Predicting the result of Long Tennis game using Machine learning and historical data

1st Sai Krishna Thode
dept. Information Science
University of North Texas
Denton, USA
saikrishnathode@my.unt.edu

2nd Kishan Suwal
dept. Information Science
University of North Texas
Denton, USA
kishansuwal@my.unt.edu

3rd Joshnavi Chenreddy
dept. Information Science
University of North Texas
Denton, USA
joshnavichenreddy@my.unt.edu

4th Ashwitha Eravelly
dept. Information Science
University of North Texas
Denton, USA
Ashwithaeravelly@my.unt.edu

***Abstract*—Tennis is an intense and unpredictable sport that serves as a captivating playground for fans around the globe. One of the hardest tasks however, is to forecast a match's result.. While mystifying fans, predicting the winner of a match is crucial for fans looking for evidence and betting markets based on which to bet. Conventional methods based on common sense and expert analysis, on the other hand, frequently fail due to the complexity of the aspects involved in the determination of results. The discipline of data science, which employs machine learning algorithms to analyze historical data, gives hope for an accurate prediction. This paper aims to increase the accuracy of forecasting tennis match outcomes. By analyzing historical data and using advanced machine learning methodologies, the research attempts to achieve the objective. The goal comprises building models to forecast outcomes, deciding to influence factors, and evaluating models using several algorithms. The research obtains a broad and rich dataset from various sources, such as Kaggle, including players' statistics, outcomes, ranks, and their respective betting odds. The project embarks on a mission to explore data and pre-process it before engineering relevant features.**

Keywords: Tennis match prediction, Machine learning, Historical data analysis, Predictive modeling, Data science, Sports analytics.

I. Introduction

Tennis, a game characterized by exhilarating rallies and intense suspense, looms large in the minds and hearts of sports lovers across the globe. The sport's fascinating allure stems not only from its sheer display of skill and athleticism but also from the level of unpredictability it offers; every match provides a cocktail of skill, strategy, and mental strength like no other. But within every thrilling serve and volley lies an intriguing mystery: the ability to accurately predict match results. [7]The application of the ability to predict tennis match outcomes is expansive: for the fans, it promises the possibility of identifying the likely winners, thus making the watching experience more wholesome, while for the various betting markets, it forms the core of the customers' decision-making matrix. The challenge, however, has been developing match outcome prediction models, which, traditionally, have revolved around a subjective analysis of result histories or expert opinions. As a result, the scope of the models has always been somewhat limited in capturing all factors influencing the outcome, from players' form and fitness to the nature of the courts and associated weather conditions. The advent of data science in this generation of computers opens up a new scope of sports analytics: the integration of classical statistics and prediction algorithms to analyze vast amounts of historical data. This study hopes to find out how the integration of these disciplines can push the boundaries of accuracy in predicting tennis match outcomes. The study hopes to use a vast range of historical player statistics, their head-to-head match records, tournament history, and betting odds data to develop models offering nuanced outcomes in match predictions. The study is also fueled by the desire to transform the sport's analysis method: from subjective guesses to objective decisions based on proper data. The application of the scientific method, integrating the behavioral aspects of the sport with the rigidity of classical statistics, promises to change the future of predicting tennis match outcomes.

However, a fresh method for approaching this issue is offered by the expanding discipline of data science. Large amounts of historical data can be analyzed using machine learning to find patterns that might not be immediately apparent using more traditional techniques. The ability to make more accurate forecasts is one of the potential benefits of this data-driven strategy, which may also alter how industry experts and tennis enthusiasts perceive and anticipate tennis match results. The objective of this research is to forecast tennis match winners by using data analysis techniques. There are many other factors in the sport that might affect the outcome, such as player ability, tournament setup, and historical performance. Our goal is to develop predictive algorithms that are capable of providing precise forecasts by analyzing a large dataset of past matches. This helps tennis professionals and betting markets make educated decisions, in addition to offering helpful information to spectators.

In this research, we used Linear Regression,Logistic Regression,Decision Tree and Random Forest to predict the result of the game.

II. Statement of the Problem

The main problem faced in this field is the successful prediction of tennis match winners, which is a critical issue for both hardcore enthusiasts and the betting market. Despite the abundance of expert opinion and intuitive understanding of

the process, the current methodologies fall short of accounting for the multitude of factors affecting the match. The balance between player skills or form, the conditions of the tournament, and the factors external to the competition themselves is compromised by an immense combination of variables. This project gains from the developments in data science and the ability to recognize patterns in data by analyzing models and historical data in this way. The project is fundamentally aimed at defying the aforementioned challenge by utilizing machine learning algorithms and the analysis of historical data to compile models that would be more accurate in determining the winner of the game. An extensive analysis and experimentation will enable stakeholders to gain vital knowledge in the sector of sports analytics.

III. Literature Review

Yue, Chou, Hsieh, and Hsiao (2022) carried out research on utilizing the Glicko model to predict tennis matches. Their research, published in PloS One, explored the application of this model and its potential in predicting tennis match outcomes. While their study focused specifically on tennis, it underscored the importance of leveraging rating systems and performance metrics to enhance decision-making processes in sports analytics [14]. Herbinet (2018) delved into the realm of football result prediction using machine learning models. Herbinet emphasized the significance of leveraging advanced algorithms to forecast football outcomes accurately. This research highlighted the potential of machine learning in sports analytics and underscored the importance of feature selection and model optimization [8]. Joseph, Fenton, and Neil (2006) contributed to the field of sports prediction by exploring Bayesian nets and other ML techniques for predicting football results. Published in Knowledge-Based Systems, their study showcased the effectiveness of probabilistic models in capturing complex relationships between team performance and match conditions, further advancing the arsenal of tools available for sports analytics [10]. Prasetio (2016) presented a study on predicting football match results with logistic regression at the 2016 International Conference on Advanced Informatics. This research underscored the importance of statistical modeling techniques in sports prediction, particularly in the context of logistic regression. By estimating the likelihood of different outcomes based on historical data, Prasetio's work contributed to the growing body of literature on sports analytics [11]. Baboota and Kaur (2019)performed machine learning-based predictive analysis and modeling of football results, with a particular emphasis on the English Premier League. The International Journal of Forecasting released their research, which highlighted how important feature selection and model optimization are to getting precise forecast. This study provided valuable insights into the application of machine learning techniques in football result prediction, contributing to the evolving landscape of sports analytics [2]. Hvattum and Arntzen (2010) explored the use of ELO ratings for predicting match results in association football. Published in the International Journal of Forecasting, their research highlighted the utility of ELO ratings in assessing relative skill levels of teams and its relevance in sports analytics. By leveraging historical data and rating systems, Hvattum and Arntzen's study offered valuable insights into the predictive modeling of football match outcomes [9].

IV. Objective of the study

The objective of of the study is to predict the game result by using machine learning in historical data. During which we will consider different features and machine learning methods that will yield the higher accuracy.

- To develop predictive models for tennis match outcomes using ML algorithms such as Linear Regression,Logistic Regression,Random Forest and Decision Tree.
- To identify the key features contributing to the result of the game .
- To evaluate the performance of the predictive models using different ML algorithms and determining the best model among them.

V. Data collection

We obtained the historical tennis match data from Kaggle. The data collecting procedure entails obtaining the data from a variety of sources, including internet databases, sports websites, and specialist datasets.

Data Source Link:

https://www.kaggle.com/code/mayankneil/country-wise-analytics-visualizations-tennis/input?select=ltennisindv.csv.

Usually, this data contains information about players, matches, competitions, rankings, and other relevant information. The data must be cleansed, processed, and arranged after it has been gathered in order to be analyzed. This could entail managing missing values, eliminating duplication, and guaranteeing consistency between variables.

VI. Exploratory data analysis and hypothesis

In order to forecast tennis match outcomes using machine learning and historical data, exploratory data analysis, or EDA, is essential. During this stage, the dataset is analyzed and visualized in order to understand its overall structure, identify trends, find abnormalities, and gain insights that guide the creation of hypotheses and predictive models [6].

A. Information of the Data

The df.info() function in pandas provides a brief summary of a DataFrame, which is comparable to a table with rows and columns. It provides information about the number of rows and columns, the types of data (text or integers) in each column, the number of missing values, and the amount of memory used by the DataFrame. When working with a fresh dataset, this is useful as it provides an overview of its structure and enables you to identify any problems before delving into a more thorough examination.

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1844 entries, 0 to 1843
Data columns (total 23 columns):
 #   Column                 Non-Null Count  Dtype
---  ------                 --------------  -----
 0   tourney_id             1844 non-null   object
 1   tourney_name           1844 non-null   object
 2   surface                1844 non-null   object
 3   draw_size              1844 non-null   int64
 4   tourney_level          1844 non-null   object
 5   tourney_date           1844 non-null   object
 6   match_num              1844 non-null   int64
 7   winner_id              1844 non-null   int64
 8   Player_seed            1844 non-null   int64
 9   Result of the game     1844 non-null   int64
 10  Dominent_playing_hand  1844 non-null   object
 11  Player height          1773 non-null   float64
 12  Player_nationality     1844 non-null   object
 13  Player_age             1844 non-null   float64
 14  N_ace                  1814 non-null   float64
 15  w_df                   1814 non-null   float64
 16  w_svpt                 1814 non-null   float64
 17  w_1stIn                1814 non-null   float64
 18  w_1stWon               1814 non-null   float64
 19  w_2ndWon               1814 non-null   float64
 20  w_SvGms                1814 non-null   float64
 21  w_bpSaved              1814 non-null   float64
 22  w_bpFaced              1814 non-null   float64
dtypes: float64(11), int64(5), object(7)
memory usage: 331.5+ KB
```

Fig. 1. Information of the Data

B. Stastistical Analysis of the Data

df.describe() function in pandas offers a statistical overview of the numerical columns of a DataFrame. It provides important metrics that aid in your comprehension of the distribution and properties of your data. Upon executing df.describe(), a table with the subsequent statistics for every number column is produced.

```
df.describe(include = 'all') # describing all numerical and categorical data
```

	tourney_id	tourney_name	surface	draw_size	tourney_level	tourney_date	match_num	winner_id	Player_seed	
count	1844	1844	1844	1844.000000	1844	1844	1844.000000	1844.000000	1844.000000	1
unique	312	45	1	NaN	1	222	NaN	NaN	NaN	
top	2009-425	Barcelona	Clay	NaN	A	19-07-2004	NaN	NaN	NaN	
freq	24	276	1844	NaN	1844	28	NaN	NaN	NaN	
mean	NaN	NaN	NaN	36.659436	NaN	NaN	81.120390	105053.610087	4.816703	
std	NaN	NaN	NaN	10.290573	NaN	NaN	103.986547	8392.649765	3.295539	
min	NaN	NaN	NaN	28.000000	NaN	NaN	21.000000	100644.000000	1.000000	
25%	NaN	NaN	NaN	32.000000	NaN	NaN	26.000000	103476.500000	2.000000	
50%	NaN	NaN	NaN	32.000000	NaN	NaN	30.500000	104022.000000	4.000000	
75%	NaN	NaN	NaN	48.000000	NaN	NaN	47.000000	104745.000000	7.000000	
max	NaN	NaN	NaN	64.000000	NaN	NaN	507.000000	210013.000000	17.000000	

11 rows × 23 columns

Fig. 2. Statistical Analysis of the Data

VII. Data Analytics

To forecast the result of a prolonged tennis match using machine learning and historical data, a scientific approach to data analytics is required. Gathering pertinent data, cleaning and preparing it, doing exploratory data analysis (EDA), developing extra features, choosing a suitable model, training the model, and assessing its performance are usually several crucial phases involved in this process.

A. Checking for Null values and Duplicates

A crucial step in data preprocessing is to look for duplicates and null values, as these might lead to mistakes and skew analysis. Duplicates introduce bias by placing an excessive amount of weight on repeated data points, while null values, which indicate missing data, can cause computations to malfunction and lower the accuracy of machine learning models. df.isnull(), which indicates where data is missing, and df.isnull().sum(), which counts the amount of missing values in each column, are two tools in pandas that you can use to identify null values. Use df.duplicated().sum(), which counts all duplicate rows, and df.duplicated().flags(), which marks repeated rows, to discover duplicates. Null values can be handled by using df.dropna() to remove them or df.fillna() to fill them with a specified value. The df.drop_duplicates() function gets rid of duplicates. After examining the dataset we came to know that there are no duplicates. To preserve data integrity and guarantee the accuracy of analysis and modeling, these problems must be resolved.

B. Data preprocessing and Data cleaning

In data preprocessing and data cleaning, replacing missing values with the mean is an often employed technique. This technique, which is also known as mean imputation, involves taking the mean of all the values that are present in a column and using that mean to replace any missing or null values.Here the figure shows filling null values with mean and inspecting their correlation.

```
#Data Cleaning
#Remove the column which has NaN values. the number has reduced to 1743 from 1844
table_1_filter = table_1.dropna()
table_1_filter
```

	Dominent_playing_hand	Player_nationality	Result of the game	Player_seed	Player_age	Player height	N_ace	w_df	w_svpt	w_1stIn	w_1stWon	w_2ndWon
0	L	ARG	1	7	24.501027	183.0	6.0	7.0	66.0	38.0	31.0	10.0
1	R	ESP	1	4	22.995209	180.0	1.0	0.0	34.0	18.0	17.0	11.0
2	R	MAR	1	2	28.481862	193.0	8.0	8.0	93.0	53.0	41.0	16.0
3	R	AUS	1	8	23.978097	180.0	4.0	5.0	77.0	35.0	24.0	25.0
4	R	AUS	1	8	23.978097	180.0	5.0	5.0	84.0	40.0	32.0	25.0
...	...	...	...	...	...	...	...	...	...	...	...	...
1833	R	SRB	0	1	33.911020	188.0	2.0	6.0	116.0	72.0	44.0	21.0
1834	R	SRB	0	5	29.141684	185.0	1.0	4.0	57.0	30.0	18.0	17.0
1836	R	ESP	0	5	33.013005	183.0	1.0	1.0	65.0	47.0	29.0	7.0
1839	R	ARG	0	4	28.673511	170.0	0.0	4.0	102.0	70.0	44.0	13.0
1841	R	ESP	0	6	29.771389	188.0	4.0	1.0	53.0	44.0	24.0	3.0

1743 rows × 17 columns

Fig. 3. Dropping the Null columns

```
# filling null values with mean and inspecting their correlation
hf = df.copy(deep = True)
for col in null_col:
    hf[col] = hf[col].fillna(hf[col].mean())
plt.figure(figsize=(10, 8))
sns.heatmap(df[numerical_col].corr(), annot=True, cmap='coolwarm', fmt=".2f", linewidths=0.5)
plt.title('Correlation Heatmap')
```

```
Text(0.5, 1.0, 'Correlation Heatmap')
```

Fig. 4. Replacing missing values with mean

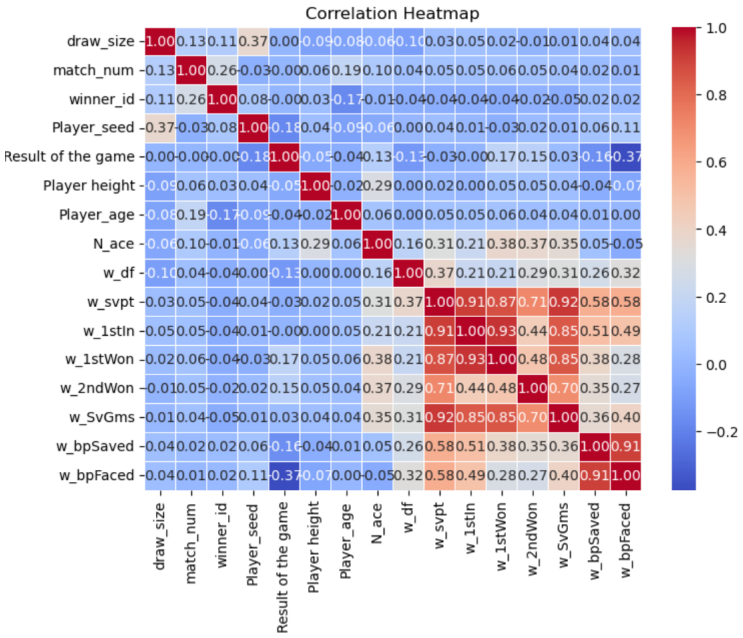

Fig. 5. Correlation Heat map for numerical columns

From this Heat map we can say that w_bpfaced feature is highly correlated with our target variable i.e. result of the game.

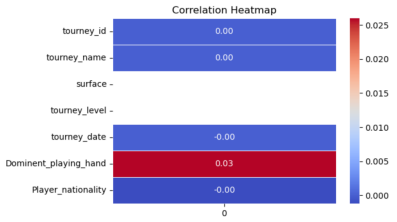

Fig. 6. Correlation Heat map for categorical columns

Out of all categorical columns Dominent_playing_hand fetaure is some what highly correlated to our target column i.e., result of the game.

From careful observation between dropping the missing columns and replacing missing values with mean we came to know that there is no much difference between the both so we decided to replace missing values with mean.

C. Checking whether the target variable is baised or not

Here we are checking whether the target variable i.e. Result of the game is baised or not using Gussian Distribution.

```
In [16]: # Inspecting whether our target data is baised or unbaised

column_data = df['Result of the game'].dropna()
mu, std = norm.fit(column_data)

plt.hist(column_data, bins=20, density=True, alpha=0.6, color='b',)

xmin, xmax = plt.xlim()
x = np.linspace(xmin, xmax, 100)
p = norm.pdf(x, mu, std)
plt.plot(x, p, 'k', linewidth=2)

plt.xlabel('Result of the game')
plt.ylabel('Density')
plt.title('Gaussian Distribution of ' + column_name)

# Show the plot
plt.show()
```

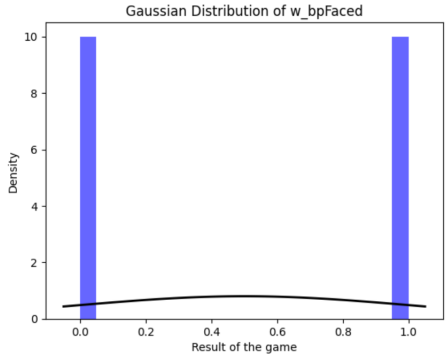

Fig. 7. Gussian Distribution of w_bpfaced

Here the graph shows that the target variable i.e. Result of the game is not baised.

D. Assigning the target variable and independent variable

As linear regression and logistic regression model we kept target variable as “Result of the game” and rest of the columns as independent variables which would help us to predict the result. The independent variables were ’Dominant playing hand’, ’Player nationality’, ’Player seed’, ’Player age’,’ Player height’, ’N_ace’, ’w_df’, ’w_svpt’,’Surface’ and ’w_1stIn.

E. Converting the string value to numerical

Since Decision tree only takes the numerical value for running the model, we converted features which has string value to numerical values. The features impacted were ’Dominant playing hand’, ‘Surface’ and ’Player nationality’. In order to change the value we used LabelEncoder from sklearn.

```
from sklearn.preprocessing import LabelEncoder

# Since decision tree takes only numberical values we change the playing hand and nationality to the number
le_Dominent_playing_hand = LabelEncoder()
le_Player_nationality = LabelEncoder()
le_surface = LabelEncoder()

inputs_d['Dominent_playing_hand_n'] = le_Dominent_playing_hand.fit_transform(inputs_d['Dominent_playing_hand'])
inputs_d['Player_nationality_n'] = le_Player_nationality.fit_transform(inputs_d['Player_nationality'])
inputs_d['surface_n'] = le_surface.fit_transform(inputs_d['surface'])
inputs_d.head()
```

	Dominent_playing_hand	Player_nationality	Player_seed	Player_age	Player height	N_ace	w_df	w_svpt	w_1stIn	w_1stWon	w_2ndWon	w_SvGms
0	L	ARG	7	24.501027	183.0	6.0	7.0	66.0	38.0	31.0	10.0	11.0
1	R	ESP	4	22.995209	180.0	1.0	0.0	34.0	18.0	17.0	11.0	7.0
2	R	MAR	2	28.481862	193.0	8.0	8.0	93.0	53.0	41.0	16.0	15.0
3	R	AUS	8	23.978097	180.0	4.0	5.0	77.0	35.0	24.0	25.0	12.0
4	R	AUS	8	23.978097	180.0	5.0	5.0	84.0	40.0	32.0	25.0	11.0

Fig. 8. Converting the numerical string value to numerical

F. Machine Learning Models

Computational algorithms called machine learning models are made to learn from data so they can make judgments or predictions without the need for explicit programming [12] .These models operate on datasets to carry out various tasks like classification, prediction, grouping, and recommendation by spotting patterns, correlations, and structures. As the models learn from the patterns in the data, their parameters are adjusted during the training process. They can make educated guesses on fresh, untainted data once they have been trained. Applications for machine learning are numerous and include speech and picture recognition, financial forecasts, driverless cars, and sports analytics. In this research, we used four machine learning models namely:

1) Linear Regression
2) Logistic Regression
3) Decision Tree
4) Random Forest

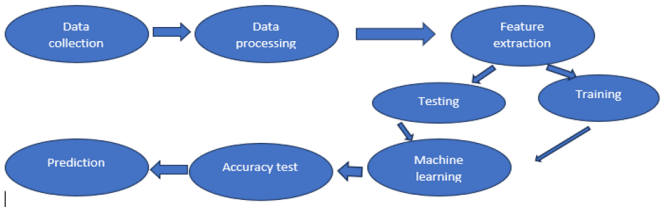

Fig. 9. work flow

- Linear Regression

A statistical technique called linear regression is used to model the relationship between two continuous variables. In data analysis and predictive modelling, its one of the most straightforward and widely applied methods [13].

Many fields, including economics, finance, social sciences, sports, engineering, and more, use linear regression. It is used to forecast results, understand relationships between variables, and identify critical elements influencing particular phenomena.

Following are our approaches for developing multiple regression models

1) Multi linear regression using all numerical features.
2) Linear regression using highly correlated column (i.e., w_bpfaced) out of all numerical columns.
3) Multi linear Regression with both highly correlated columns in numerical and categorical(i.e, w_bpfaced, Dominent_playing_hand respectively).
4) Multi linear Regression with all columns.

- Logistic Regression

A statistical method called logistic regression is used to predict the probability of a binary outcome by taking into account multiple predictor variables. This type of regression analysis is specifically designed for scenarios in which the result can fall into one of two groups, such as pass/fail or yes/no. Many industries, including the social sciences, finance, marketing, sports and medicine, use logistic regression extensively. Important tasks like predicting customer attrition, evaluating credit risk, and diagnosing medical conditions are performed with its assistance. Because of its simplicity, ease of comprehension, and efficacy in tasks where the results are binary, it is regarded as a foundational tool in machine learning and statistics.

Following are our approaches for developing logistic regression models

1) Multinominal regression using all categorical features.
2) Logistic regression using highly correlated column (i.e., Dominent_playing_hand) out of all categorical columns.

- Decision Tree

Decision Tree is a very popular model used for classification and regression. It is a non-parametric supervised learning method. The main reason we decided to run our data in this model is, it can handle both numerical and categorical data. Similarly, it can also handle multi-output problems and we can tune the hyper parameter to make our model better [1].

- Random Forest

Random Forest tree is another classification method which has better generalization performance, robustness to overfitting and improve accuracy (GeeksforGeeks, 2024). This can handle big data set as well so we choose this model to run the prediction [3].

```
#training the data set to 70% train and 30% test
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size =0.3, random_state=20)
x_train
```

Fig. 10. Splitting data for testing and training

```
print(grid.best_params_,grid.best_score_)
{'max_features': 5, 'n_estimators': 150} 0.7680327868852459
```

Fig. 11. Best estimator for Random forest tree and its accuracy

For running the Random Forest tree, we kept the data set same as Decision tree. First checked the NaN and null value. Second kept the same target variable and independent variables and third we split the data into train and test with 70% to 30% ratio. The one thing that we did different for this model is, we hyper tuned the parameter even before we ran the model. After the hyper tuning the parameter we got the best estimator as 'max_features': 5, and 'n_estimators': 150

VIII. DATA VISUALIZATION AND RESULTS

A. Results for Linear Regression

1) Multiple regression using all numerical features
   Mean Squared Error: 0.171
   R-squared: 0.314
2) Linear regression using highly correlated column (i.e., w_bpfaced) out of all numerical columns
   Mean Squared Error: 0.203

R-squared: 0.186

3) Multiple regression with both higly correlated columns in numerical and categorical(i.e, w_bpfaced and Dominent_playing_hand respectively)
   Mean Squared Error: 0.21
   R-squared: 0.158
4) Multiple regression with all columns
   Mean Squared Error: 0.198
   R-squared: 0.208

B. Results for Logistic Regression

1) Multinominal regression using all categorical features
   Accuracy-0.52
   Mean absolute error-0.48
2) Logistic regression using highly correlated column (i.e., Dominent_playing_hand) out of all categorical columns
   Mean Absolute Error: 0.472

From all regression models we found out that the model developed using all numerical columns have low mean squared error which indicates it is good model compared to all other models. So we decide to select the regression model which developed using all numerical columns Now our focus is to tune this model by adjusting some hyper parameters. [5]Now we need to tune the hyperparameters for the model that gives low mean square error i.e., model developed using all numerical columns. [] Here our focus is to find the best alpha values for lasso and ridge regression using grid and random search. We use alpha as 0.1, 1, 10, 20 and 30. Out of all these alpha values we got lowest mean square error for alpha equal to 0.1 using lasso regression (L1 regularization). [4]

Metrics	Before parameter tuning	After parameter tuning
Mean Square Error	0.1714	0.1634
R-Square Error	0.3137	0.3457

Fig. 12. Results comparison before and after hyper parameter tuning

C. Results for Decision tree

We ran the model in decision tree using DecisionTreeClassifier in jupyter note book. After the model is ran successfully we took its Mean square error and R2 to check its model performance. The MSE came out 0.32 and R2 came out -0.28 which means on average, the squared difference between the actual and predicted values is relatively high. Also the result says my model is performing poorly and may be worse than simply predicting the mean of the dependent variable.

The result wasn't good for our model which could be because of our wrong parameters. To get the best parameter we ran RandomizedSearchCV from sklearn. With this function it gave the best estimator for our data as criterion='entropy', max_depth=5, max _features='5'. With this optimized hyper parameter it game us the accuracy of 73% in Train data and 69% in the testing data.

```
from sklearn.metrics import mean_squared_error, r2
y_pred = model.predict(X_test)

mse = mean_squared_error(y_test, y_pred)
r2 = r2_score(y_test, y_pred)

print("Mean Squared Error (MSE):", mse)
print("R-squared (R2):", r2)
```

```
Mean Squared Error (MSE): 0.3173996175908222
R-squared (R2): -0.27669774418399484
```

Fig. 13. Model accuracy of decision tree without hyper parameter

```
print (f'Train Accuracy - : {model_rf.score(X_train,y_train):.3f}')
print (f'Test Accuracy - : {model_rf.score(X_test,y_test):.3f}')
```

```
Train Accuracy - : 0.734
Test Accuracy - : 0.694
```

Fig. 14. Accuracy test after optimizing hyper parameter

D. Results for Random forest

The model with max feature 5 and estimator 180 gave the accuracy of 77%. This is the best accuracy so far among all the model we have Run. In below Fig we can see that the area highlighted with yellow provided the highest accuracy.

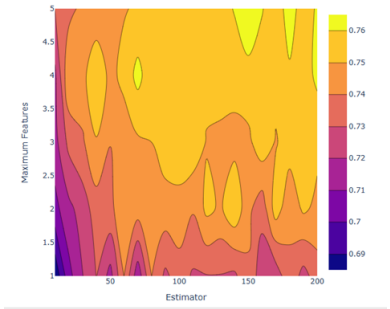

Fig. 15. Hyper parameter tuning

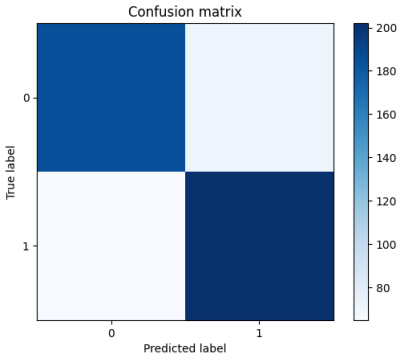

Fig. 16. Confusion matrix plot

This time for the performance test we using confusion matrix. Confusion matrix had provide

True positives as 185

False positive as 71

False negatives as 65 and

True negatives as 202

with accuracy of 74%

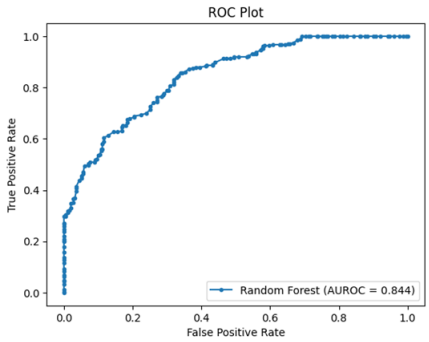

Fig. 17. ROC Plot for Random forest model

For the performance check of the model we ran the ROC plot. It gave the AUROC value of 84% which is a very good value to attend. The ROC plot proves

- The model has good discriminatory power
- The model's performance is significantly better than random chance, but it may not be perfect
- And it is a good model that has a relatively good ability to distinguish between the two classes.

IX. CONCLUSION

After evaluating multiple models with our dataset to achieve the most accurate predictions for match results, we found that Random Forest emerged as the top performer. It boasted an impressive accuracy rate of 76% and demonstrated strong performance with an 84% score. Notably, Random Forest effectively addressed issues like over-fitting and outliers, outperforming other models in this regard. It's worth highlighting that our model surpassed the Glicko model proposed by Yue, Chou, Hsieh, and Hsiao (Yae, 2022) for predicting tennis match outcomes. Leveraging insights from their work titled "A Study of Forecasting Tennis Matches via the Glicko Model," we prioritized robust feature selection and further enhanced performance by optimizing hyper parameters. This strategic approach led to our model's superior predictive capability. In summary, by prioritizing the inclusion of impactful features, employing sophisticated feature engineering techniques, and continually refining our models through iterative optimization, we can strive towards achieving even better results in predicting tennis match outcomes in the future.

Model	Accuracy
Decision Tree	72%
Random Forest tree	76%

REFERENCES

[1] Ch Sai Abhishek, Ketaki V Patil, P Yuktha, KS Meghana, and MV Sudhamani. Predictive analysis of ipl match winner using machine learning techniques. *Int. J. Innov. Technol. Explor. Eng.*, 9(2S):430–435, 2019.

[2] Rahul Baboota and Harleen Kaur. Predictive analysis and modelling football results using machine learning approach for english premier league. *International Journal of Forecasting*, 35(2):741–755, 2019.

[3] Rory Bunker and Teo Susnjak. The application of machine learning techniques for predicting match results in team sport: A review. *Journal of Artificial Intelligence Research*, 73:1285–1322, 2022.

[4] Sophie Chiang. Machine learning for table tennis match prediction. *arXiv*, 2023.

[5] Alexander De Seranno. *Predicting Tennis Matches Using Machine Learning*. PhD thesis, Ghent University Ghent, Belgium, 2020.

[6] Zijian Gao and Amanda Kowalczyk. Random forest model identifies serve strength as a key predictor of tennis match outcome. *Journal of Sports Analytics*, 7(4):255–262, 2021.

[7] Wei Gu and Thomas L Saaty. Predicting the outcome of a tennis tournament: Based on both data and judgments. *Journal of Systems Science and Systems Engineering*, 28:317–343, 2019.

[8] Corentin Herbinet. Predicting football results using machine learning techniques. *MEng thesis, Imperial College London*, 2018.

[9] Lars Magnus Hvattum and Halvard Arntzen. Using elo ratings for match result prediction in association football. *International Journal of forecasting*, 26(3):460–470, 2010.

[10] Anito Joseph, Norman E Fenton, and Martin Neil. Predicting football results using bayesian nets and other machine learning techniques. *Knowledge-Based Systems*, 19(7):544–553, 2006.

[11] Darwin Prasetio et al. Predicting football match results with logistic regression. In *2016 International Conference On Advanced Informatics: Concepts, Theory And Application (ICAICTA)*, pages 1–5. IEEE, 2016.

[12] Michal Sipko and William Knottenbelt. Machine learning for the prediction of professional tennis matches. *MEng computing-final year project, Imperial College London*, 2, 2015.

[13] Sascha Wilkens. Sports prediction and betting models in the machine learning age: The case of tennis. *Journal of Sports Analytics*, 7(2):99–117, 2021.

[14] Jack C Yue, Elizabeth P Chou, Ming-Hui Hsieh, and Li-Chen Hsiao. A study of forecasting tennis matches via the glicko model. *Plos one*, 17(4):e0266838, 2022.