COMP20008 Assignment 2 Report

Group: 80

Team Members:

Yilan Chen

Qingwen Tan

Liangyue Shen

Kangyu Zhu

Word count: 1619 (excluding table)

Aim

The aim of this project is to investigate, in 3 different servers: North America, Korea, and Europe, whether there is a relationship between League of Legends game-related factors and the role that players choose, as well as to determine which game factors have the greatest impact on the role. Finally, predict whether a player chooses to go to the lane or jungle through the behavior of the game. Players who like to play top lane or jungle may be interested in the results, which can improve their performance. The result may also benefit professional players and teams to develop training plans. Hence, the project helps to expand the understanding of the behavior of the top lane and jungle and improve training plans.

Datasets

The game League of Legends datasets were gathered from all challenger players' match history from January 2022 from three different servers: North America, Korea, and Europe. There are 5771 pieces of data with 20 columns in EUmatch dataset, 5760 pieces of data with 20 columns in NAmatch dataset, and 5697 pieces of data with 20 columns in KRmatch dataset.

Each piece of data describes the performance of a player in a single game and it introduces:

- d_spell: summoner spell on d key
- f_spell: summoner spell on f key
- champion: champion being played.
- side: side of map player is on red/blue
- assists: number of assists in match
- damage_objectives: damage to objectives
- damage_building: damage to building
- damage_turrets: damage to turrets
- deaths: number of deaths
- kda: (assist+kills)/deaths
- kills: number of kills
- turret_kills: number of turrets kills
- role: role being played out of the 2 (Top_Lane_and Jungle and other)

The role in the three datasets was the response variable for our investigation.

Data wrangling

After separating data into training and testing levels, we measure the dataset's quality as our first step. It has a high uniqueness score that assures minimized duplicates or overlaps and consistency. However, there are some nan values inside the data that are evenly spread in every feature. For these missing values, we decide to do the data imputation. The computer system directly extracts the data, so we assume humans' inappropriate manipulation causes the missing value. They are MAR and MCAR; therefore, removing the data with missing values is safe. For kda, we use formulas to do the fill. Firstly check if kda is nan or not. If it is nan, check if kills, assists, and deaths are nans; if all of them are not nan. We use the formula $kda = \frac{kill+assist}{death}$. Otherwise, we use $kda = kill + assist$. For the rest of the kda with nan values, we decide to delete them since they are a tiny part of the whole data. We are choosing the median to fill the rest of the nan value rather than the mean to avoid reducing the variance in the dataset as our primary imputation method. We do data processing for train and test, and respectively the difference is mainly infill median. We use the train-median fill train for a train set, then make MI feature selection. In the next step, use the test-median fill test and then directly put the function selected by the train set used in the test. The reason why filling nan for train and test separately is to avoid the information of test contained in the train set, thereby affecting the model. There are hundreds of outliers using the IQR Method, and we decide to keep them to avoid losing any valuable information.

Analysis Methods

Methods

Considering the project aim and datasets chosen, some analytical techniques were more relevant than others. Since this investigation deals with categorical data (roles) and inferring the relationship between other variables, we used KNN, Logistic regression, Decision Tree, and Random Forest techniques. We deliberately chose to use Chi-square test feature selection techniques to get which factors are important determinants of role (a nominal variable).

Training-test Split

Before further analysis, the data is split into training and test sets (0.33/0.67 split). The training set is used as a technique for evaluating the performance of a machine learning algorithm, whereas the test set is used to assess the fit and evaluate the models.

Preliminary Analysis

To gain a preliminary understanding of which game-related factors have the greatest impact on the role that players choose, we created heat maps between each variable with Pearson correlation and selected significant factors through the Chi-square test Feature Selection.

Firstly, based on the datasets from the Europe server, on initial inspection of the heat map (Figure 1), we observed that related to the role, the d_spell, damage_building, damage_taken, and damage_total showed deep colors. On the heap map, the larger values contained in a data matrix are represented as the deeper colors. So, these four factors seemed to hold correlations with the role. Furthermore, the Pearson correlation of damage_taken and role is 0.44, and that of damage_building and role is 0.39. Therefore, both damage_taken and damage_building show weak correlations with the role.

To further select the significant factors, we planned to use the Chi-square test Feature Selection. We were able to observe the p-value with the Chi-square test to determine the relationship between the testing feature and the response variable – role. If the p-value is less than 0.05, that means the null hypothesis can be rejected, and the feature is selected. Looking at the results in Table 1, we observed that 11 features are selected and they are 'd_spell', 'f_spell', 'champion', 'assists', 'damage_building', 'kills', 'level', 'time_cc', 'turret_kills', 'vision_score', and 'minions_killed'. After that, we also analyze the data from the other two datasets in the same way.

Secondly, based on the datasets from Korea server, the heap map (Figure 2) shows that damage_building (r = 0.43), damage_taken (r = 0.41), and damage_total (r = 0.31) hold weak correlation with the role. Furthermore, looking at the results from Chi-square test Feature Selection in Table 2, there are 16 features be selected, which are 'd_spell', 'f_spell', 'champion', 'assists', 'damage_objectives', 'damage_building', 'damage_turrets', 'kills', 'level', 'time_cc', 'damage_taken', 'turret_kills', 'vision_score', 'damage_total', 'gold_earned', 'minions_killed'.

Thirdly, based on the datasets from North America server, the heap map (Figure 3) shows that damage_building (r = 0.38), damage_taken (r = 0.42), and damage_total (r = 0.31) hold weak correlation with the role. Furthermore, looking at the results from Chi-square test Feature Selection in Table 3, there are 12 features be selected, which are 'd_spell', 'f_spell', 'champion', 'assists', 'damage_building', 'kills', 'level', 'time_cc', 'damage_taken', 'turret_kills', 'vision_score', 'minions_killed'.

Figure 1. Heat map with Pearson correlation for the dataset of Europe server.

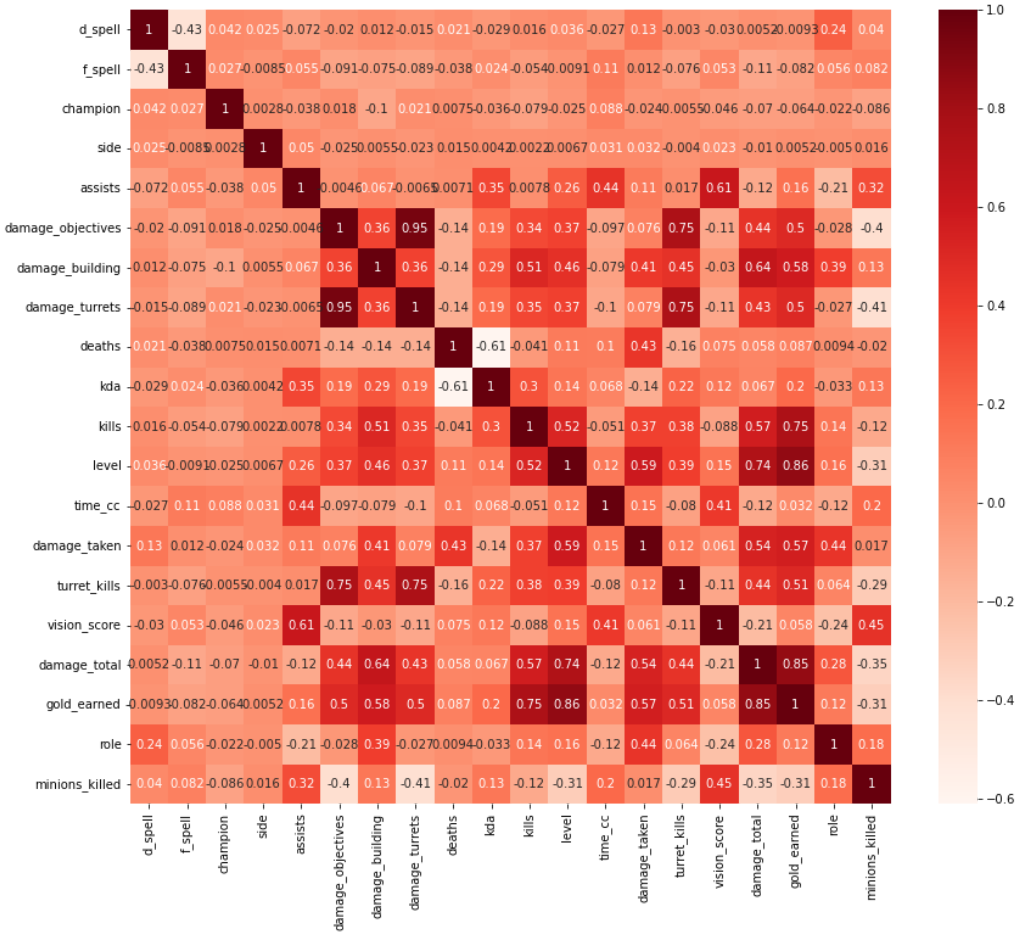


Table 1. The Chi-squared and p-value of selected features for dataset EUmatch

Feature	Chi-square	p-value

d_spell	752.79	0.00
f_spell	898.21	0.00
champion	2658.08	0.00
assists	181.02	0.00
damage_building	3087.42	0.02
kills	133.68	0.00
level	107.04	0.00
time_cc	212.79	0.00
turret_kills	61.38	0.00
vision_score	365.08	0.00
minions_killed	112.44	0.00

Figure 2. Heat map with Pearson correlation for dataset of Korea server.

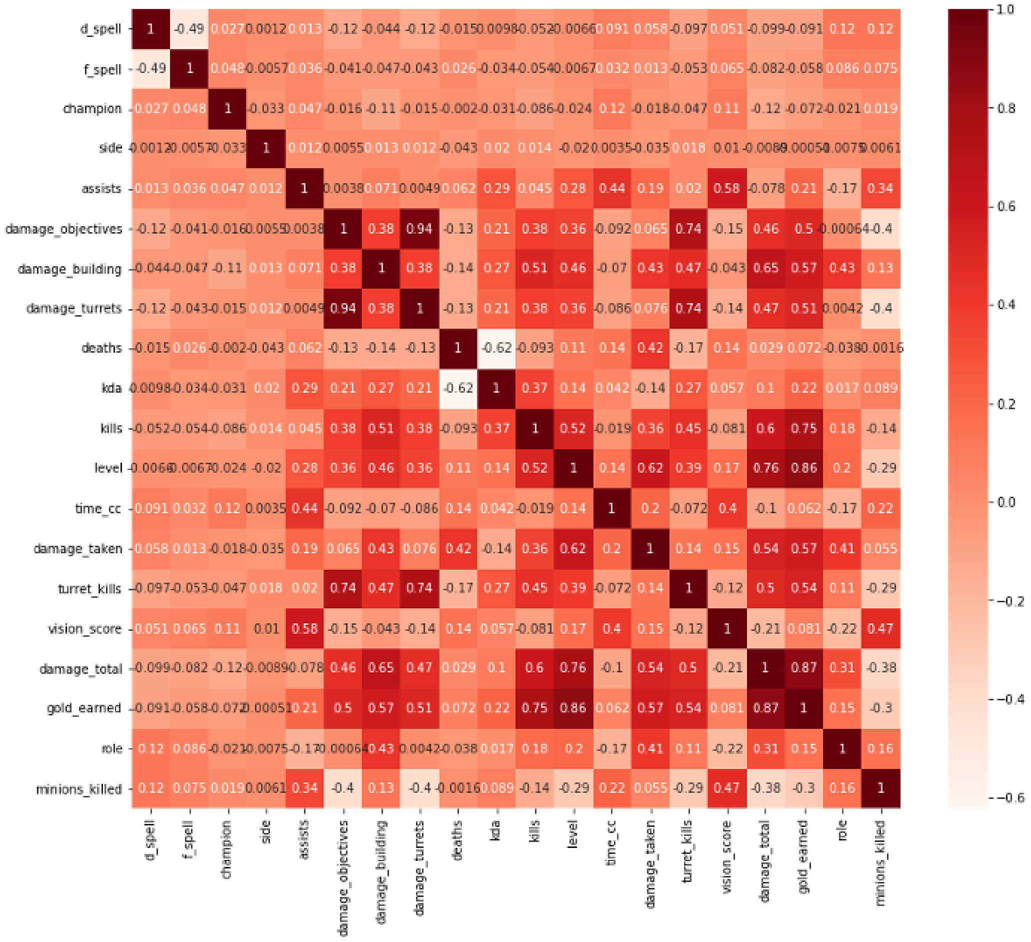


Table 2. The Chi-square and p-value of selected features for dataset KRmatch

Feature	Chi-square	p-value
d_spell	1107.34	0.00
f_spell	724.27	0.00
champion	2639.50	0.00
assists	118.68	0.00
damage_objectives	2224.44	0.02
damage_building	2987.37	0.00
damage_turrets	2213.85	0.03
kills	176.24	0.00
level	142.93	0.00
time_cc	309.23	0.00
damage_taken	3111.39	0.01
turret_kills	124.85	0.00
vision_score	325.55	0.00
damage_total	3377.58	0.02
gold_earned	3016.24	0.01
minions_killed	88.49	0.00

Figure 3. Heat map with Pearson correlation for dataset of North America server.

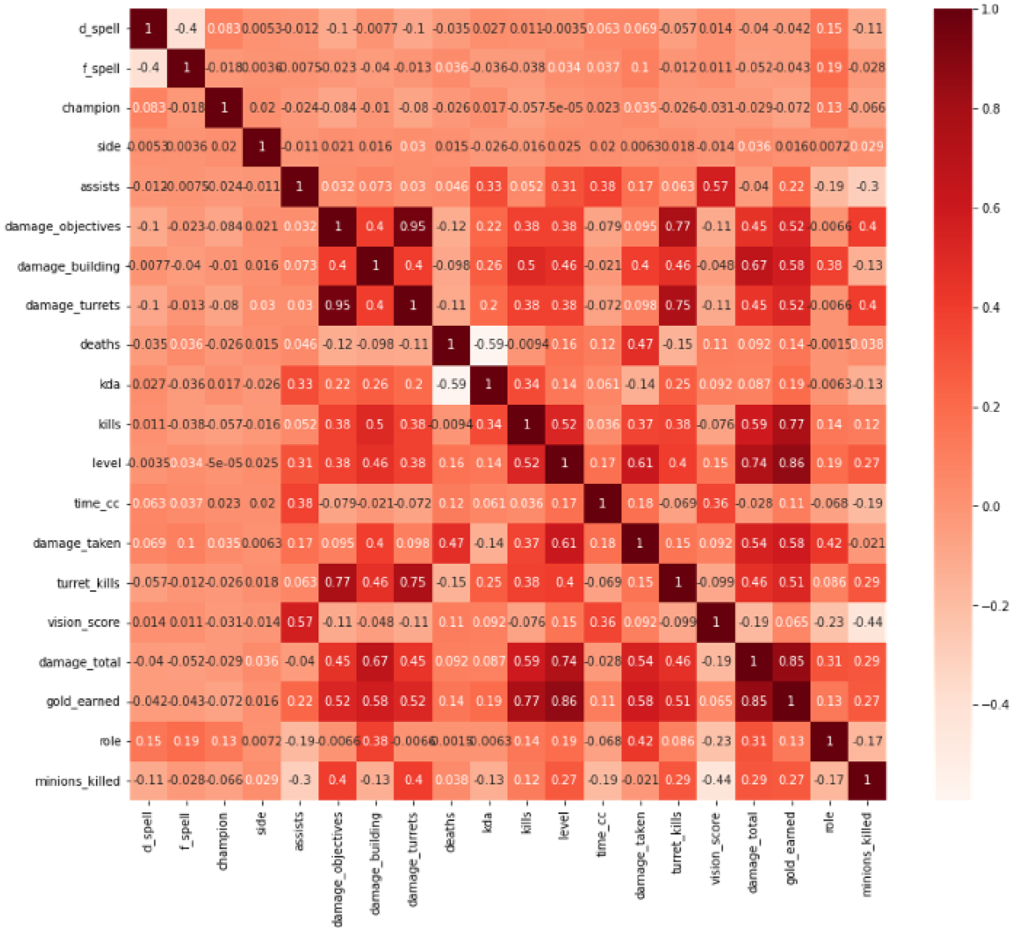


Table 3. The Chi-square and p-value of selected features for dataset NAmatch

Feature	Chi-square	p-value
d_spell	935.67	0.00
f_spell	927.82	0.00
champion	2623.60	0.00
assists	144.67	0.00
damage_building	2996.91	0.01
kills	119.74	0.00
level	145.29	0.00
time_cc	165.37	0.00
damage_taken	3180.00	0.045
turret_kills	46.06	0.00
Vision_score	327.70	0.00

minions_killed	103.64	0.00

Modelling

As the role was the response variable in our investigation, it held a binary classification problem. In this project, we fitted four models in total, KNN, Logistic regression, Decision Tree, and Random Forest technique, and the Random Forest technique was the best model for all three datasets, to get a better understanding of whether game-related factors have the meaningful impact on the role. To compare these models, it mainly focused on four parameters, accuracy, precision, recall, and F1-score, where F1-score is a composite parameter related to precision and recall. Table 4, Table 5, and Table 6 indicated the comparison of these four models from the dataset of Europe, Korea, and North America respectively. A larger value of accuracy and F1-score showed a better model. Therefore, the Random Forest was the best model with the largest parameter values.

Table 4. Parameters for four different models respectively of dataset EUmatch.

Models	Accuracy	Precision	Recall	F1-score
KNN	0.61	0.56	0.52	0.54
Logistic Regression	0.72	0.71	0.64	0.67
Decision Tree	0.73	0.71	0.64	0.67
Random Forest	0.82	0.84	0.73	0.78

Table 5. Parameters for four different models respectively of dataset KRmatch.

Models	Accuracy	Precision	Recall	F1-score
KNN	0.77	0.73	0.69	0.71
Logistic Regression	0.81	0.79	0.73	0.76
Decision Tree	0.81	0.79	0.73	0.76
Random Forest	0.88	0.89	0.80	0.84

Table 6. Parameters for four different models respectively of dataset NAmatch.

Models	Accuracy	Precision	Recall	F1-score
KNN	0.67	0.63	0.59	0.61

Logistic Regression	0.81	0.79	0.74	0.76
Decision Tree	0.81	0.79	0.74	0.76
Random Forest	0.86	0.87	0.79	0.83

Discussion

From the previous analysis, Random Forest is the best model for three dataset. The accuracy of Random Forest for three datasets is more than 0.8 and F1-sorce of Random Forest for dataset NAmatch and KRmatch are also more than 0.8. Although F1-sorce of Random Forest for dataset EUmatch is 0.78, it is close to 0.8. Furthermore, the Random Forest AUC for EUmatch (Figure 4) is 0.92, for KRmatch (Figure 5) is 0.94 and for NAmatch (Figure 6) is 0.94. The results are significant and the Random Forest model is a valuable model, which can work well.

Decision tree and Logistic Regression have lower and similar accuracy and F1-socre than Random Forest. But the performance of KNN is much worse than the other three models, which is less than 0.7 in datasets NAmatch and EUmatch.

Compared with the other three models, Random Forest is more complex and Random Forest does not depend on the importance of the simple feature, it constructs a multitude of decision trees at training time, and then allows for better data generalization.

Figure 4. Random Forest AUC for the dataset of Europe server

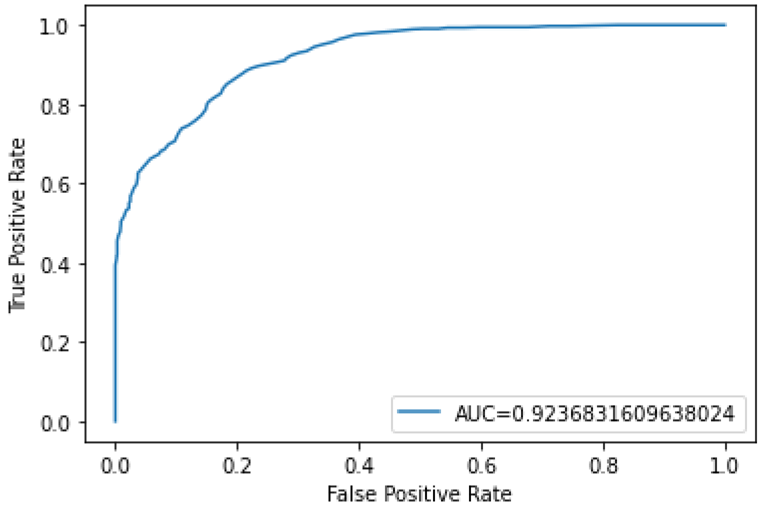


Figure 5. Random Forest AUC for the dataset of Korea server

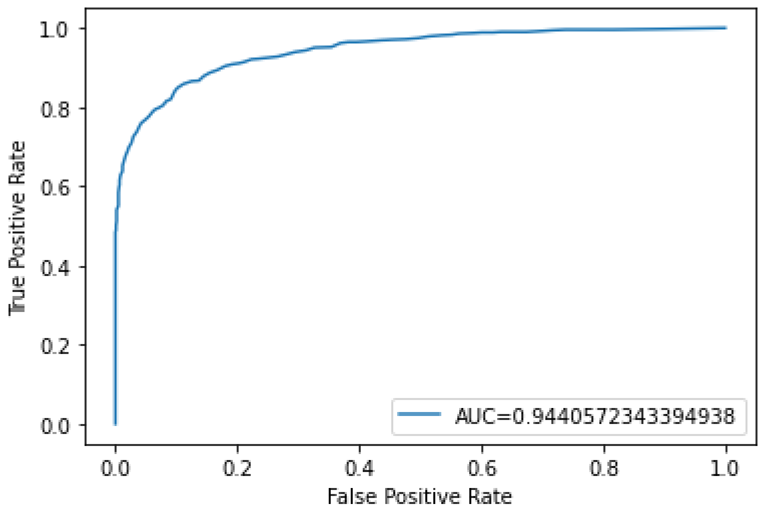
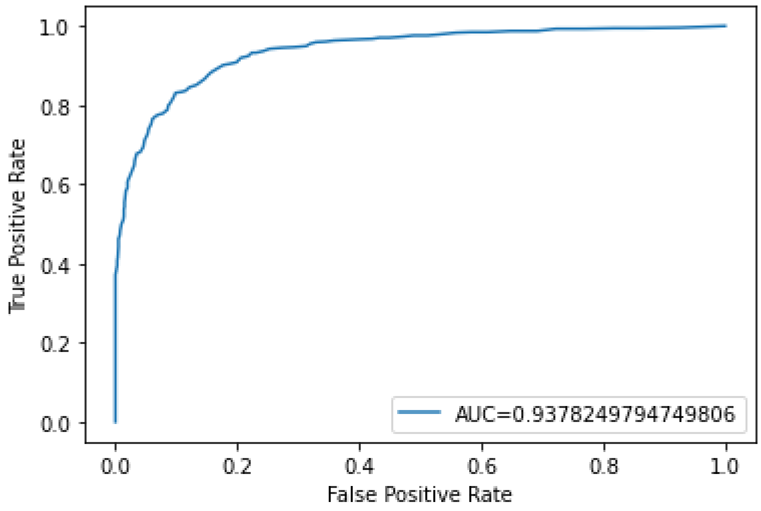


Figure 6. Random Forest AUC for the dataset of North America server



Evaluation

Single train-test split is used to evaluate the performance of the machine learning algorithm, instead, k-fold cross validation would produce more accurate results than the train-test split as it reduces the randomness for extreme cases.

As our aim is to predict the role of a player by firstly finding the related factors which have a relationship with “role”. Based on the general knowledge of the game, there are five roles, top, jungle, mid, ADC, and support. Each role has different jobs and of course, their in-game performance is totally different. However, the dataset we are given combines the top and jungle into “Top_Jungle” and mid, support, and ADC into “Other”, which makes our prediction model not as good as if we are given five roles instead.

Moreover, there are too many missing values in the dataset, we will lose too much data if we remove all the missing data, instead, we fill them by substituting their median value, which may cause the data to be distributed differently.

In the future, doing k-fold cross-validation will be a great improvement. When collecting the data, make sure to keep the data as complete, and high quality as possible for further analysis. A better model such as XGboost can also be used in the future.