Predictive Machine Learning Model for Optimal Player Selection in Fantasy Cricket: Maximizing Points and Winning Fantasy Premier League

1. ABSTRACT

Fantasy sports have recently gained popularity on a global scale, enthralling fans and bringing to the creation of a plethora of platforms devoted to this online gaming activity. The exciting challenge of selecting a virtual squad made up only of real-life sportsmen is at the heart of this appeal, setting the stage for competitive fan battles whose teams' fortunes reflect the real-time performances of their preferred players. However, manually selecting an ideal fantasy squad is a really difficult undertaking needing in-depth understanding of cricket and substantial study. The objective of this project was to use machine learning and optimisation techniques to create an automated, data-driven method for selecting fantasy cricket teams. Historical Indian Premier League (IPL) match data is obtained with relevant features. Numerous exploratory data analyses were carried out to comprehend feature distributions and relationships with the target variable of fantasy points obtained after this data had been cleaned and duplicate values had been checked. Different machine learning approaches, including but not limited to Linear Regression, Random Forest, CatBoost were investigated in order to build a predictive model to forecast player fantasy points whose in depth description can be found in the subsequent sections of the dissertation. The Root Mean Squared Error (RMSE) metric was used to evaluate each prediction model's performance after rigorous training and validation. The best model (Weighted Average) obtaining a test RMSE of 6.4114. Even though we will explain why we ultimately chose CatBoost as our final model later on in our dissertation. Runs scored and wickets taken were identified as the top predictors using a feature importance analysis, demonstrating the validity of the model. Utilising predictive analytics to evaluate individual player performances is the first phase in this multi-part, complicated research. In essence, the player point forecasts were provided as raw data to the constructed linear programming model. The primary objective was to create an 11-player fantasy squad with a captain and vice-captain with the intention of maximising predicted fantasy points. An integer linear programming model was developed using Python's PuLP library, adhering to the rules established by the well-known fantasy cricket platform Dream11. These rules encompass a cap on players from a single team, the inclusion of players with a variety of roles, and tight budget constraints. The method that emerges suggests an ideal 11-player team that is predicted to score a total of 687.18 points and is based on predictive analytics and skillful optimisation techniques. The captain and vice-captain, two vital positions in fantasy cricket that are essential for gaining extra points, are included in this team. By reading the dissertation in further detail, one may learn about the many factors that went into these choices, including the subtleties of cricket and the crucial part that data science plays in sports analytics.

2. INTRODUCTION

In 2003, T20 Cricket was first introduced in England. It gained huge popularity because to its concise structure. T20 entered India as well because of its penchant for high-octane action. 2008 saw the start of the Indian Premier League (IPL), a 20-20 cricket competition organised by the BCCI. The IPL T20 cricket competition has been put on by BCCI annually [1]. The world of sports has experienced a paradigm shift as a result of the emergence of sports analytics. Following the publication of "Moneyball: The Art of Winning an Unfair Game" a book by Michael Lewis [2] in 2003, the term "Sports Analytics" became widely used in the fields of sports and statistical analysis. The arrival of the movie “Moneyball [3]" in 2011 contributed to its popularity's growth. "Moneyball [3]" in 2011. In the movie "Moneyball," the general manager of the Oakland Athletics, Billy Beane, a former baseball player himself, employs analytical techniques to glean insights from the Player's historical statistics and makes crucial decisions for his club, the Oakland A's. To build a competitive baseball club on a tight budget, he heavily depended on the use of analytics, and his team broke all previous records for winning 20 consecutive games in the history of Baseball. Because of the revolutionary success of using analytics in sports to make judgements, individuals have started using them not only in Baseball but in all of the main, wildly popular organised sports, including Basketball, American Football, Soccer, Tennis, and Cricket [4]. Innovative digital platforms like Dream11 have developed in the middle of this sports analytics revolution, turning passive cricket viewing into an engaging activity. With more than 11 crore users participating in fantasy sports for a variety of sports, including basketball, hockey, football, kabaddi, and fantasy cricket, Dream11 is India's largest fantasy sports website [5]. With a worth of more than USD 1 billion, it was founded in the year 2012 and has since become the first Indian gaming firm to join the "Unicorn Club" [6]. The business charges a nominal fee for customers to take part in fantasy games, to put it simply. It creates a corpus with the money from membership fees paid by users and utilises it to pay out prizes to the users who create the most successful teams. Given a budgetary limit, a user must assemble his own squad for a forthcoming match. From a pool of players picked from both sides competing in the match, he or she must choose players, with each player incurring a set cost. Each team must provide a certain minimum number of players. There must also be adherence to the limitations on the number of wicketkeepers, bowlers, all-rounders, and batters [6]. Importantly, a team can have a maximum of 7 players from the same competing side. Each player receives a score at the conclusion of the game based on how well they performed in the match after being thusly chosen. The person who's fantasy squad receives the highest scores wins the cash reward. Some individuals believe that fantasy sports are really a sophisticated form of gambling, which is prohibited in India. The legitimacy of the fantasy sports industry has been contested in court cases, but the Supreme Court of India rejected the cases, ruling that betting and fantasy games cannot be regarded equally since they demand different levels of analytical and subject-matter expertise [6]. Making the ideal Dream11 team, as alluring as it may sound, is no easy task. It necessitates extensive research, acute analytical abilities, and a thorough comprehension of the sport. In order to tackle this problem, this dissertation develops a data-driven method for choosing optimal fantasy cricket teams by utilising machine learning and optimisation approaches. Historical IPL(Indian Premier League) match data was obtained with features like Individual_Match No, Player, Team, Opp_Team , RH/LH, Match_Type, Ground, Role, Runs, Wickets, Dream 11_ Points, Dismissal, Playercost. The dataset got 1320 columns and 13 rows. After cleaning and checking for duplicate values various exploratory data analysis methods like univariate analysis, bivariate analysis, analysis based on roles and players, analysis based on teams and checking external factors impact were done. Before moving onto modelling several data pre-processing steps were involved like scaling numeric features, one hot encoding of categorical columns and binary encoding. . Different machine learning approaches, including Linear Regression, Random Forest, CatBoost, Weighted Average, Stacked Ensemble, XGBoost, LightGBM, and others, were investigated in order to build a predictive model to forecast player fantasy points. Recognizing the power of collective decision-making, two voting ensemble combining the strengths of multiple models were also constructed and evaluated. The Root Mean Squared Error (RMSE) metric was used to evaluate each prediction model's performance after rigorous training and validation. Hyperparameter tuning was also done to maximize model performance. Among these algorithms, the models CatBoost, Random Forest, Weighted Average, Voting Ensemble (Random Forest + CatBoost), and Voting Ensemble (Random Forest + CatBoost + Linear Regression) all showed promising results. With the best model (Weighted Average) obtaining a test RMSE of 6.4114. Even though we will explain the rationale behind using CatBoost as our final model later. An optimisation model was created using Python's PuLP package, and player fantasy point projections were then included. By using an integer programming approach, it was possible to choose a fantasy team that would maximise predicted points while still adhering to the limitations imposed by actual fantasy platform regulations. There were restrictions on the budget for selecting players, the number of players who could make up a squad, and the size and makeup of the team. After taking into account all restrictions, the model was able to forecast a final dream 11 team with an estimated point total of 687.18. It was also properly determined who would be the captain and vice captain. The necessity for arbitrary human input in team selection was removed by the combined prediction-optimization technique. This study shows evidence of the mutually beneficial interaction between sports and data science. It gives insights into the mechanics of cricket player performances in addition to meeting the needs of the increasing fantasy sports community . This research lays out a road map for similar initiatives in other sports by combining predictive analytics with optimisation.This introduction is followed by a thorough literature analysis that explains the theoretical foundations of the techniques and the historical setting. The complexity of the data, the prediction models, the process of optimisation, and the results are thoroughly examined in later chapters. The discussion, future work and conclusion signal the end. A thought-provoking fusion of sports, data analytics, and optimisation is promised in the subsequent story. The intricate network of cricket, its players, and the broad framework of data science will begin to fall apart as we go along, revealing a wealth of new information.

3.LITERATURE REVIEW

Muthuswamy and Lam [7] forecasted how Team India's current bowlers will perform in 2008 against the top seven international cricket teams. They made predictions about the number of runs and wickets an Indian bowler will likely give up in an upcoming ODI match. Iyer and Shard [8] demonstrated how to classify the bowlers and batsmen using neural networks to anticipate player performances. He independently classified them into the three ideal categories of performer, moderate, and failure. By contrasting and evaluating the advantages and disadvantages of the two competing teams, Jhanwar and Paudi [9] foresaw the result of a cricket match. Mukharjee [10] rated the performance of bowlers and batters in teams by using social network analysis. Based on a player's popularity on Twitter, he identified three possible outcomes for their performance: High, Average, and Low. The author attempted to establish a link between popularity and accomplishment, which is intriguing but impractical. [11] provided an example of how regression may be used to forecast continuously valued variables, such as runs scored, wickets taken, and so on. Other dependent variables are used, such as the opposition's performance, the location, the innings, etc. With the use of a backpropagation network and radial basis network function, [12] provided an integer optimisation technique for selecting a team of 11 players based on a variety of considerations, including limits on the team's budget, role, and other variables. The user's selected degree of risk was taken into account when developing a system. The skill level of each participant was indexed (graded) through a method of integer optimisation that was created by [13]. Separately created from the elements previously existing in the data were the bowling, batting, and all-rounder indexes. The goal of [14] was to precisely forecast how many Fantasy Premier League points each football player will accumulate during the course of the season. To do this, three machine learning models were contrasted: Linear Regression, Decision Tree, and Random Forest. The greatest performance of the three models was by the Linear Regression model, demonstrating how well it predicts Fantasy Premier League points. By verifying three models that may be used to make predictions, the paper adds to the body of work already done in this field. The accuracy of the models has been evaluated and their validity tested using historical data going back to the 2016–17 season. [4] examines the application of regression and classification algorithms to forecast player performance in fantasy cricket games played on the Dream-11 Fantasy Sports Platform (FSP). In their early research, the authors used multiclass classification algorithms such Naive Bayes, Random Forest, Multiclass SVM, and Decision Trees to experiment with classification models to predict the performance of batsmen and bowlers. They discovered that regressor models, however, offered more accurate forecasts for player performance. Regressor models, as opposed to categorization models, offer more accurate findings, the authors determined, making them a better method for forecasting player success in fantasy cricket games. The ideal lineup of 11 players for the fantasy cricket team was chosen by the authors using a mix of the Greedy and Knapsack Algorithms.The Extra Trees Regressor Model (ETR) was shown to be the best regressor method by using the PyCaret Python Library to make accurate predictions. Incorporating objective and subjective aspects of the game, the authors additionally used the Plotly Python Library to visualise team and player performances. The authors' prediction methodology boosted the likelihood that fantasy cricket players on Dream-11 FSP would score big. The most important physical and technical performance factors associated with team quality in the Chinese Super League were found using multinomial logistic regression by [15]. Recent studies on regression techniques in sports performance can be found in [16], [17], [18], [19], etc. [20] predicts fantasy points using regression models and uses python’s Pulp library to build optimized dream11 team adhering to all constraints. [21] applies linear programming to build optimized team in fantasy cricket.

3. METHODOLOGY

The detailed data science procedure used to create a machine learning model for forecasting Dream11 fantasy cricket points and integrate it with optimisation to choose the best fantasy team is described in this section.

4.a) DATA COLLECTION AND PREPARATION

The dataset used in this project was sourced from [22]. The original name of the dataset was IPL_2020_Daily_data.csv which we changed to IPL. The dataset description was given as The players' Dream 11 Fantasy points for the IPL 2020 season are included in the dataset along with Matchwise statistics for each and every player. The player's runs scored, wickets taken, and other independent factors are included in the dataset. These features are useful for advanced analytics and predictive modelling [22]. The original dataset had an empty useless column in the beginning as the first column which we have removed manually. The dataset did not contain 'Playercost' column which is a very crucial feature for our final task of predicting an optimal dream11 team following all constraints so we made an additional column 'Playercost' . We obtained the playercost of 90% of the players in IPL 2020 season from ipl_squad_points.csv dataset available at [23]. For the remaining 10% of the players we used web scrapping to get their costs. We created an additional column in our IPL(original name - IPL_2020_Daily_data.csv) dataset named 'Playercost' and input all values obtained through ipl_squad_points.csv dataset and web scrapping to make our final dataset ready. The final dataset in which we worked contained all these features :

1) Individual Match No : Team wise match number (Number of Matches a team played) represented with prefix 'M'.

2) Player : Name of Players.

3) Team : Name of team 'Player' belongs {There are 8 Teams => 8 Unique values}.

4) Opp Team : Opposition Team (The team against which the player is playing).

5) RH/LH : Whether the Player is Right Handed(RH) or Left Handed (LH).

6) Match Type : Chasing or Defending (Defending : Batting first and setting up a target score for Opposite team, while Chasing is chasing the Defending team's target score).

7) Ground : Stadium, where the Match was played (There only 3 Grounds available => 3 Unique Values).

8) Role : Players Role in the Team (BAT (Batsman), BALL (Bowler), WK (Wicket-Keeper), ALL (All-Rounder) => 4 Unique Values).

9) Runs : Runs scored by the Player (Missing Values in the Runs scored Column states that the player did not get the chance to bat).

10) Wickets : Wickets Taken by the Player (Missing Values in the Wickets scored Column states that the player did not get the chance to BOWL).

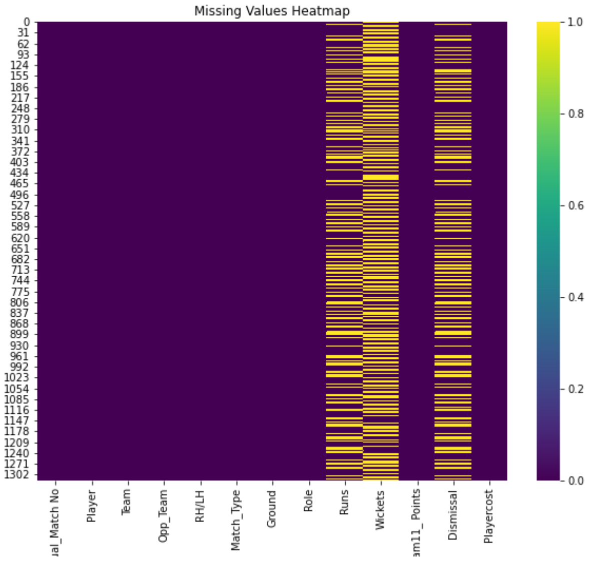
11) Dream11 Points : Fantasy Points earned by the Player as per Dream11 Rules and Regulations.

12) Dismissal : Mode of Dismissal. How the player got OUT !

13) Playercost : Cost to buy player for fantasy team.

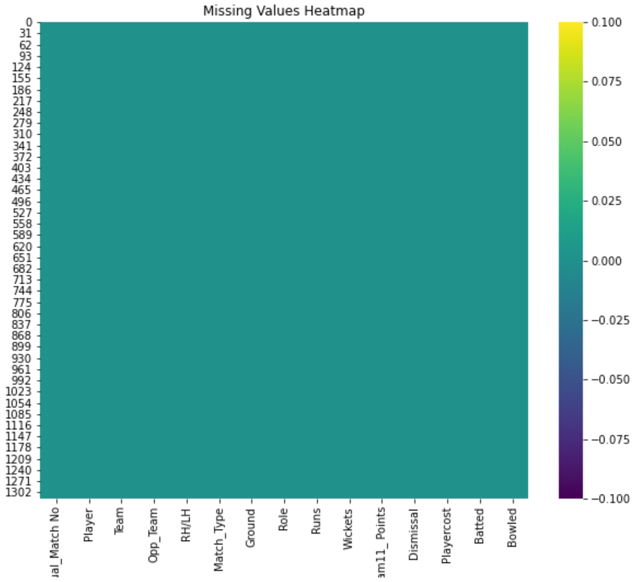
The data had 1320 rows and 13 columns.

Before jumping onto exploratory data analysis there were several steps needed to be taken care of the data. While checking statistical summary of numeric columns we could not find wickets, after checking we found out that it's datatype is showing as object and there were empty strings containing only whitespace in wickets column which we replaced with NaN after this we converted wickets column to numeric and were able to check its statistical summary. We checked statistical summary of categorical columns as well and while we got 0 duplicate values there were 444 null values in runs column, 642 null values in wickets column and 447 null values in dismissal column. **FIGURE-1**



In order to take care of the null values we followed these steps :

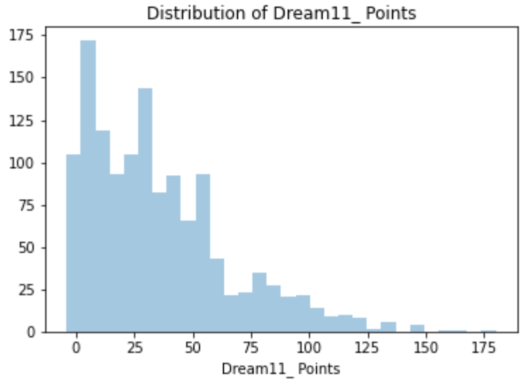
The dataset was prepared for analysis by handling missing values in a number of different ways. Based on whether a player's Runs or Wickets columns contained non-null values, two indicator columns, "Batted" and "Bowled," were developed to indicate if a player had batted or bowled in a game, respectively. When a player did not bat, their dismissal mode was automatically filled with the phrase "Did Not Bat." These records were revised in line with the assumption that a player remained "Not Out" in situations where runs were scored but dismissal information was unavailable. To signify that a player did not score any runs or take any wickets during that match, missing values in the Runs and Wickets columns were replaced with 0. All missing values were handled correctly with pertinent imputation following these procedures in data preparation to allow for further analysis. No more missing data were found, according to validation tests. The cleaned dataset was now prepared for further exploratory analysis and modelling with no problems with the data's quality.



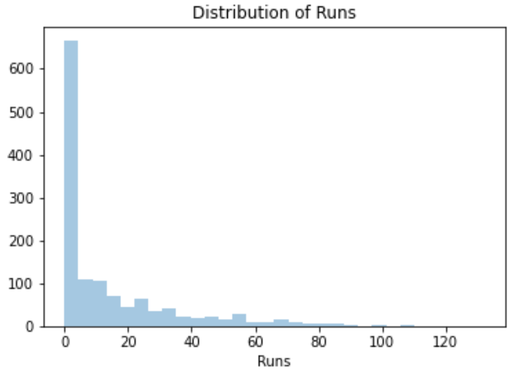
4.b) EXPLORATORY DATA ANALYSIS

It's critical to comprehend and evaluate the data we're working with before diving into the machine learning models and optimisation strategies. Providing insights into data patterns, correlations, anomalies, and influencing variables, exploratory data analysis (EDA) offers this fundamental understanding. Data scientists utilise exploratory data analysis (EDA) to examine and analyse data sets and summarise their key properties, frequently using data visualisation techniques. It makes it simpler for data scientists to find patterns, identify anomalies, test hypotheses, or verify assumptions by determining how to best modify data sources to obtain the answers they need. A greater knowledge of the variables in the data set and their connections to one another is provided by EDA, which is generally used to investigate what data may disclose beyond the formal modelling or hypothesis testing activity. It can also assist in determining whether the statistical methods we are thinking about using for data analysis are acceptable [24].

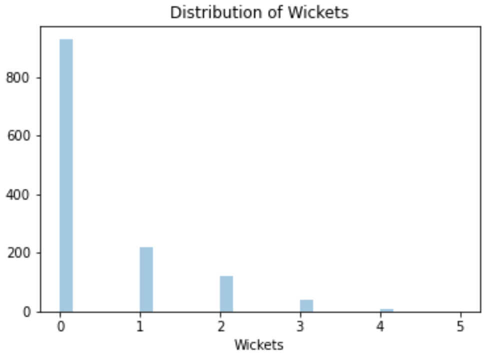
Univariate Analysis



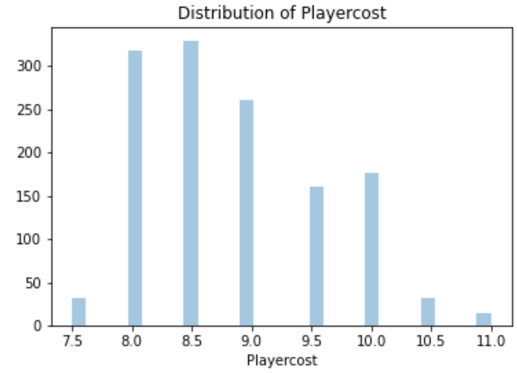
Average dream11 points is around 35.64 (but there is large standard deviation of 30.47). So players can score anywhere from a few points to 60-70 points typically. 75% of players score upto 52 points with half of all players scoring 29 points or below. Top 25% of the players scored above 52 points. Data is widely spread and indicates a range of dream11 points. Blocks decreases as we move from left to right so data is right skewed meaning most players score low or average and there are very few who outperform others (150-175) dream11 points.



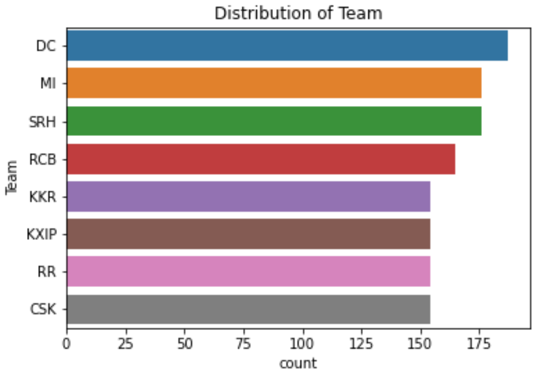
Half of the players score below 14 runs and half above that. Average number of runs scored is approximately 21.13. There is sd of 22.14 meaning there is considerable fluctuation about runs scored by players in different matches. Highest recorded score is 132 runs. 75% of the players score upto 31 runs or less and 25% of the players score only upto 4 runs.



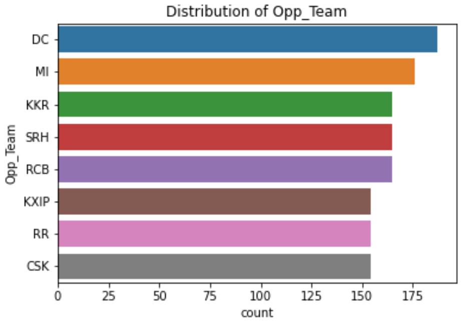
A significant number of bowlers does not manage to take any wicket. Maximum wickets taken is 5. 75% of players take 2 wickets or fewer. 25% of players do not manage to take any wickets. 50% of the players take at least 1 wicket while the other half does not get wickets. Standard deviation (Sd) of 0,976 approx suggests that there is reasonable amount of variation in the wickets taken by players across matches.



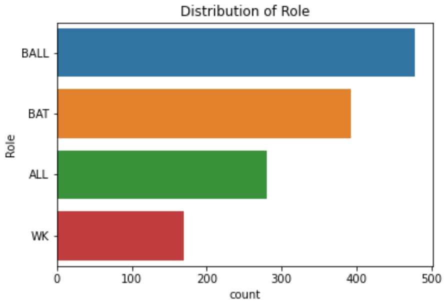
Player value is 8.85 on average. This implies that the average player is reasonably priced, perhaps providing fantasy league players with good value for their money. The player expenses are subject to modest, but not much, variance, with a standard deviation of 0.77. This suggests that the price structure may be moderately organised and not entirely random. The concentration of players in and around this price range is shown by the median value of 8.5, which is extremely close to the mean. A quarter of the players are valued at 8 or less. In contrast, the top quartile of the scale is at 9.5, which denotes that three-quarters of the players are evaluated at that level or lower. Players that are likely to transform the game, have consistently outstanding performances, and may have a sizable fan base are have the highest cost of 11. Overall, the Playercost distribution suggests a certain prudence in the pricing plan, with the majority of players being modestly priced. The few players with costs close to the top maximum of 11 are likely to be the standout performances, and even if they are expensive, they might be essential to winning fantasy leagues.



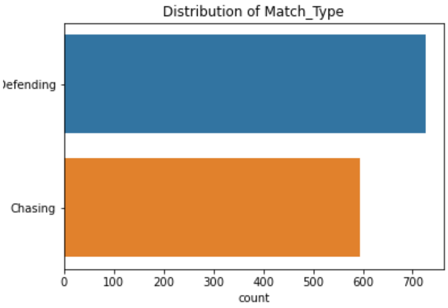
This means players of DC are more present in this dataset followed by others and CSK being the lowest.



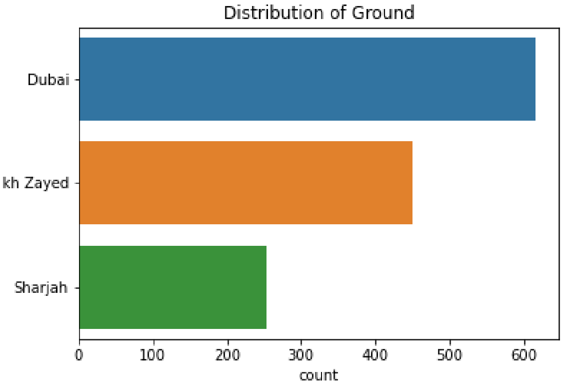
For distribution of opposite teams as well we can see the exact same scenario.



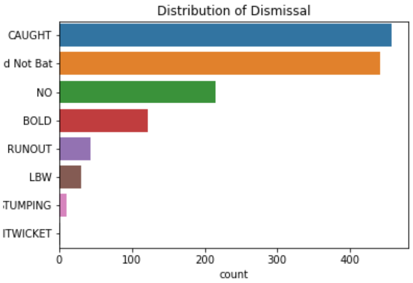
The dataset consists of more bowlers than any other role. Wicketkeeper role is the least represented in the dataset which is obvious because there are more batsmen and bowlers playing cricket than wicketkeepers and all-rounders.



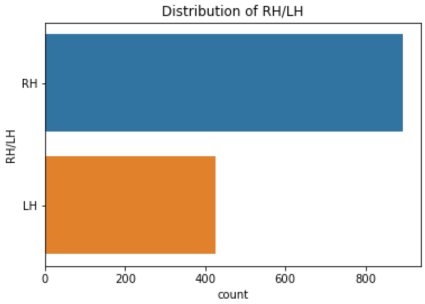
Players in defending scenario are more frequent in the dataset than in the chasing scenario.



Most matches were played in Dubai ground followed by kh Zayed and Sharjah.

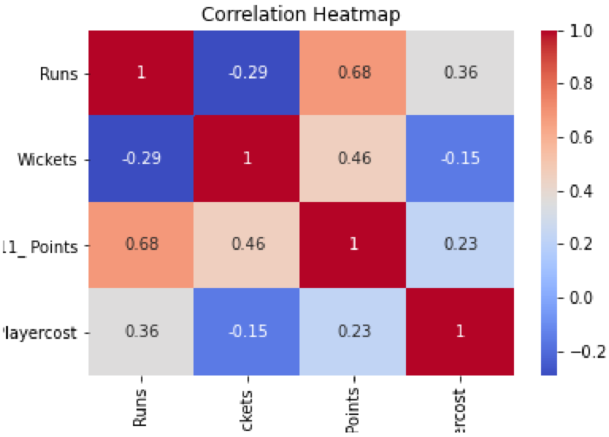


Caught is the most frequent occuring mode of dismissal in the dataset and the least one is hitwicket.



Right handed players are more present in the dataset than left handed players.

Bivariate Analysis



Runs and Wickets (-0.29): In general, players who score more runs tend to take less wickets, and vice versa, according to this negative correlation between runs and wickets (-0.29). Given that top-order batsmen are less likely to bowl and bowlers often score fewer runs, this makes basic sense.

Runs and Dream11 Points (0.68): Runs and Dream11 Points Correlation (0.68), which is very positive, shows that a player's run total has a considerable influence on their Dream11 point total. Batsmen who constantly score are very significant assets in a fantasy squad because when a player scores more runs, his fantasy points often rise.

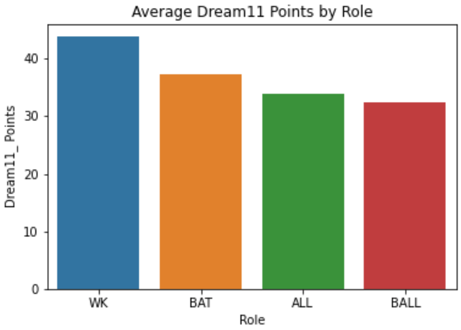
Runs and Player Cost (0.36): The correlation indicates that players who score more runs often have a higher cost in fantasy sports. The correlation suggests a mediocre connection between runs scored and player value, notwithstanding its weakness.

Wickets and Dream11 Points (0.46): A moderately positive connection means that a player's Dream11 point total is significantly impacted by the wickets he takes. In games where bowling is favoured, bowlers or all-arounders who take wickets are useful for fantasy teams since they are likely to earn more fantasy points.

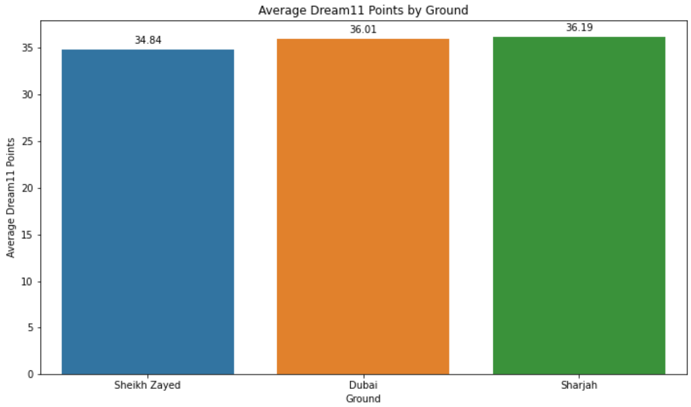
Wickets and Player Cost (-0.15): The weakly negative correlation between wickets taken and player cost (-0.15) indicates that there isn't a significant connection between the two variables. It does, however, give the impression that players who record more wickets may have a lower fantasy price. This could be as a result of other aspects of player value, such their batting average or general level of popularity.

Player Cost and Dream11 Points (0.23): Players that earn more Dream11 points are typically more expensive, according to this positive association between Dream11 Points and Player Cost (0.23). Although performance (as measured by Dream11 points) does impact player cost, the link isn't very strong, suggesting that other factors, such as player reputation may also be at play.

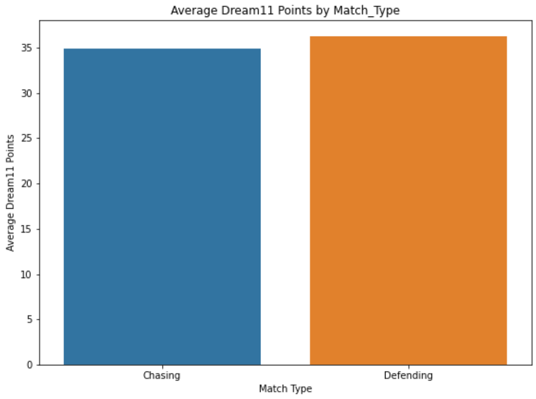
For those who participate in fantasy sports, knowing these relationships is crucial since it offers insights into how various elements of a player's performance could effect their fantasy points and worth. As an illustration, given their strong positive association with Dream11 points, concentrating on high-scoring batters and wicket-taking bowlers or all-rounders may be a successful tactic. But to fully understand the intricacies of player cost, a more thorough study must be conducted, taking into account both internal and external variables.



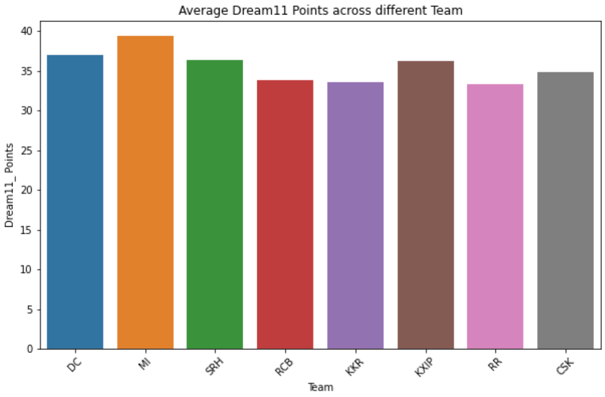
The bar graph offers perceptions on how various cricket team roles performed. In terms of average Dream11 points, the wicketkeeper (WK) position stands out. This can be linked to the variety of tasks that a wicketkeeper is required to perform; in addition to participating in dismissals behind the wickets, they frequently play important roles as batsman. The wicketkeepers come first, followed by the batters (BAT), all-rounders (ALL), and then the bowlers (BALL). The dominance of batsmen and all-rounders may be a reflection of their steady contributions to the team in terms of running up runs and claiming wickets. Even though bowlers are crucial, their performances might vary based on the surface, the opposition, and other outside circumstances. When choosing their teams, fantasy sports players should take this distribution into account. Prioritising wicketkeepers and batters with a history of high scoring might be a tactic. However, to combat the unpredictable nature of matches, a balanced squad with a mix of all positions is still necessary.



The bar plot offers a visual comparison of the players' average Dream11 points over three different cricket grounds. Sharjah got the highest average dream11 points scored by players followed by Dubai ground and Sheikh Zayed.The disparities in average dream11 points between the grounds imply that each ground's unique external elements, such as the pitch's characteristics, the size of the boundaries, or even the presence of the crowd, may have an impact on how well players perform. When selecting players depending on the venue, this knowledge might be vital for fans of fantasy sports.

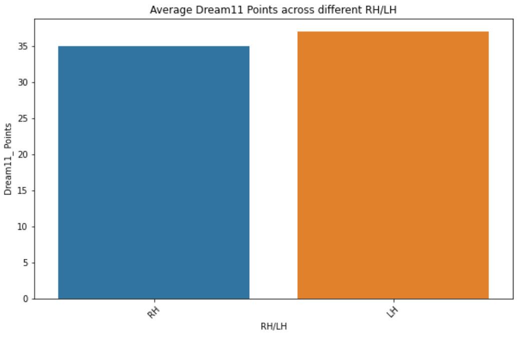


The bar plot illustrates how players' average dream11 points fluctuate depending on whether they are defending or chasing, giving viewers an understanding of how different match conditions could affect a player's success in fantasy sports. The average dream11 point was higher for players defending a score than for those chasing. This distinction shows that defending as a tactic can provide players additional chances to show off their abilities and get points. This could be explained by elements like the psychological benefit of having a score on the board or certain game dynamics that are present when defending. For instance, fielders may be in positions that provide more opportunities for catches or run-outs, while bowlers may have greater room to take wickets. Teams chasing a target, on the other hand, could experience more pressure, which could have an effect on their performance and, ultimately, their point totals. Fantasy sports fans need to be aware of these subtleties since the players they choose to play on their squad will often depend on whether they'll be chasing or defending.

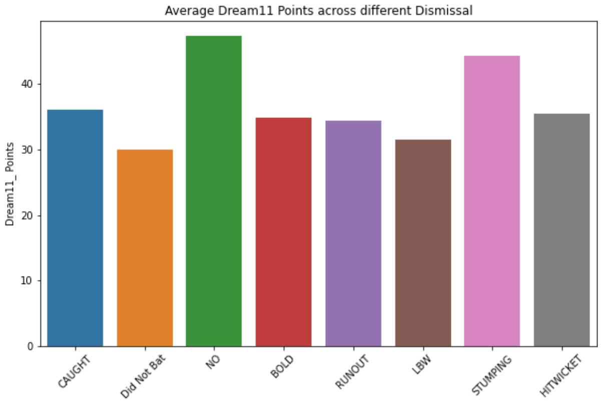


In the average Dream11 points that players from various IPL teams acquire, there is a noticeable difference shown by the visualisation. The squad whose players earned the most Dream11 points on average was Mumbai Indians (MI), which stands out. The fantasy points frequently reflect their good league performance, indicating a mix of reliable individual performances and overall team success. After MI, the players for the Delhi Capitals (DC) and Sunrisers Hyderabad (SRH) also put up fantastic displays, earning huge fantasy points. This may be a sign of a few great performances by key players or an indication of the team's general consistency throughout the season. While its players get the lowest average Dream11 points, the Rajasthan Royals (RR) are at the opposite end of the spectrum. This might indicate difficulties the team had during the season, whether they were brought on by injury, poor play, or other external reasons. It's important to note that the Kolkata Knight Riders (KKR) leans towards the lower end while Kings XI Punjab (KXIP), Chennai Super Kings (CSK), and Royal Challengers Bangalore (RCB) all fall into the mid-tier category.

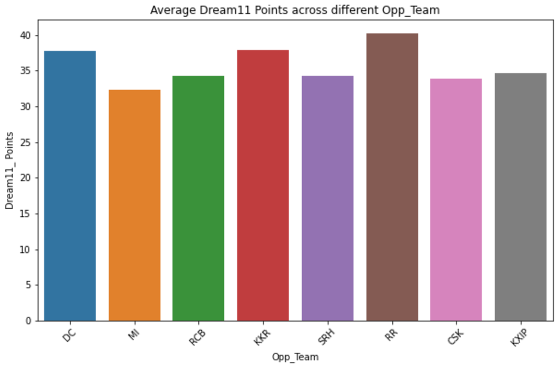
Fantasy sports players might gain insightful information from this study. Making smart decisions while assembling one's fantasy team can be aided by recognising the teams whose players frequently score high. It's important to keep in mind though that prior performance doesn't necessarily indicate how something will turn out in the future. Strategies must change as teams and players do, which happens frequently.



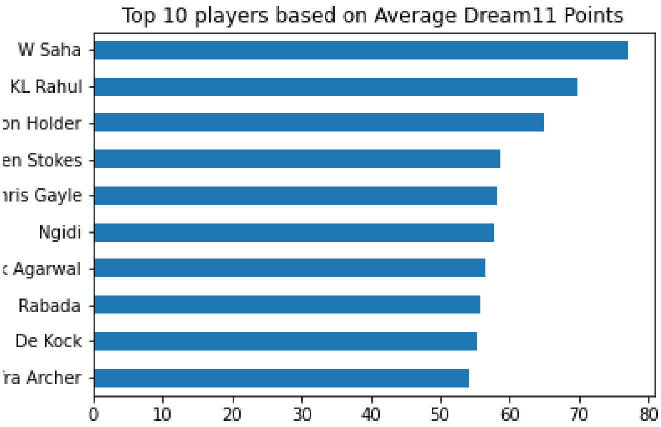
The bar graph compares how left-handed (LH) and right-handed (RH) players performed in terms of their average Dream11 points. It's interesting to note that left-handed players score somewhat higher than right-handed ones. This phenomena might be explained by a variety of factors, including the left-handed players' distinctive angles and approaches in both bowling and batting, which provide their opponents with a sophisticated challenge, or the fact that left-handed players are very uncommon, which makes them more valuable in specific match conditions. This shows that it would be advantageous for fantasy sports fans to mix players who are left- and right-handed equally. Even though they are less common, left-handers frequently provide a new dimension to the game in terms of execution and strategy.



A close examination of the dismissal method and its relationship to Dream11 points can be seen in the graph. The average point total is higher for players who haven't been sent out (labelled as "NO"). This is most likely a result of the fact that players who aren't removed frequently score more runs, leading to an increase in total points. The prominence of "Stumping" that follows emphasises the wicketkeeper's crucial function in the field. Traditional dismissal categories like "Caught," "Bowled," and "Run Out" have comparable average scores as we proceed lower in the order, indicating the typical gameplay style. Notably, 'Did Not Bat' receives a lower score, which is reasonable given that these individuals were unable to make an offensive contribution. To increase the likelihood of picking individuals less likely to get dismissed early or those who consistently contribute with the bat or ball, it is essential to take into account the players' batting positions and prior performances while making player selections.



The results of players when playing against other teams are shown in this graph in terms of Dream11 points. The greatest average point totals are amassed by players who play against RR, indicating that games against RR may be high-scoring or provide more opportunity for players to shine. Players generally earn the fewest points when playing against MI, making those matchups appear to be the most difficult. This may be a sign of MI's exceptional bowling and fielding skills, which limit the chances for scoring for the opposition. When choosing their squad, fantasy gamers may take into account these trends, giving players who will be facing teams that they have traditionally scored more against additional importance. The present makeup of players and teams must constantly be taken into consideration, though.



Using average Dream11 points, the horizontal bar graph displays the top cricketers. These guys stand out because they continuously gave excellent performances throughout the games. Their high averages show that they are capable of excelling in a variety of match situations in addition to their unique abilities. Every fantasy sports player has to keep an eye on these elite athletes. A fantasy team's chances of scoring highly might be significantly increased by having them on it. However, because sports performances might vary, it's equally important to monitor their present fitness levels and form.

4.c) DATA PREPROCESSING

Before jumping into the modelling part there were several preprocessing and feature engineering steps involved which we will discuss in detail in this section.

- **One Hot Encoding** - Categorical columns like 'Team', 'Opp_Team', 'Ground', 'Role', and 'Dismissal' were transformed using one-hot encoding. Depending on how they are implemented, certain machine learning algorithms, such decision trees, can function directly with categorical data, but the majority need all input and output variables to have a numerical value. Any categorical data must thus be converted to integers. Data can be converted using one hot encoding as a means of getting a better forecast and preparing the data for an algorithm. With one-hot, we create a new category column for each categorical value and give it a binary value of 1 or 0. A binary vector is used to represent each integer value. The index is denoted by a 1 and all values are zero. When there is no association between the data, one hot encoding is helpful. The order of the numbers is considered significant by machine learning algorithms. They will therefore see a larger number as superior or more significant than a lower one. Despite being useful in some ordinal contexts, this can cause problems with predictions and poor performance when input data lacks ordering for category values. One hot encoder steps in to rescue the day at that point. Our training data is improved by one hot encoding, which also makes it more adaptable and expressive. We are able to calculate a probability for our values more quickly when we use numerical numbers.[25]

- **Label Encoding** The columns "RH/LH" and "Match_Type" were label-encoded. "RH/LH" indicates a player's batting handedness, where Right-handedness (RH) is represented by 0 and Left-handedness (LH) by 1. For the "Match_Type" column, "Defending" is mapped to 0, while "Chasing" is mapped to 1. Since there are only two possible values of each of these columns we used label encoding. In order for machine learning models, which can only fit numerical data, to function, categorical columns must be converted into numerical ones using the label encoding approach [26].

- **Feature Scaling** - Continuous variables with a wide range of scale and magnitude are represented by the columns "Runs," "Wickets," and "Playercost." These columns were standardised using the Standard Scaler from scikit-learn to make sure that no one characteristic unduly dominates the model. By performing this process, each of these columns is changed to have a mean of 0 and a standard deviation of 1, making the dataset more suitable for algorithms that take feature scales into account. The Python pickle package was used to serialise and store the scaler once the Standard Scaler had been adjusted to the dataset. This step is crucial as the same scaling parameters need to be applied to any future data when deploying the model in a real-world scenario or during model validation. The values of features or variables in a dataset can be scaled using the data preparation technique known as "feature scaling." To prevent the dominance of features with higher values and to guarantee that all features contribute equally to the model, this is done. When working with datasets that contain features with varying ranges, units of measurement, or orders of magnitude, feature scaling becomes important. In such circumstances, the volatility in feature values may result in biassed model performance or issues throughout the learning process. Standardisation, normalisation, and min-max scaling are just a few of the typical methods used for feature scaling. The distributions and relative relationships between the feature values are maintained while being adjusted using these techniques. It is simpler to create precise and efficient machine learning models by applying feature scaling to the dataset's features, which scales them to a more uniform scale. Scaling allows for meaningful feature comparisons, enhances model convergence, and avoids certain traits from overshadowing others based just on their size. With a single standard deviation and values centred around the mean, standardisation is the scaling approach that we employed in our research. The attribute's mean becomes 0 as a result, and the distribution that results has a unit standard deviation. The formula is

$$X' = \frac{X - \mu}{\sigma}$$

where μ is the mean of the feature values.
X' is the standardized score
X is the original value

σ is the standard deviation of the feature values [27].

- **Feature Reduction and Player Mapping** - The "Player" column was duplicated and kept on its own as "player_mapping." Reverse mapping is therefore made possible, which may be beneficial for understanding the results and making predictions about certain players in the future.
  'Player' and 'Individual_Match No' columns were removed from the dataset. The justification for this is because individual match numbers and direct player names are nominal in nature and useless for predictive modelling. In their place, they can cause overfitting or add noise to the models. The technique of lowering the amount of features in a calculation that uses a lot of resources without losing crucial information is referred to as feature reduction or feature dropping [28].

  The dataset was prepared to be utilised for modelling when these data processing processes were finished. Model selection, training, validation, and assessment are the subsequent processes in the technique, which guarantee the reliability and accuracy of the forecasts obtained from the selected model.

4.d) MODELLING

Dataset Splitting

The dataset was divided into The target variable (denoted as y) and the features (denoted as X). The "Dream11 Points" serve as the study's target variable. All other dataset attributes are included in the features.

First, we divided the data into two sections in order to guarantee the generalizability of the models.

Training-Validation Set(80% of our data): We utilised a sizable portion of this data—the training-validation set, which makes about 80% of our data—to construct and fine-tune our models.

Test Set (20% of our data): This is set aside, and we used it at the very end to assess how well our model worked on unobserved data.

We next divide the large portion (Training-Validation Set) once again as follows:

Training Set: We utilised this section to train our model, which comprises 60% of the original data.

Before the final (Test Set), we used the Validation Set, which represents 20% of our original data to check and adjust our model.

So if we compare it to school then training set is like classwork where learning happens. The Validation Set is comparable to quizzes. It aids in better adjusting and preparation for final exam. The Test Set serves as the final exam. We may now assess our actual learning.

Models Used

To begin with, for our project we used a variety of regression models in the first step like

- Linear Regression
- CatBoost
- Random Forest
- KNN
- Decision Tree
- AdaBoost
- SVM

Though many papers have used classification models for predicting player performance some of the best papers like [14] have made regression models as their choice. [4] concluded that "Hence, with our product we proved that to predict the performance of a player in a prospective game, we must use regressor models and not classification models as suggested by our base paper". All this led me to make an informed choice of using regression models.

The training set was used to train each of the regression models and a validation set was used to verify it. In order to evaluate the model, the following performance measures were used:

coefficient of determination (R^2)

Root Mean Squared Error (RMSE)

Mean Absolute Error (MAE)

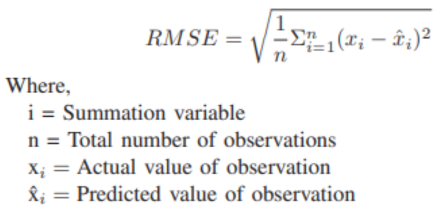
coefficient of determination (R^2): The ability of a statistical model to forecast an outcome is measured by the coefficient of determination (R^2). The model's dependent variable is a representation of the result. R^2 can take on a range of values between 0 and 1, with 1 being the maximum. Simply said, a model's R^2 will be closer to 1 if it is more accurate in making predictions [29].

The formula to calculate R^2 is

$$R^2 = 1 - \frac{SS_{res}}{SS_{tot}}$$

Where SSres is the residual sum of squares and SStot is the total sum of squares [14].

Root Mean Square Error(RMSE): The standard deviation of the residuals is represented by the root mean square error (RMSE). When compared to the data points on the regression line, the RMSE measures how dispersed the residuals are. To put it another way, it shows how closely the data falls along the line of best fit. The following formula is used to compute it:

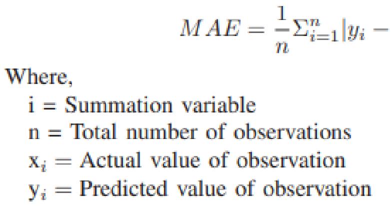


$$RMSE = \sqrt{\frac{1}{n}\Sigma_{i=1}^{n}(x_i - \hat{x}_i)^2}$$

Where,
i = Summation variable
n = Total number of observations
x_i = Actual value of observation
$\hat{x}_i$ = Predicted value of observation

[14].

Mean Absolute Error (MAE): This statistic measures the accuracy of a regression model. The average of each individual prediction error on every occurrence in the test set is the mean absolute error of a model with relation to that test set. For each occurrence, the prediction error is the discrepancy between the real value and the predicted value. Using the following formula, it is determined:



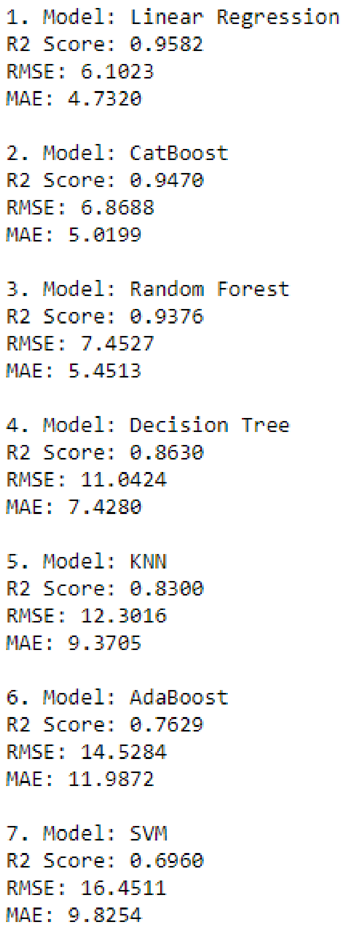
$$MAE = \frac{1}{n}\Sigma_{i=1}^{n}|y_i - x_i|$$

Where,
i = Summation variable
n = Total number of observations
x_i = Actual value of observation
y_i = Predicted value of observation

[14].

Based on each model's performance in relation to the three metrics, a composite rank was calculated.

We can see how our primary models performed and their rankings.



```
1. Model: Linear Regression
R2 Score: 0.9582
RMSE: 6.1023
MAE: 4.7320

2. Model: CatBoost
R2 Score: 0.9470
RMSE: 6.8688
MAE: 5.0199

3. Model: Random Forest
R2 Score: 0.9376
RMSE: 7.4527
MAE: 5.4513

4. Model: Decision Tree
R2 Score: 0.8630
RMSE: 11.0424
MAE: 7.4280

5. Model: KNN
R2 Score: 0.8300
RMSE: 12.3016
MAE: 9.3705

6. Model: AdaBoost
R2 Score: 0.7629
RMSE: 14.5284
MAE: 11.9872

7. Model: SVM
R2 Score: 0.6960
RMSE: 16.4511
MAE: 9.8254
```

Based on the primary evaluation the top 3 models turned out to be Linear Regression, CatBoost and Random Forest.

Linear Regression Model : A linear model is one in which it is assumed that the relationship between the input variables (x) and the sole output variable (y) is linear. More specifically (x), it can be shown that y can be calculated by combining the input variables linearly. When there is just one input variable (x), the process is known as simple linear regression. In statistical literature, when there are many input variables, multiple linear regression is the phrase employed. Given that the player's points might be influenced by a number of features, we employed the multiple regression model in this instance. Some of the independent variables are ground, role, runs, wickets etc. The dependent variable in this case is dream11 points [14].

CatBoost : The decision tree technique known as CatBoost, or "Category Boosting," makes use of gradient augmented decision trees. The prediction shift that happens during training is what it seeks to minimise. A succession of decision trees are built one after the other during training. Every new tree that is built has less loss than the ones that came before it. It employs the overfitting detector to prevent overfitting and, when activated, stops the development of trees [30]. The exponential increase in feature combinations is addressed by CatBoost by employing a greedy method for each additional split of the current tree. Every feature that contains more than one hot max size OHMS (an input parameter) is subject to the following actions by CatBoost:

1) Subdividing the records at random.

2) Taking the labels and turning them to integers

3) Transforming categorical features to numerical values.

The formula it uses to do so is

$$avg\_Target = \frac{count\_In\_Class + prior}{total\_Count + 1}$$

where count_In_Class denotes the total dream11 points for a given category feature value.

Total_Count is the number of earlier objects that have the same categorical feature value.

Prior is a constant that was created from the algorithm's starting parameters [31].

Random Forest : Random Forest Regression: Using the ensemble learning method, Random Forest Regression is a supervised learning technique for regression. To provide a more accurate forecast than a single model, the ensemble learning technique integrates predictions from many machine learning algorithms. A Random Forest builds a lot of decision trees during training, and each tree outputs the mean of the classes as its forecast. It is a strong and accurate model. It frequently succeeds on a variety of problems, including ones with non-linear connections. However, there are significant disadvantages, including the lack of interpretability, the potential for overfitting, and the need to select the amount of trees to incorporate in the model meaning tuning of the hyperparameters is required [14].

We decided to proceed further to hyperparameter tuning and later stages with our top 3 models Linear Regression, CatBoost and Random Forest.

Hyperparameter Tuning

The settings or knobs that may be adjusted before starting a training task to regulate how an ML algorithm behaves are known as hyperparameters. They can significantly affect how long it takes to train a model, how many resources are needed for the infrastructure (and how much that costs), how well the model converges, and how accurate it is. In contrast to hyperparameters, which are established before the training job is run and do not change throughout training, model parameters are learned as part of the training process. [32].

Based on our primary evaluation we selected CatBoost model and Random Forest for Hyperparameter tuning

By utilising 5-fold cross-validation and optimising for the negative mean squared error, GridSearchCV was used to look for the optimum hyperparameters. Cross validation was used to avoid overfitting.

GridSearchCV - The machine learning model is assessed using the GridSearchCV technique for a variety of hyperparameter values. Because it looks for the optimal collection of hyperparameters from a grid of hyperparameter values, this method is known as GridSearchCV. For instance, suppose we wanted to configure two hyperparameters of the Logistic Regression Classifier model, C and Alpha, with various sets of values. The optimal model will be created using the grid search approach by building several iterations of the model using all feasible combinations of hyperparameters.

The hyperparameters chosen for tuning in the two models were:

Random Forest – Number of estimators, Maximum features, Maximum depth, Minimum samples split, Minimum samples leaf.

CatBoost – Number of iterations, Learning rate, Depth.

The best hyperparameters for the models were:

Random Forest – [Number of estimators: 100, Maximum features: auto, Maximum depth: 40, Minimum samples split: 10, Minimum samples leaf: 1]

CatBoost- [Number of iterations: 200, Learning rate: 0.05, Depth: 4]

Final Model Training and Testing

Following the determination of the ideal hyperparameters, the models Linear Regression, CatBoost, and Random Forest were trained on a combined dataset (training + validation sets), and then evaluated on the test set.

We checked the feature importance for Random Forest and CatBoost models and the results were pretty obvious that runs and wickets these two columns were the most important ones.

The performance of the models on the test set were recorded and we investigated the performance of more models. We will discuss about the performance of all the models in detail in the Results and Analysis section.

Some more info about CatBoost and Random Forest readers can get from here [33]

Exploring More Models

We explored more models, trained them on training + validation sets and checked their output on test set to check if it's possible to lower down the error of the predictive model.

Weighted Average: Different weights are given to each model, indicating how important they are for making predictions. For instance, the responses of these two colleagues will be given more weight than those of the other if two of your coworkers are critics but the others lack any prior knowledge in this area [34].

Stacking: Stacking is an ensemble learning technique that uses predictions from multiple models [34]. Stacking is a powerful method to improve prediction accuracy by leveraging the strengths of multiple individual models. Our base models are CatBoost, Linear Regression and Random Forest who first predict on the data and a meta-model Linear Regression is trained on these predictions to give a final prediction.

XGBOOST: XGBoost (extreme Gradient Boosting) is an advanced implementation of the gradient boosting algorithm. It has high predictive power and is almost 10 times faster than the other gradient boosting techniques [34].

LightGBM: When the dataset is enormous, Light GBM outperforms every other method. When applied to a large dataset, Light GBM runs more quickly than the other techniques. While other algorithms operate using a level-wise approach pattern, Light GBM is a gradient boosting framework that employs tree-based methods and follows a leaf-wise approach [34].

Voting Ensembles:

A machine learning ensemble model called a "voting ensemble" (also known as a "majority voting ensemble") integrates the predictions from many other models. It is a method that can help models perform better, ideally outperforming the performance of any individual model utilised in the ensemble. The forecasts from several models are combined in a voting ensemble. Regression or classification may be done using it. Calculating the average of the model predictions in the case of regression entails this [35]. In our case we did two voting ensemble models one by combining Linear Regression, Random Forest , CatBoost and another one by combining CatBoost and Random Forest.

We checked the performance of all these models on the test set along with Linear Regression, CatBoost and Random Forest which we will explain in detail in the results and analysis section. Even though the Root Mean Squared Error(RMSE) of Weighted Average model came the least we still went with CatBoost model to predict dream11 points . Our decision of doing so will also be justified in the results and analysis section. With the help of Catboost model we predicted the dream11 points of players on which we will have a look at the next section.

NOW A BIT EXPLAIN ABOUT THE 2^{ND} PROBLEM WHY YOU ARE USING PULP TO SOLVE IT AND GIVE SOME REFERENCES.

These points acted as an input for our second and final optimization task where our model gave the output of best dream11 team along with captain and vice-captain following all constraints and maximizing predicted points.

We did some basic preprocessing steps with the data before moving onto the linear programming problem.

Data Preprocessing

Prediction Inclusion: The CatBoost model was used to obtain predicted player performance numbers. The original dataset was then combined with these predictions to create a single data structure.

Dataframe Optimisation: The necessary columns "Player," "Team," "Role," "Playercost," and "Predicted Points" were added to a new dataframe called "optimized_df."

Data cleaning: To confirm the accuracy of the data used in the optimisation process, we examined optimized_df and eliminated any null values and duplicate entries.

Setting up the Linear Programming Problem

Problem Definition: With the help of the PuLP library, we sought to maximise the total projected dream11 points of a dream cricket team while keeping in mind a variety of restrictions (such as the team's budget, the number of players, their responsibilities, etc.).

Variables Initialization

player_vars: Binary variables indicating whether or not each player is on the team (1 if so, 0 otherwise).

captain_vars & vice_captain_vars: Binary variables that determine the captain and vice-captain, respectively.

Objective Function

The primary goal was to maximize the predicted points of the players selected. Base points for players, double points for the captain , and 1.5 times vice captain points were all included in this.

Constraints Imposition

The model was subjected to the following constraints:

Team Structure: There must be exactly 11 players chosen for the team.

Leadership: Only one player can be designated as the captain and one as the vice-captain. In order to be eligible for captaincy or vice-captaincy, a player also has to be a member of the chosen team.

Budget Limit: The total cost of the players that were chosen should not be greater than 100.

Team Composition: Maximum 7 players from one real-world team may be selected for the fantasy team.

Player Role Restriction: Players were chosen within predetermined minimum and maximum ranges, with consideration given to different positions (such as wicketkeepers, batsmen, bowlers, and all-rounders).

Wicketkeepers – 1 (Minimum) 4 (Maximum)

Batsmen – 3 (Minimum) 6 (Maximum)

Bowlers – 3 (Minimum) 6 (Maximum)

All-rounders – 1 (Minimum) 4 (Maximum)

Model Solution

The PuLP library's built-in solver was used to resolve the model. As a result, a list of 11 selected players as well as specific captain and vice-captain selections were given, with the goal of maximising predicted points while respecting all restrictions. The details of this result will be discussed in the next section.

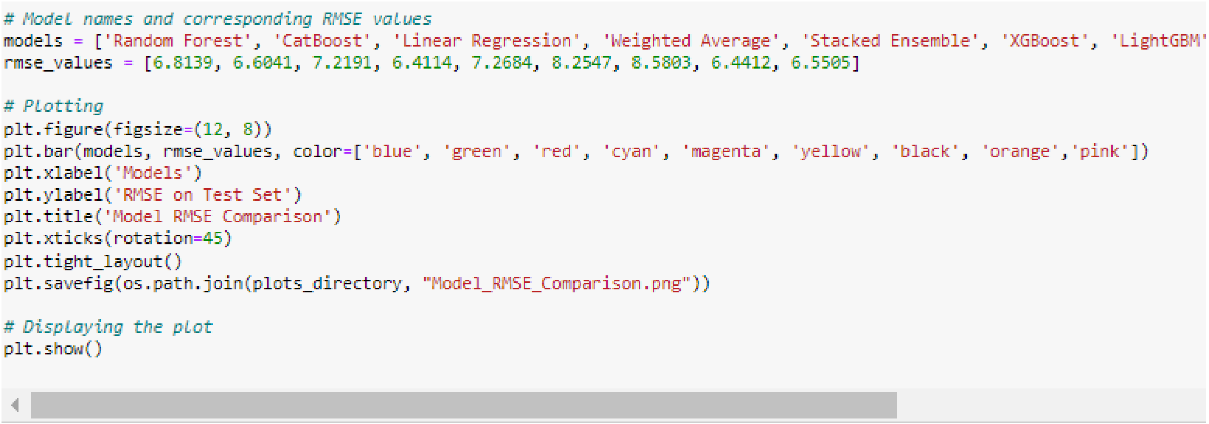
Validation Checks

Following optimisation, a number of claims were made to support:

- 11 players have been chosen.
- The total cost of the players is within the allocated budget.
- The roles and the count of the selected players match the restrictions.
- No single cricket team contributes more than seven players to the fantasy lineup.
- The chosen team includes the captain and vice-captain, who are separate members.

5.RESULTS AND ANALYSIS

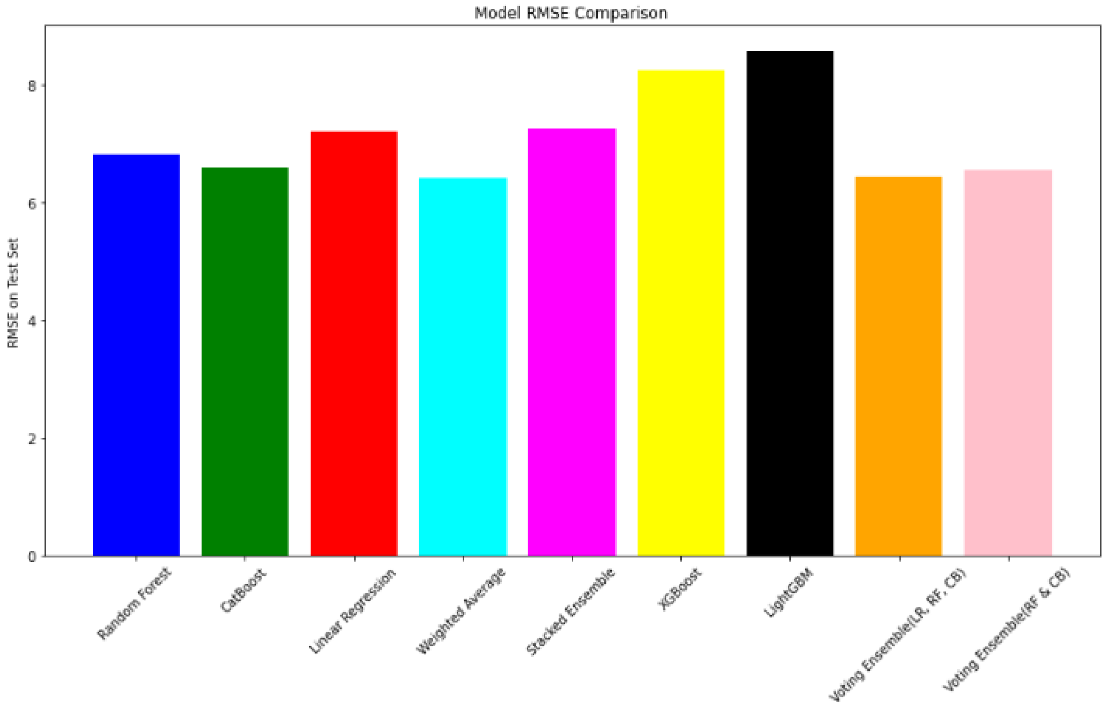
The findings from our models for predicting Dream11 points and the optimisation model for team selection are presented and discussed in this part.



```
# Model names and corresponding RMSE values
models = ['Random Forest', 'CatBoost', 'Linear Regression', 'Weighted Average', 'Stacked Ensemble', 'XGBoost', 'LightGBM'
rmse_values = [6.8139, 6.6041, 7.2191, 6.4114, 7.2684, 8.2547, 8.5803, 6.4412, 6.5505]

# Plotting
plt.figure(figsize=(12, 8))
plt.bar(models, rmse_values, color=['blue', 'green', 'red', 'cyan', 'magenta', 'yellow', 'black', 'orange','pink'])
plt.xlabel('Models')
plt.ylabel('RMSE on Test Set')
plt.title('Model RMSE Comparison')
plt.xticks(rotation=45)
plt.tight_layout()
plt.savefig(os.path.join(plots_directory, "Model_RMSE_Comparison.png"))

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



The RMSE values of the dream11 points prediction model on the test set are as follows:

Random Forest – 6.8139

CatBoost – 6.6041

Linear Regression – 7.2191

Weighted Average – 6.4114

Stacking – 7.2684

XGBoost – 8.2547

LightGBM – 8.5803

Voting Ensemble (Linear Regression + Random Forest + CatBoost) – 6.4412

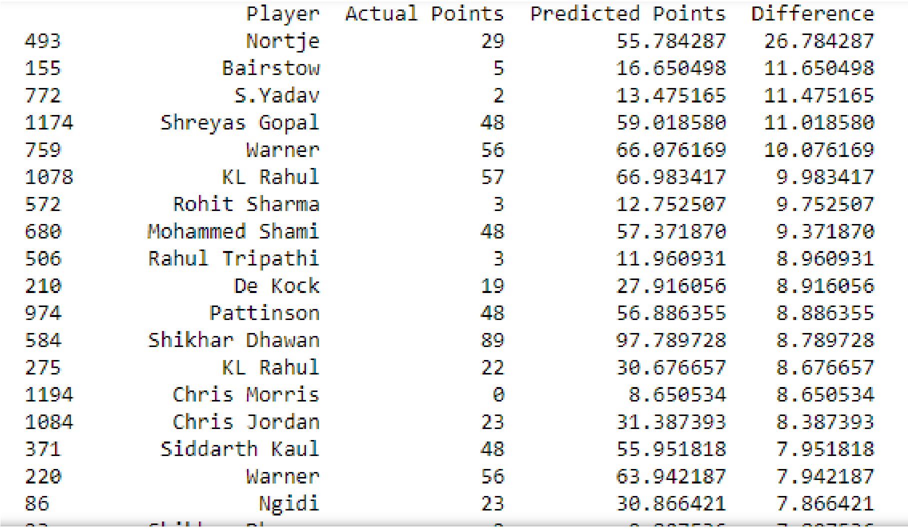
Voting Ensemble (Random Forest + CatBoost) – 6.5505

The RMSE of Weighted Average came least with 6.4114 which means it is the model that will make the least errors. Though the RMSE of catboost came out to bm than Weighted Average and both the voting ensemble models we selected it to be our main model because of the following reasons:

- Compared to ensemble models, CatBoost is simpler by nature because it is a single model. Interpretation and comprehension of the findings may be facilitated by using a single model. Multiple model inspections and handling potential understanding discrepancies between them are not necessary.
- Ensemble models that combine different models like Random Forest, CatBoost, Linear Regression can be computationally expensive. In particular when training on bigger datasets, this may prolong training durations.
- Although ensemble models, particularly when integrating many models, might improve RMSE, there is also a danger of overfitting the training data, which may result in less effective real-world performance.
- With no requirement for preprocessing, CatBoost can handle categorical features directly. Using CatBoost can offer better and more reliable handling of your dataset's categorical characteristics if it has a large number of them without the extra work of one-hot encoding or other modifications.
- The improvement in RMSE is minimal.
- Under some circumstances or for particular sorts of data points, ensemble models, particularly those combining several types of algorithms, may occasionally yield unpredictable predictions. A single model, such as CatBoost, could produce forecasts that are more reliable and steady.
- A single model, such as CatBoost, may be simpler to construct and scale when used in real-world applications than an ensemble of many models. In production contexts, this may lead to simpler maintenance and improved performance.

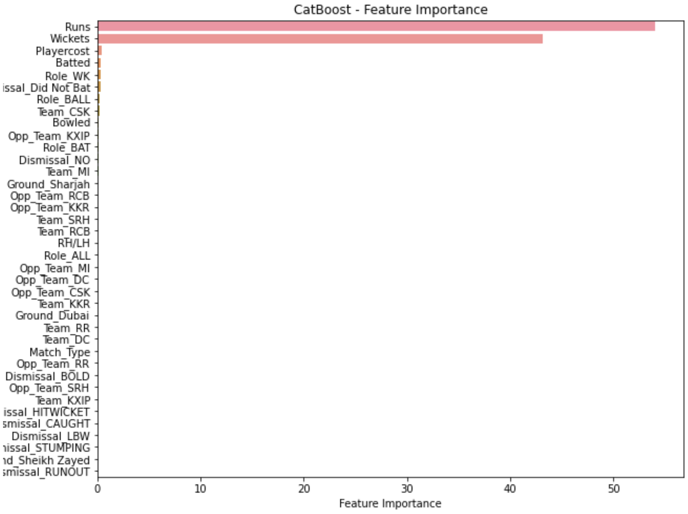
  Also there are many research papers like [36] [20], where we can see outputs from CatBoost is stable and it is constantly coming forward as a top model.

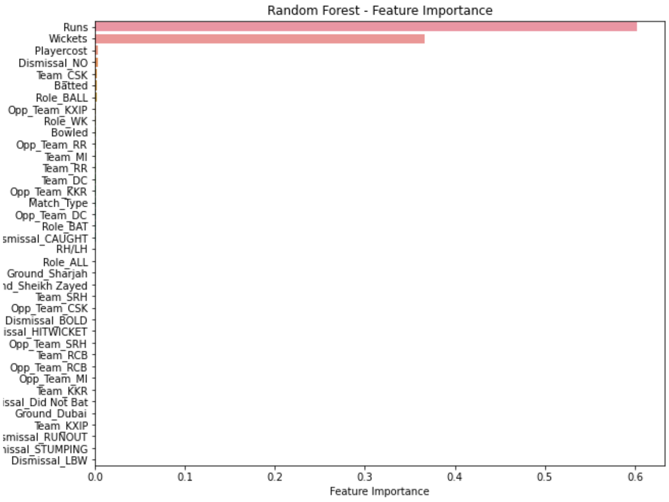
Finally we did prediction of dream11 points on test set with CatBoost some of whose output we will show below:



```
                Player  Actual Points  Predicted Points  Difference
493             Nortje             29         55.784287   26.784287
155           Bairstow              5         16.650498   11.650498
772            S.Yadav              2         13.475165   11.475165
1174      Shreyas Gopal            48         59.018580   11.018580
759             Warner             56         66.076169   10.076169
1078          KL Rahul             57         66.983417    9.983417
572       Rohit Sharma              3         12.752507    9.752507
680     Mohammed Shami             48         57.371870    9.371870
506      Rahul Tripathi             3         11.960931    8.960931
210            De Kock             19         27.916056    8.916056
974          Pattinson             48         56.886355    8.886355
584     Shikhar Dhawan             89         97.789728    8.789728
275           KL Rahul             22         30.676657    8.676657
1194      Chris Morris              0          8.650534    8.650534
1084      Chris Jordan             23         31.387393    8.387393
371      Siddarth Kaul             48         55.951818    7.951818
220             Warner             56         63.942187    7.942187
86               Ngidi             23         30.866421    7.866421
```

As obvious runs and wickets came out to be the top contributing features for both Random Forest as well as CatBoost model.





The Result of the Optimization model using Pulp library came out as follows:

Selected Players: ['Ravi Bishnoi', 'Vijay', 'Nicolas Pooran', 'Siddarth Kaul', 'Pattinson', 'Prabhsimran Singh', 'Murugan Ashwin', 'Sam Curran', 'Chris Gayle', 'W Saha', 'Rinku Singh']

Captain: W Saha

Vice Captain: Chris Gayle

It came with the output of 11 best players that would help user get maximum dream11 points following all constraints.

It selected the captain and vice captain from the selected players list.

The player with the highest projected dream11 points was selected as captain since his points would become 2x and the second highest one was selected as vice-captain whose points would become 1.5x.

It predicted total points the selected team would get which came out to be 687.18 points in this case.

Selected Players:

Ravi Bishnoi: 30.01 points

Vijay: 31.25 points

Nicolas Pooran: 62.77 points

Siddarth Kaul: 55.95 points

Pattinson: 56.89 points

Prabhsimran Singh: 17.93 points

Murugan Ashwin: 55.66 points

Sam Curran: 32.60 points

Chris Gayle: 74.09 points

W Saha: 108.29 points

Rinku Singh: 16.40 points

Captain:

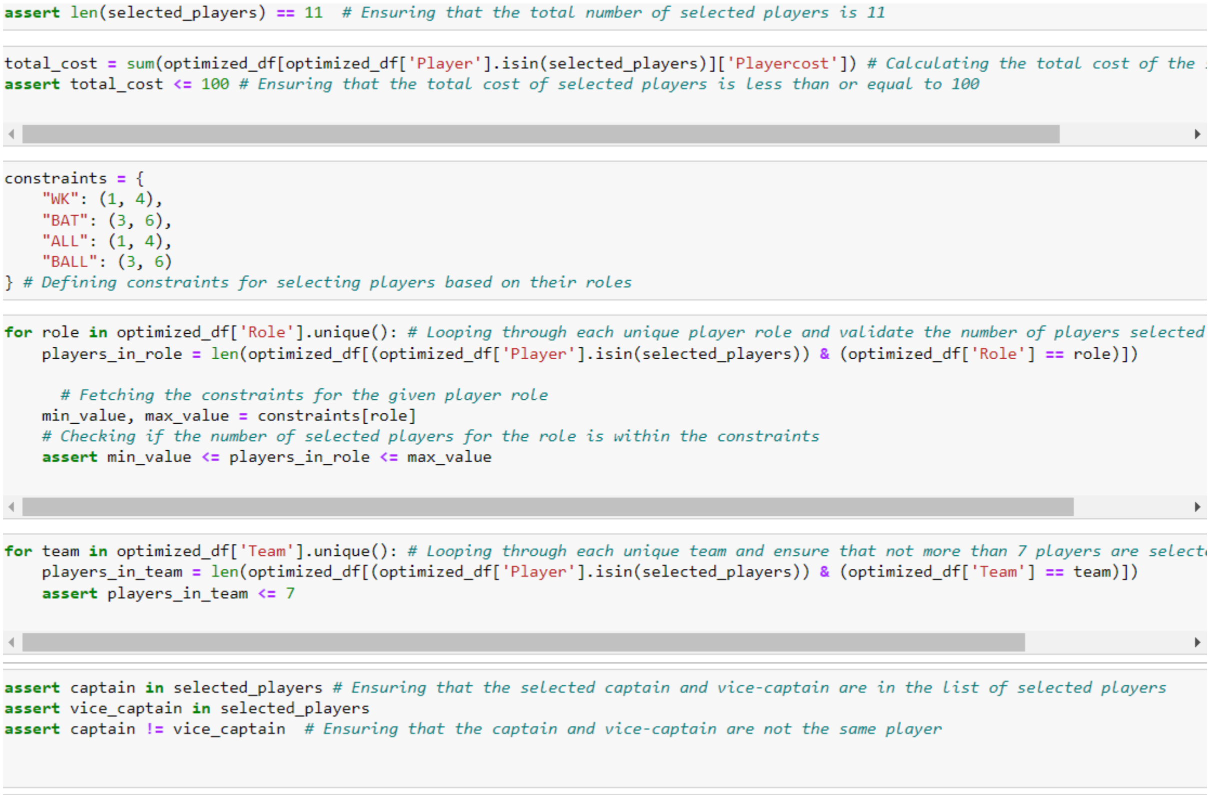
W Saha: 216.58 points (includes bonus)

Vice Captain:

Chris Gayle: 111.14 points (includes bonus)

Total Predicted Points of Selected Team: 687.18 points

The status of the model showed ‘optimal’.



```python
assert len(selected_players) == 11  # Ensuring that the total number of selected players is 11

total_cost = sum(optimized_df[optimized_df['Player'].isin(selected_players)]['Playercost']) # Calculating the total cost of the
assert total_cost <= 100 # Ensuring that the total cost of selected players is less than or equal to 100

constraints = {
    "WK": (1, 4),
    "BAT": (3, 6),
    "ALL": (1, 4),
    "BALL": (3, 6)
} # Defining constraints for selecting players based on their roles

for role in optimized_df['Role'].unique(): # Looping through each unique player role and validate the number of players selected
    players_in_role = len(optimized_df[(optimized_df['Player'].isin(selected_players)) & (optimized_df['Role'] == role)])

      # Fetching the constraints for the given player role
    min_value, max_value = constraints[role]
    # Checking if the number of selected players for the role is within the constraints
    assert min_value <= players_in_role <= max_value

for team in optimized_df['Team'].unique(): # Looping through each unique team and ensure that not more than 7 players are select
    players_in_team = len(optimized_df[(optimized_df['Player'].isin(selected_players)) & (optimized_df['Team'] == team)])
    assert players_in_team <= 7

assert captain in selected_players # Ensuring that the selected captain and vice-captain are in the list of selected players
assert vice_captain in selected_players
assert captain != vice_captain  # Ensuring that the captain and vice-captain are not the same player
```

Validation of Optimization constraints is done properly. All the constraints are working properly in the model since there are no ‘AssertionErrors’.

The CatBoost model offers a balance of accuracy, simplicity, and computing economy, even though the Weighted Average model had the lowest RMSE, according to our investigation and the following outcomes. The optimal squad choice not only adheres to the specified constraints but also maximises the expected Dream11 points, illustrating the potential value of our strategy for fantasy sports enthusiasts.

6. DISCUSSION

Our attempt to estimate Dream11 fantasy points of cricket players using machine learning models turned out to be a viable strategy. CatBoost is an excellent option despite not having the lowest possible RMSE because to its balance of accuracy, interpretability, and computational economy. This is consistent with research that frequently ranks models not just according to their correctness but also according to how useful they are overall in a certain situation. Additionally, a useful method for choosing fantasy sports teams has been demonstrated by fusing optimisation tactics with machine learning forecasts. In addition to following the Dream11 requirements, our optimisation model was able to maximize the expected points. The importance of the features runs and wickets is a key finding from our investigation. Their prevalence demonstrates the applicability of our concept and is consistent with the underlying dynamics of cricket. In any cricket-related model, these direct factors that affect a team's success inevitably become key predictors.

The approach of the study demonstrates the enormous potential and benefit of combining optimisation and machine learning predictions, particularly in the context of fantasy sports.

7. LIMITATIONS

It's important to be aware of our approach's limits despite the positive results:

1. The dataset has extremely limited data and is only available for one IPL season.
2. Additional web scraping efforts were required since complete player costs weren't readily available.
3. Our dataset does not include outside variables that might greatly affect a player's on-field performance, such as injuries, team dynamics, and unexpected changes.
4. There were several things lacking, including catch, weather information, strike rate, economy rate, 4s, 60s, 50s, and 100s, recent form, and psychological variables.
5. Inherently working under the presumption that previous trends are predictive of future results, the models mainly rely on historical fantasy performance. Given the unpredictability of sports, this may not always be true.

8. FUTURE WORK

Our research lays the groundwork for a number of upcoming improvements:

1. It could be possible to improve the forecasts by include current player performance or momentum as a characteristic.
2. More data with all important characteristics spanning multiple seasons is required to provide effective modelling.
3. The creation of an automated pipeline that can gather and analyse the most recent match data will ensure that the model is kept up to date, according to our plans.
4. Further testing with ensemble approaches and looking into various machine learning frameworks like neural networks might produce some insightful comparison results.
5. Complex limitations, such as advantageous player pairings and alliances, can be added to optimise outcomes.
6. New features like catch, weather information, strike rate, economy rate, 4s, 60s, 50s, and 100s, recent form, and psychological variables could be manually added or updated in some dataset
7. In order to further democratise data-driven decision-making in this field, a user-friendly interface may be created that provides fantasy sport players with real-time, optimised team choices. (model_deployment).

9. CONCLUSION

This study sheds light on the revolutionary possibilities of a data-driven technique inside the vast world of fantasy sports. We've started along an analytical path, ending in an improved way to selecting fantasy cricket teams, by leveraging the power of machine learning and optimisation. The CatBoost model, a cutting-edge algorithm renowned for its capacity to offer exact forecasts, is at the core of this method. In our case, it provided a strong foundation for our optimisation efforts by anticipating players' fantasy points. The outcomes of our investigation unmistakably show a big improvement. We have been able to create teams using our optimisation approaches that scrupulously follow platform-specific constraints in addition to maximising predicted points. The conventional, frequently intuitive and emotionally-driven manual selection procedures are in sharp contrast to this scientific approach. Our method reduces the inconsistent nature of subjective human judgement, increasing the chance of success in fantasy leagues. It is based on actual data and methodical computing. Although our technique faces certain challenges, this is true of all scientific endeavours. An element of uncertainty is introduced by the unpredictable nature of sports, which might involve a wide range of unpredictable occurrences, such as unexpected player performances or untimely injuries. Despite the robustness of our model, historical data serve as the cornerstone around which its forecasts are built. This dependence necessitates an ongoing updating system to keep our models current with the most recent advancements in the cricketing world. There are countless opportunities in the future. The prediction power of the algorithm may be improved by enlarging our dataset to include more seasons and significant characteristics like player form, weather conditions, and even psychosocial factors. Our forecasts might be improved by testing a wide range of machine learning architectures, possibly exploiting deep learning or experimenting with ensemble methods. The user experience may be completely transformed if this technology is integrated into a real-time, user-friendly platform. Imagine a situation where fantasy sports fans, independent of their analytical prowess, are provided with cutting-edge, data-driven team choices at the push of a button. This idea may make fantasy sports more inclusive and democratise success. The road is far from finished, despite the fact that we have made tremendous progress in developing a data-driven strategy to selecting fantasy cricket teams. There are several opportunities for improvement and growth, and as we move forward down this path, the blending of data science and sports offers a future in which strategy, rather than random chance, will rule the fantasy sports scene.

10. REFERENCES

[1] Sudhamathy, G. and Meenakshi, G., 2020. PREDICTION ON IPL DATA USING MACHINE LEARNING TECHNIQUES IN R PACKAGE https://www.semanticscholar.org/paper/PREDICTION-ON-IPL-DATA-USING-MACHINE-LEARNING-IN-R-Sudhamathy-Meenakshi/4c54b7e036baaab65e9876f722dc29055a279356

[2] Lewis, Michae. Moneyball: The Art Of Winning An Unfair Game. New York : W.W. Norton, 2004.

[3] Miller, Bennett, Michael De Luca, Rachael Horovitz, Brad Pitt, Steven Zaillian, Aaron Sorkin, Stan Chervin, et al. 2012. Moneyball. Culver City, CA: Sony Pictures Home Entertainment.

[4] Kumar S, S., Prithvi, H.V. and Nandini, C. Data Science Approach to predict the winning Fantasy Cricket Team Dream 11 Fantasy Sports.

https://doi.org/10.48550/arXiv.2209.06999

[5] Dream 11 official site: https://www.dream11.com/fantasy-cricket

[6] Singla, Saurav & Shukla, Swapna. (2020). Integer Optimisation for Dream 11 Cricket Team Selection. INTERNATIONAL JOURNAL OF COMPUTER SCIENCES AND ENGINEERING. (PDF) Integer Optimisation for Dream 11 Cricket Team Selection (researchgate.net)

[7] S. Muthuswamy and S. S. Lam, "Bowler Performance Prediction for One-day International Cricket Using Neural Networks," in Industrial Engineering Research Conference, 2008.

[8] S. R. Iyer and R. Sharda, "Prediction of athletes performance using neural networks: An application in cricket team selection," Expert Systems with Applications, vol. 36, pp. 5510-5522, April 2009

[9] M. G. Jhanwar and V. Pudi, "Predicting the Outcome of ODI Cricket Matches: A Team Composition Based Approach," in European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECMLPKDD 2016 2016), 2016

[10] S. Mukherjee, "Quantifying individual performance in Cricket - A network analysis of batsmen and bowlers," Physica A: Statistical Mechanics and its Applications, vol. 393, pp. 624-637, 2014.

[11] Rodrigues, Nigel, et al. ”Cricket Squad Analysis Using Multiple Random Forest Regression.” 2019 1st International Conference on Advances in Information Technology (ICAIT),IEEE, 2019.

[12] Singla, Saurav, and Swapna Samir Shukla. ”Integer Optimisation for Dream 11 Cricket Team Selection.” International Journal of Computer Sciences and Engineering (2020).

[13] Agrawal, Prachi, and Talari Ganesh. ”Selection of Indian Cricket Team in ODI using Integer Optimization.” Journal of Physics: Conference Series. Vol. 1478. No. 1. IOP Publishing, 2020

[14] M. Bangdiwala, R. Choudhari, A. Hegde and A. Salunke, "Using ML Models to Predict Points in Fantasy Premier League," 2022 2nd Asian Conference on Innovation in Technology (ASIANCON), Ravet, India, 2022, pp. 1-6, doi: 10.1109/ASIANCON55314.2022.9909447.

https://ieeexplore.ieee.org/document/9909447

[15] G. Yang, A.S. Leicht, C. Lago, M.Á Gómez

Key team physical and technical performance indicators indicative of team quality in the soccer Chinese super league

Res Sports Med, 26 (2) (2018), pp. 158-167

Google Scholar

[16] M. Silva, R. Marcelino, D. Lacerda, P.V. João

Match analysis in Volleyball: a systematic review

Monten J Sports Sci Med, 5 (1) (2016), pp. 35-46 Google Scholar

[17]] M. Gómez, S. Ibáñez, I. Parejo, P. Furley

The use of classification and regression tree when classifying winning and losing basketball teams

Kinesiol: Int J Fundam Appl Kinesiology, 49 (1) (2017), pp. 47-56 Google Scholar

[18] C. Chalitsios, T. Nikodelis, V. Panoutsakopoulos, C. Chassanidis, I. Kollias

Classification of soccer and Basketball players' jumping performance characteristics: a logistic regression approach

Sports, 7 (7) (2019), p. 163 Google Scholar

[19] D. Gamble, J. Bradley, A. McCarren, N.M. Moyna

Team performance indicators which differentiate between winning and losing in elite Gaelic football

Int J Perform Anal Sport, 19 (4) (2019), pp. 478-490 Google Scholar

[20] https://medium.com/analytics-vidhya/dream11-team-predictor-with-python-and-machine-learning-f0dfce1489eb

[21] https://towardsdatascience.com/creating-a-fantasy-cricket-team-application-of-linear-programming-4b60c261702d

[22] https://www.ibm.com/topics/exploratory-data-analysis

[22] https://www.kaggle.com/datasets/sukanthen/dream11-ipl2020-live

[23] https://github.com/abhishek374/dream11/blob/main/Data/ipl_squad_points.csv

[24] https://www.ibm.com/topics/exploratory-data-analysis

[25] https://www.educative.io/blog/one-hot-encoding

[26] https://www.geeksforgeeks.org/ml-label-encoding-of-datasets-in-python/

[27] https://www.analyticsvidhya.com/blog/2020/04/feature-scaling-machine-learning-normalization-standardization/

[28] https://deepai.org/machine-learning-glossary-and-terms/feature-reduction

[29] https://www.scribbr.com/statistics/coefficient-of-determination/

[30] Catboost: https://catboost.ai/en/docs/concepts/algorithm-main-stages

[31] Daoud, Essam Al. "Comparison between XGBoost, LightGBM and CatBoost Using a Home Credit Dataset." (2019).

[32]] https://medium.com/analytics-vidhya/why-hyper-parameter-tuning-is-important-for-your-model-1ff4c8f145d3

[33] https://rstudio-pubs-static.s3.amazonaws.com/740098_4d48bd29722f402abf662dd33fc67794.html

[34] https://www.analyticsvidhya.com/blog/2018/06/comprehensive-guide-for-ensemble-models/

[35] https://machinelearningmastery.com/voting-ensembles-with-python/

[36] Choudhari, S., Wagholikar, N., Swaminathan, A., and Kurhade, S., [year of publication, if available]. Dream11 IPL Team Recommendation using Machine Learning and Skill-Based Ranking of Players. Sardar Patel Institute of Technology, Mumbai, India.