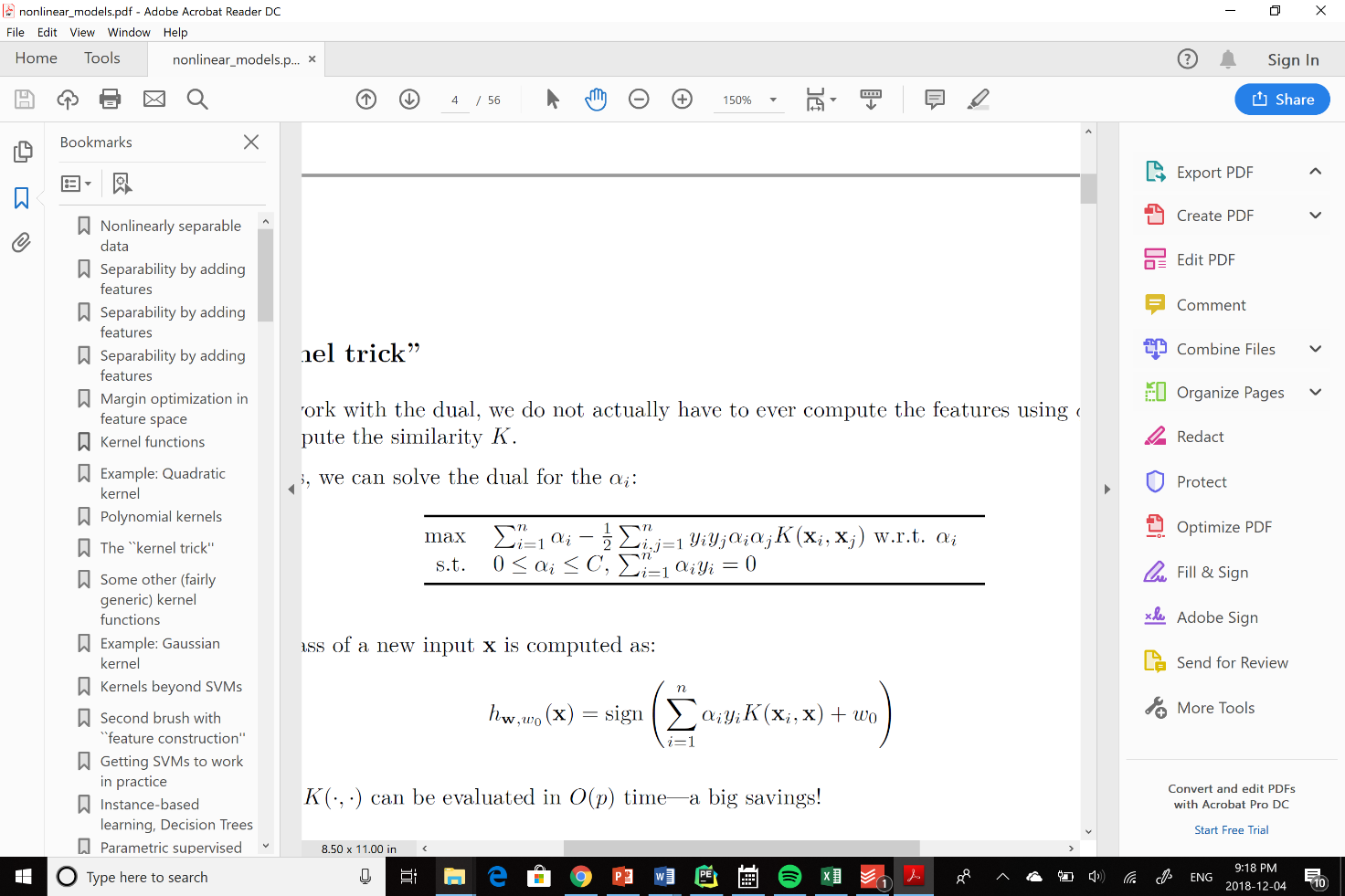
https://cdn-images-1.medium.com/max/1600/1*zi1wKAAGGt1Bn6mqo2MSFw.png

where expected loss increases as the predicted probability of the entry differs from the actual classification. Gradient boosting has several practical benefits for classification problems with sufficiently large data sets. For example, it can determine non-linear interactions as it builds successive trees, and it incorporates feature selection as well as loss optimization into a singular step. Additionally, compared to many other non-linear optimization functions, it occupies sufficiently less system resources, and has much faster run times. Finally, it has similar properties to linear regularization protocols and allows for penalization of irrelevant features or groups of features, automatically detecting regions of data that have less noise [7]. The parameters of the model were tested using an array search method that compared 300 different combinations of parameter, running 5-fold cross validation to optimize results and reduce generalization error. The final model utilized 50 fitted trees, with a maximum depth of 5, learning rate of 0.9, and a training set subsample of 80%.

The KNN model was fit using a classification task, with cross validation used to determine the value of K that minimized misclassification rate (in our case, K=13). The KNN model was fit using Euclidean distance as its distance metric, with data scaled prior to classification. KNN has the benefit of learning complicated non-linear decision boundaries by learning complex concepts via local approximations [8]. Unlike GBA, KNN does not have a built-in feature selection component, so it suffers from the curse of dimensionality in higher-level feature space. To attempt to combat this error, feature selection was performed using the GBA algorithm, with only important features being included in the final dataset used for KNN.

The SVM machine was fit using a radial basis function (RBF) kernel described as follows:

Note that larger values of σ cause the kernel to interpret similarities more generally, and smaller values of σ create more specific relationships at the risk of overfitting data [9]. The dual form of the SVM with the kernel trick incorporated is defined as:

The kernel SVM has substantial benefits when compared to a standard linear SVM (which attempts to solve for a linear decision boundary). Even when using a soft margin classification boundary, attempting to sort high dimensionality data is incredibly challenging when using a linear SVM. The kernel trick enables mapping of the dataset to a higher dimensional space through computing the dot product of the feature vectors. This allows for more efficient separations of non-linear data points in substantially more complex spaces than a standard SVM [10]. Thus, a radial basis function kernel SVM can determine more complicated relationships among the data. The parameters of the RBF SVM were tuned using a grid search method to assess potential values of σas well as the regularization parameter **C.** 178 different SVM models were created and compared using 5-fold cross validation. The final model used a regularization value of **C=100** and σ **= 0.1**. K-fold cross validation was used for tuning parameters because it is an embedded process in the GridSearch methodology; this allows for much more efficient run time when comparing and analyzing large volumes of models.

As the dataset is approximately balanced (26,824 “Made Cut” results versus 19,023 “Missed Cut”), performance was compared using both balanced accuracy and AUC-ROC measurements (see Table 1). F1-scores for outcome-specific performance were computed to identify any potential discrepancies between the model’s ability to classify positive versus negative results.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Balanced Accuracy | F1 Score for "Made Cut" | F1 Score for "Missed Cut" | AUC-ROC |
| GBA | 0.631844 | 0.73 | 0.37 | 0.64 |
| KNN | 0.5719 | 0.68 | 0.4 | 0.57 |
| SVM | 0.5718 | 0.73 | 0.25 | 0.55 |

**Table 1.** *Cross-Validated Performance Metrics for GBA, KNN, and SVM*

f

**Analysis and Results**

The best performing model was the GBA algorithm, which exhibited better performance compared to KNN and SVM in both balanced accuracy and AUC-ROC scores (as shown in Table 1). Models were evaluated based on AUC instead of F1 scores as our classes are approximately balanced (approximately half of the players in a given tournament will make the cut). As previously detailed, parameters of the boosting algorithm (tree depth, learning rate, gamma levels) were evaluated using 5-fold cross-validation, optimizing a cross-entropy loss function in the GridSearchCv package from SKLearn and a composite array of parameter combinations. 300 different boosting models were compared in validation. While the classes in the data set were balanced, F1 scores were computed for each algorithm to evaluate outcome-specific precision and recall. The GBA and SVM were equivalent in ability to predict positive outcomes (“Made Cut”) and were superior to KNN. KNN was the best at predicting negative outcomes (“Did Not Make Cut”), marginally beating GBA, and performing significantly better than the SVM. The resultant ROC curves are included in Appendix 1, Figure 1. The GBA model curve can be seen to have the best performance, with the maximum AUC. Appendix 1, Figure 2 shows the mean balanced accuracy and standard deviation of the three models. GBA not only has better balanced accuracy, but much tighter standard deviations, indicating a more consistent model as tested across 10 folds.

Given the complexity of the problem at hand, as well as the nature of the data, it is not surprising that GBA performed better than the other models. While we trimmed the initial data set of redundant variables, the dataset was still wide, and both SVMs and KNN are known to suffer from issues with higher dimensional spaces, even when an RBF kernel is used for the SVM. All three models were tested on data sets that were optimized using a Boosting Tree for feature selection to reduce dimension space. However,

since both SVM and KNN do not have feature selection steps imbedded into their architecture, the feature selection process was not optimized for the training process. Comparatively, GBA has inherent feature selection built into its loss optimization procedure. During the training step of the GBA, features are scaled based on their importance, and penalized or removed based on how important they are as predictors. This methodology allows for accurate feature weighting during the training step and helps reduce noise. The fitting procedure of a GBA tends to overfit data, minimizing the error to the point that the model will not generalize well. Combatting this either requires employing early stopping procedures to limit tree depth or having a sufficiently large dataset to reduce the chance the model fits on extraneous relationships. In our case, we have both sufficiently large data, as well as early stopping criteria (selected during the array searching and parameter optimization phase).

**Discussion**

The best methodology applied in this scenario (GBA model) was able to correctly predict whether players would make or miss the cut approximately 63.18% of the time. These results are not sufficiently accurate to derive a method through which we could profitably wager on PGA Tour tournaments. This is not an overly surprising result, both from the perspective of our dataset, and from a more qualitative analysis of the dynamics of PGA Tour golf. Our dataset had a significant amount of missing data that required imputing (approximately 18% of the long-term and short-term strokes-gained averages had to be imputed). It would be interesting to see the effect that a full strokes-gained dataset would have on model performance – presumably more complete data would be beneficial for performance – but this would require expanded shot-by-shot tracking at every PGA Tour event.

Intuitively, from observing the PGA Tour, golf is a difficult sport to predict – unexpected results are quite common. The gap between a top-20 player in the world and a borderline top-150 player who is struggling to stay on the PGA Tour is not very large. This means it is possible (and even common) for less-skilled players to outperform better overall players in the two-round sample from which the cut is determined. Even the best players in the world do not make the cut in every tournament; for example, Brooks Koepka, the current number one ranked player in the Official World Golf Rankings, missed 2 cuts out of 19 tournaments in the 2017-18 season [11]. The inverse is also true, as even the worst full-time PGA Tour players do not miss the cut in every tournament. It is possible that as an individual sport, golf performance inherently has more variance than a team sport like hockey or basketball; in a team sport, if one player performs poorly, other players may perform better and average out the overall team performance, whereas in an individual sport, there is no one else to balance out uncharacteristically poor play. Another factor that makes prediction difficult is that the same players do not compete in every tournament, which means that some tournaments have a stronger or weaker overall field of players. However, regardless of the strength of field, the same number of players make the cut in most PGA Tour events. Presumably, a weaker tournament would include more players that our model would deem likely to miss the cut. However, an equal number of players must make the cut in a tournament, no matter how difficult the field is. Finally, luck plays a part in the sport of golf. The difference between making and missing the cut often comes down to one or two shots over 2 rounds, and an unlucky bounce at the wrong time can be the difference between making the cut and missing it.

Though our model is not accurate enough to profitably wager on PGA Tour tournaments, we can infer practical information from it about how certain metrics impact player performance. Appendix 1, Figure 3 displays the most important features for predicting performance of a given player, as determined by the boosting algorithm (GBA). This feature selection was run against the entire feature set (including created features, before removing anything from the dataset), and features created during pre-processing dominated the feature importance. The short and long run strokes-gained measures (denoted by “\_SG”) had a high importance; they rendered the more granular features (such as driving distances, approach shot proximity from various distances, and putting from various distances) less important, as these were incorporated into the aggregate strokes-gained measures. This validates our reasoning for removing these features based on logical inference during the initial pre-processing phase. Additionally, the most important metrics are two created variables, quantifying time factors on player performance. The two variables, “DaysSinceCut” and “DaysSinceMadeCut,” measure the number of days since the player most recently missed and made a tournament cut. While it may seem redundant to include both features, doing so distinguishes between players who frequently make the cut and those who simply do not play frequently. A player who has made the cut the last three weeks will have the same “DaysSinceCut” value as a player who did not participate in the last three tournaments. However, by including the “DaysSinceMadeCut” metric, we can differentiate between these two groups of players, as the player who plays frequently and makes the cut will have a lower “DaysSinceMadeCut” score than the player who plays less frequently. This has implications for further time series analysis of the mental implications of current performance level (identifying whether consistently making or missing the cut impact a player’s ability to perform to a high degree). It would also be very interesting to investigate the classic “rest versus rust” argument to determine whether extended time off rejuvenates a player or leaves them out of practice.

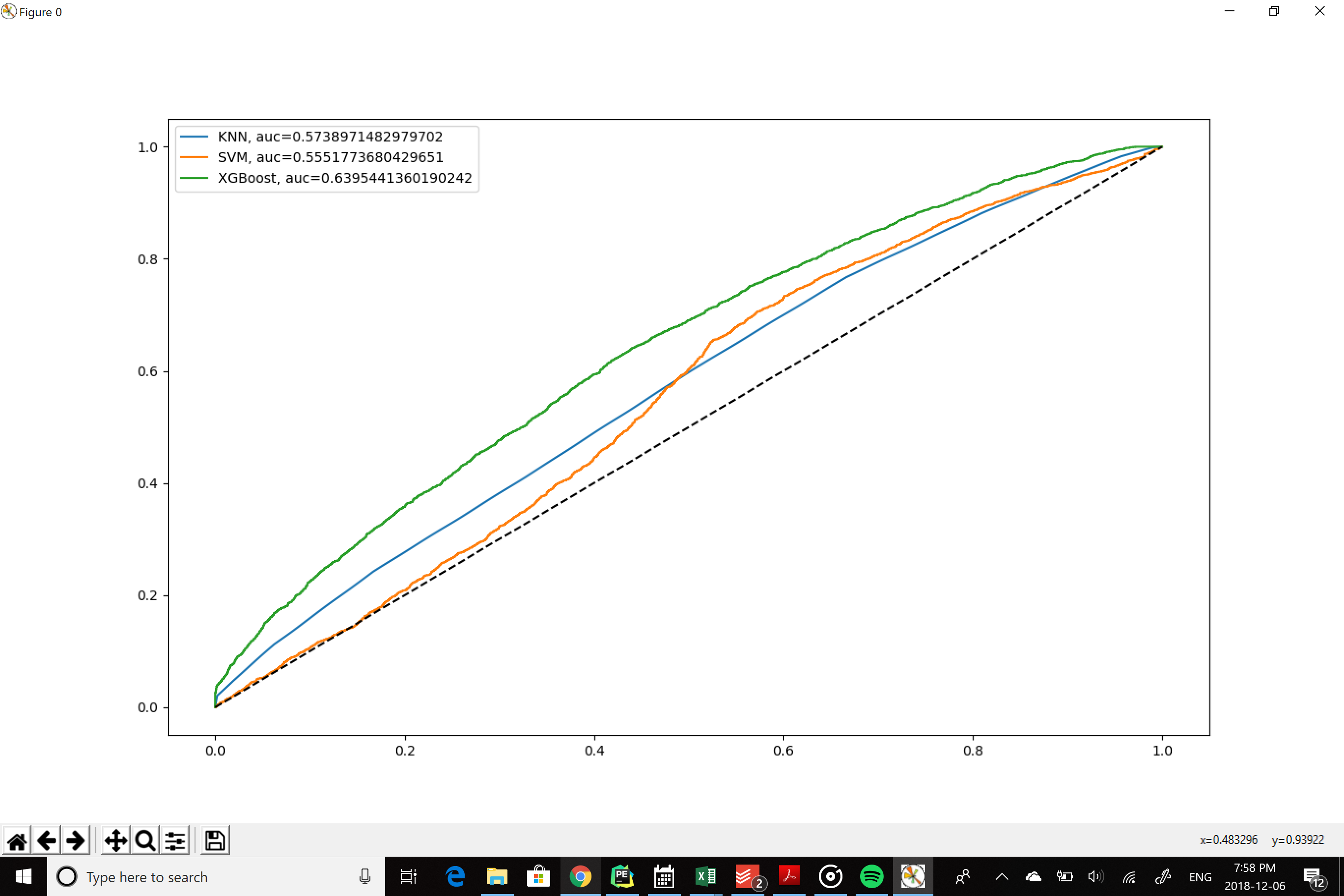
Going forward, we intend to continue our analysis and improve our model. Ideally, a future model would incorporate additional features, including player demographic data (such as age, height, and weight), and weather data (such as temperature, precipitation, and wind) to isolate player performance in specific conditions. It would also be beneficial to have strokes-gained data for every tournament, but this would require an improvement in PGA Tour shot-by-shot tracking. Interestingly, our most successful model (GBA) is much better at predicting made cuts than missed cuts. We do not have an intuitive explanation for this, so it would be fascinating to further investigate this phenomenon.

References

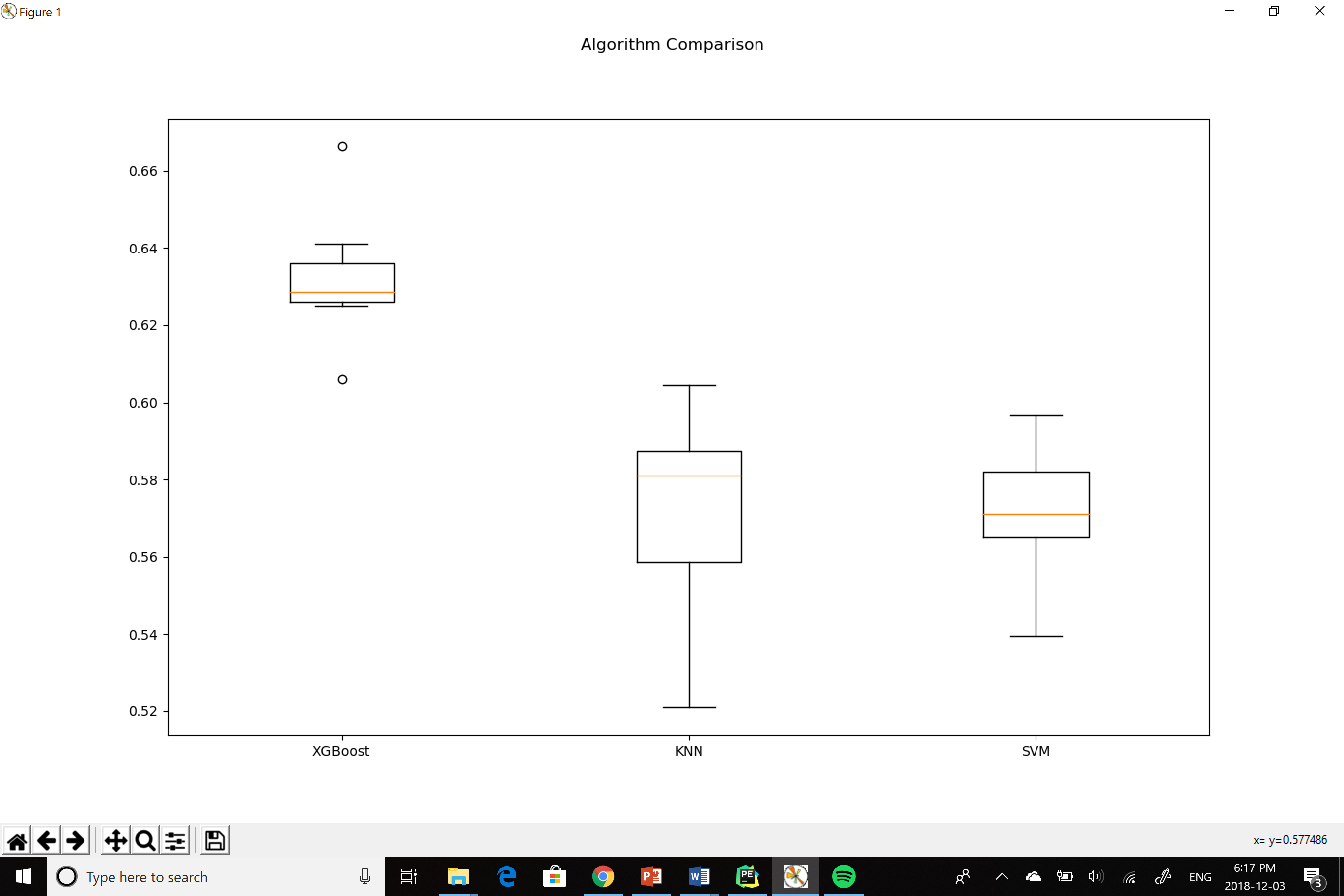
1. Grinstead, B. (2018, March 14). SUPREME COURT STRIKES DOWN PASPA, OPENING THE SPORTS BETTING FLOODGATES. Retrieved October 25, 2018, from <https://www.njonlinegambling.com/supreme-court-strikes-down-paspa/>
2. Raskin, E. (2018, June 28). U.S. Legal Sports Betting Market Projected To Be Worth Billions More Than Previously Believed. Retrieved October 25, 2018, from <https://www.usbets.com/us-sports-betting-projections-up-billions/>
3. Shmanske, S. (2005). Odds-setting efficiency in gambling markets: Evidence from the PGA Tour. Journal of Economics and Finance, 29(3), 391-402.
4. Broadie, M. (2012). Assessing golfer performance on the PGA TOUR. Interfaces, 42(2), 146-165.
5. <https://www.pgatour.com/stats/shotlinkintelligence/overview.html>
6. <https://xgboost.readthedocs.io/en/latest/tutorials/model.html>
7. Li, J., Cheng, K., Wang, S., Morstatter, F., Trevino, R. P., Tang, J., & Liu, H. (2017). Feature selection: A data perspective. ACM Computing Surveys (CSUR), 50(6), 94.
8. Keller, J. M., Gray, M. R., & Givens, J. A. (1985). A fuzzy k-nearest neighbor algorithm. IEEE transactions on systems, man, and cybernetics, (4), 580-585.
9. Andreas Müller (2012). Kernel Approximations for Efficient SVMs (and other feature extraction methods)
10. Scholkopf, B., Sung, K. K., Burges, C. J., Girosi, F., Niyogi, P., Poggio, T., & Vapnik, V. (1997). Comparing support vector machines with Gaussian kernels to radial basis function classifiers. IEEE transactions on Signal Processing, 45(11), 2758-2765.
11. <http://www.owgr.com/>

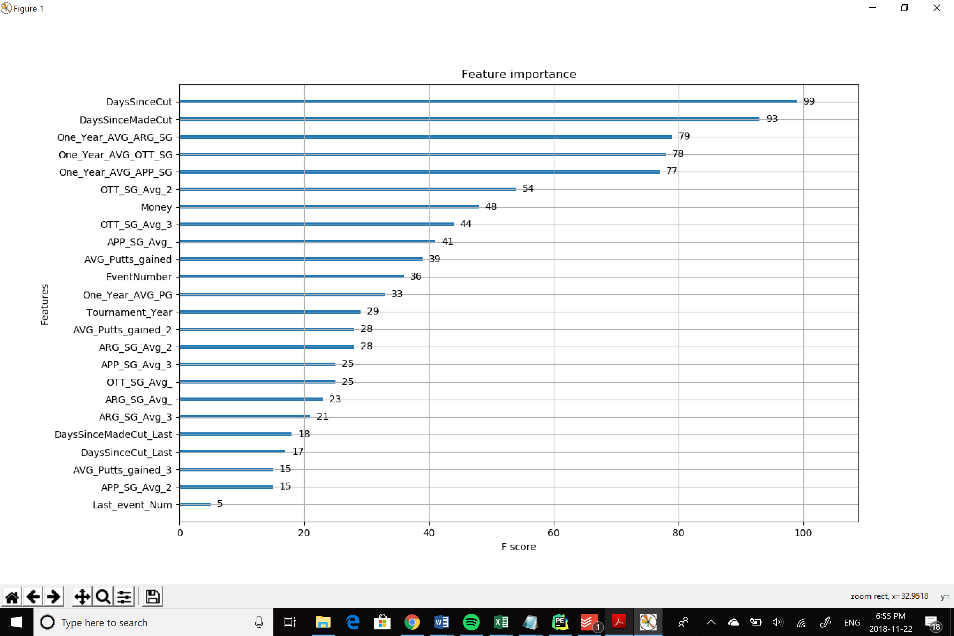
Appendix 1. Figures:

**Figure 1.** *Comparison of ROC curves for GBA, KNN and SVM*



**Figure 2.** *Boxplot of Mean Balanced Accuracy for K-fold validated GBA, KNN and SVM*



**Figure 3.** *F1 Scores for**Gradient Boosted Feature Selection*