IPL SCORE PREDICTION

BDA MINI PROJECT

Done By:
Panel A 07 - Soundaryalaxmi Mahindran
Panel B 08 - Akshaya Thakre
Panel B 46 - Srija Sriram
Panel A 52 - Dimple Agarwal

TOPIC: IPL SCORE PREDICTION

Using JSON Data from Cricket Info

INTRODUCTION

- Cricket is not just a sport, but a religion in India. Given the rise of betting in cricket and impact of endorsements when a player performs well, we wish to use Big Data Analytics to predict the score of a match, every 6 overs.
- We wish to use ML algorithms in an Apache Spark framework to give insights on unstructured JSON data collected. We will implement linear regression models and decision trees to identify best model for score prediction. Through this project, we wish to emphasis the value of data analytics to players, viewers and all stakeholders

ABOUT THE DATASET

Data is collected from the ESPN Cricket Info site [], from 2007. It has the following 22 attributes:

- **Season and start_date:** Indicate the date the match of the IPL was played
- **Venue:** Name of stadium where match was played
- **Innings:** Value 1 indicates the team played first and 2 indicates it played 2nd innings, or chasing team
- **Ball:** Ball is recorded per over, per ball where 6.1 indicates 1st ball of 6th over
- **Batting_team and bowling_team:** Names of the teams playing the match
- **Striker:** Batsman name at crease or currently batting. Every player's name is represented as first name initials followed by surname. Eg: R Dravid, V Kohli etc.
- **Non_striker:** Running batsman who alternates strike with striker
- **Runs_off_bat**: Counts no. of runs made for current ball. This is used to calculate batsman strike rate.
- **Extras, wides, no balls, byes, leg byes and penalty**: Represent the runs conceded by the bowler, but not awarded to the batsmen. These are used in calculation of bowler economy.
- **Wicket_type, player_dismissed:** Used to indicate how batsman got out. Eg: bowled SE Marsh indicates SE Marsh was clean bowled.

DATA PREPROCESSING

IPL data cannot be used for predictions directly, hence we perform following operations:

- Ball by ball data is given, hence roll data up into 1 over using ball count < 6.1.
- Drop columns of wide, no balls, byes, wicket type etc. as they have least impact on predicted score.
- Group players based on name, especially players whose name is same but different initials.
- Label encoder is used to encode stadium names, batsmen and bowlers as numbers.

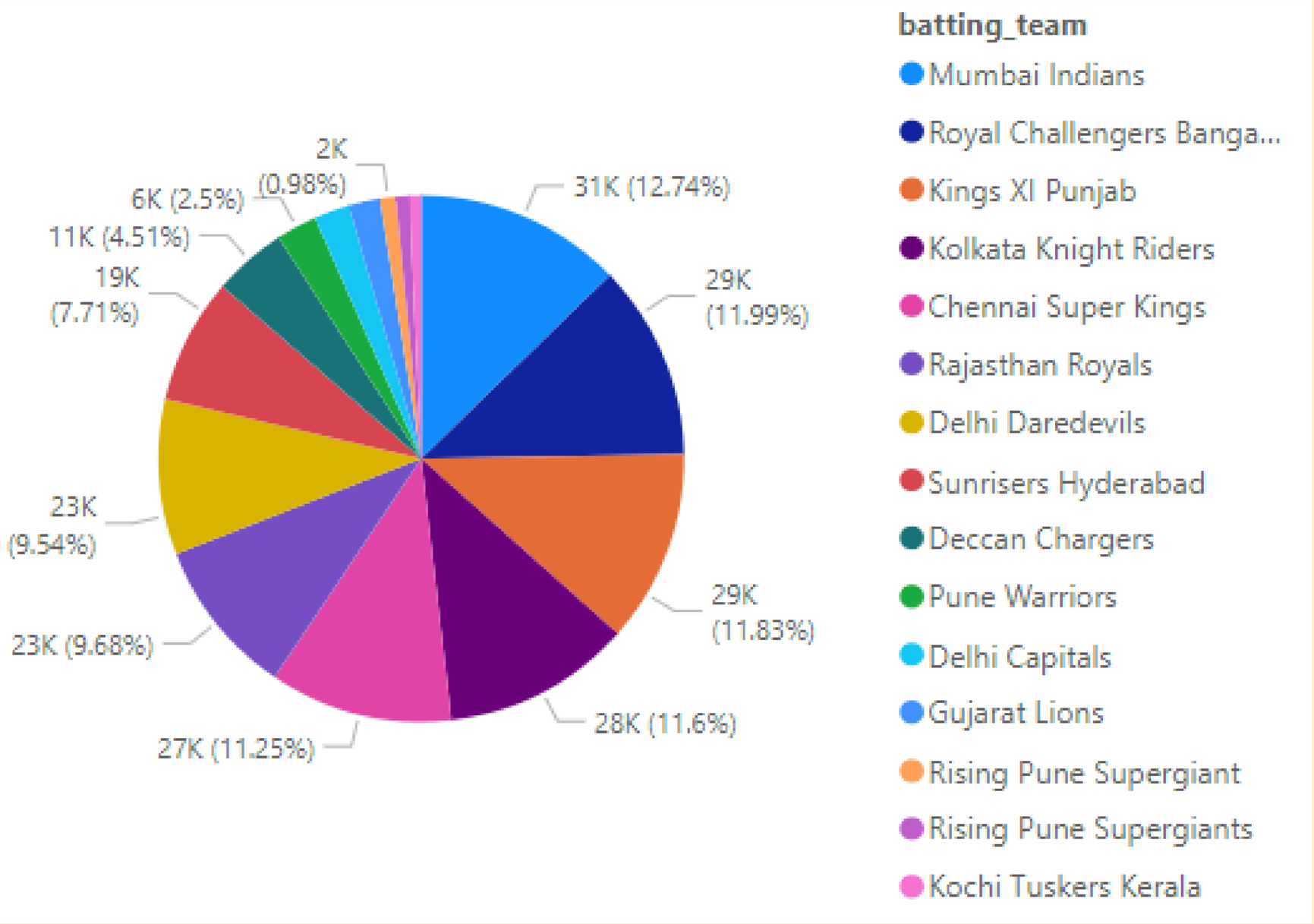
Based on IPL 2022 ranking table, encode team name as number
Final dataset, after all preprocessing operations are displayed in figure below:

	match_id	season	start_date	venue	innings	ball	batting_team	bowling_team	striker	non_striker	bowler	runs_off_bat_x	extras	wides	no
0	335982	2007/08	2008-04-18	0	1.0	0.1	0	0	0	0	0	0.0	1.0	0.0	
1	335982	2007/08	2008-04-18	0	1.0	0.2	0	0	1	1	0	0.0	0.0	0.0	
2	335982	2007/08	2008-04-18	0	1.0	0.3	0	0	1	1	0	0.0	1.0	1.0	
3	335982	2007/08	2008-04-18	0	1.0	0.4	0	0	1	1	0	0.0	0.0	0.0	
4	335982	2007/08	2008-04-18	0	1.0	0.5	0	0	1	1	0	0.0	0.0	0.0	

Fig 1. Dataset after Preprocessing

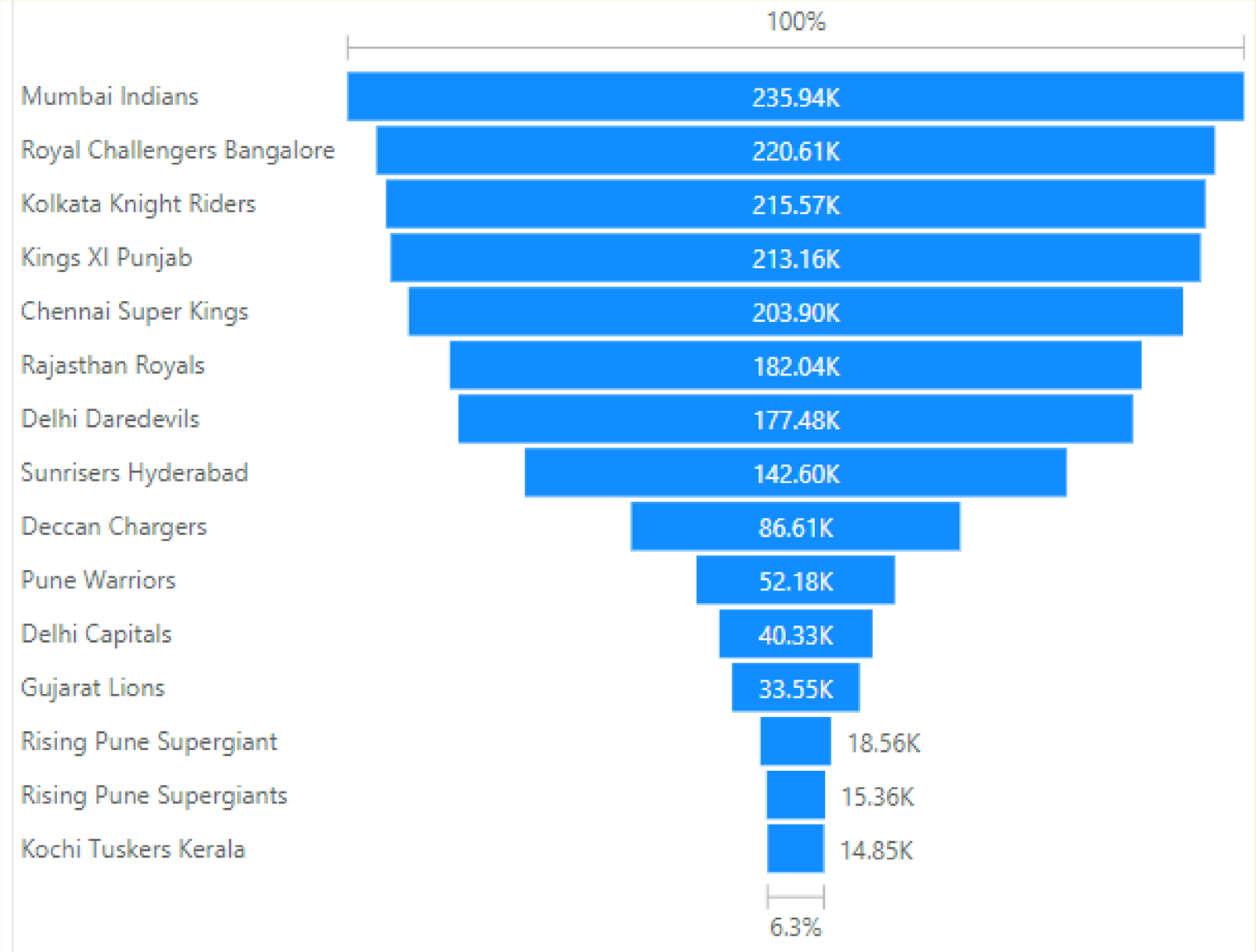
VISUALIZATIONS USING POWER BI

Fig 2. Sum of Runs Made By Batting Team



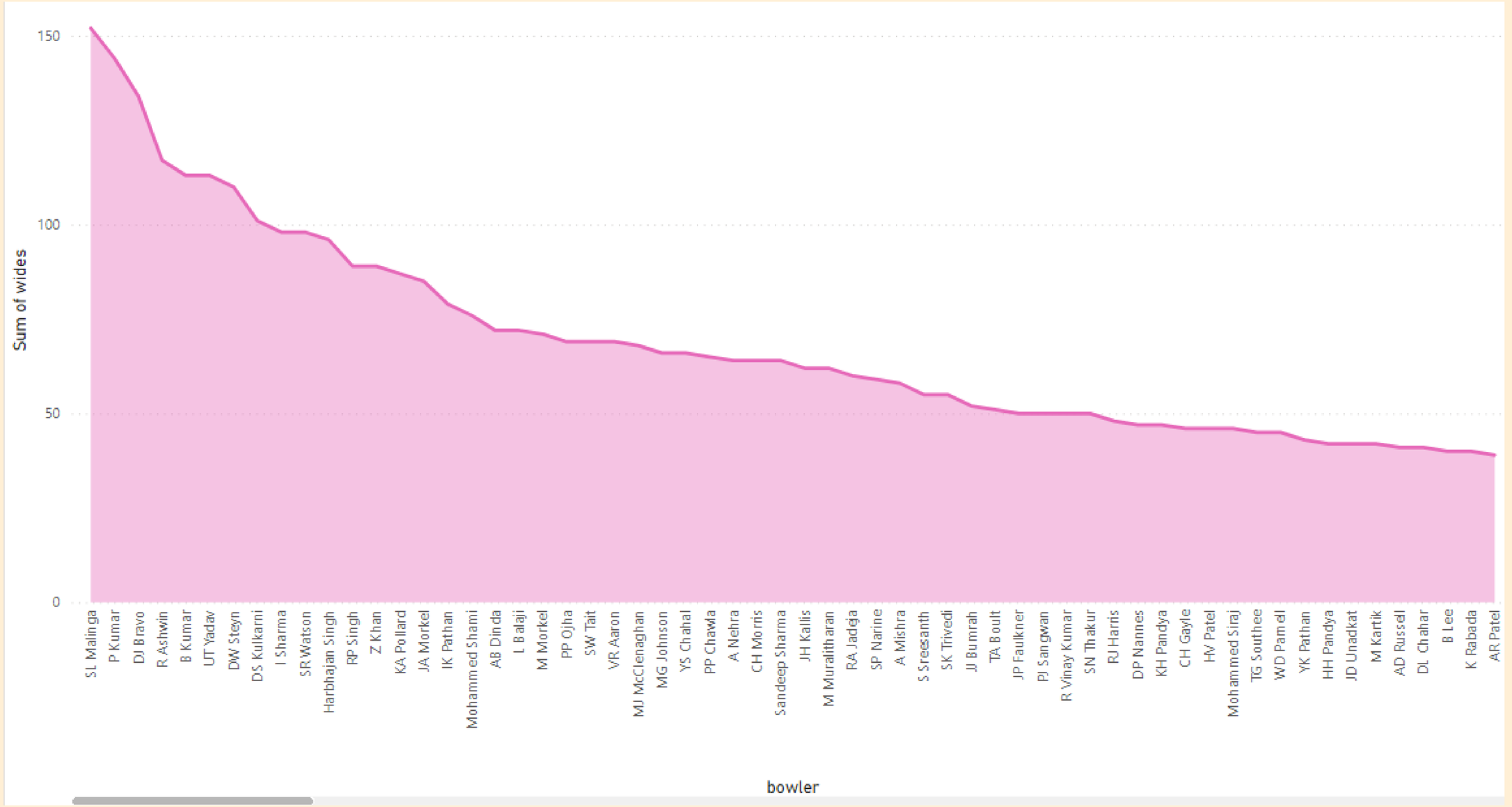
VISUALIZATIONS USING POWER BI

Fig 3. Sums of balls made by the Bowling Team



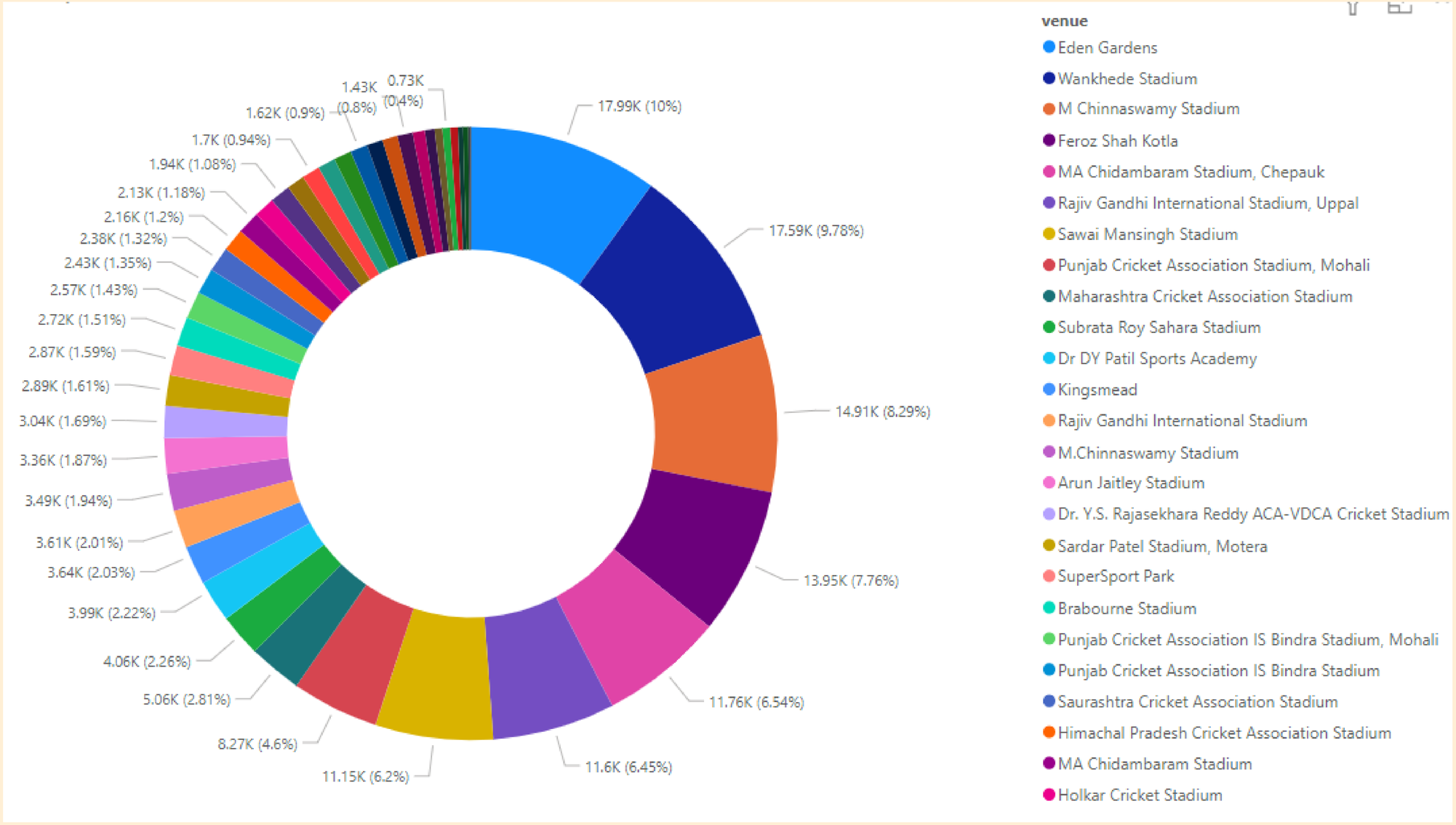
VISUALIZATIONS USING POWER BI

Fig 4. Sums of Wides made by Bowler



VISUALIZATIONS USING POWER BI

Fig 5. Count of Season By Venue



TECH STACK

Apache Spark

- Apache Spark is a data processing framework that can quickly perform processing tasks on very large data sets, and can also distribute data processing tasks across multiple computers, either on its own or in tandem with other distributed computing tools.
- These two qualities are key to the worlds of big data and machine learning, which require the marshalling of massive computing power to crunch through large data stores.
- Spark also takes some of the programming burdens of these tasks off the shoulders of developers with an easy-to-use API that abstracts away much of the grunt work of distributed computing and big data processing.

METHODS

Linear Regression

- Linear regression is a basic and commonly used type of predictive analysis. The overall idea of regression is to examine two things:
- 1. Does a set of predictor variables do a good job in predicting an outcome (dependent) variable?
- 2. Which variables in particular are significant predictors of the outcome variable, and in what way do they–indicated by the magnitude and sign of the beta estimates–impact the outcome variable?
- These regression estimates are used to explain the relationship between one dependent variable and one or more independent variables. The simplest form of the regression equation with one dependent and one independent variable is defined by the formula y = c + b*x, where y = estimated dependent variable score, c = constant, b = regression coefficient, and x = score on the independent variable.

METHODS

Logistic Regression

- A logistic regression model predicts a dependent data variable by analyzing the relationship between one or more existing independent variables.
- For example, a logistic regression could be used to predict whether a political candidate will win or lose an election or whether a high school student will be admitted or not to a particular college.
- These binary outcomes allow straightforward decisions between two alternatives.

METHODS

Decision Tree Regression

- Decision Tree is a decision-making tool that uses a flowchart-like tree structure or is a model of decisions and all of their possible results including outcomes, input costs, and utility.
- Decision tree regression observes features of an object and trains a model in the structure of a tree to predict data in the future to produce meaningful continuous output. Continuous output means that the output/result is not discrete, i.e., it is not represented just by a discrete, known set of numbers or values.

METHODS

Random Forest Regression

- Random forest is a Supervised Machine Learning Algorithm that is used widely in Classification and Regression problems.
- It builds decision trees on different samples and takes their majority vote for classification and average in case of regression.
- One of the most important features of the Random Forest Algorithm is that it can handle the data set containing continuous variables as in the case of regression and categorical variables as in the case of classification.
- It performs better results for classification problems.

RESULTS

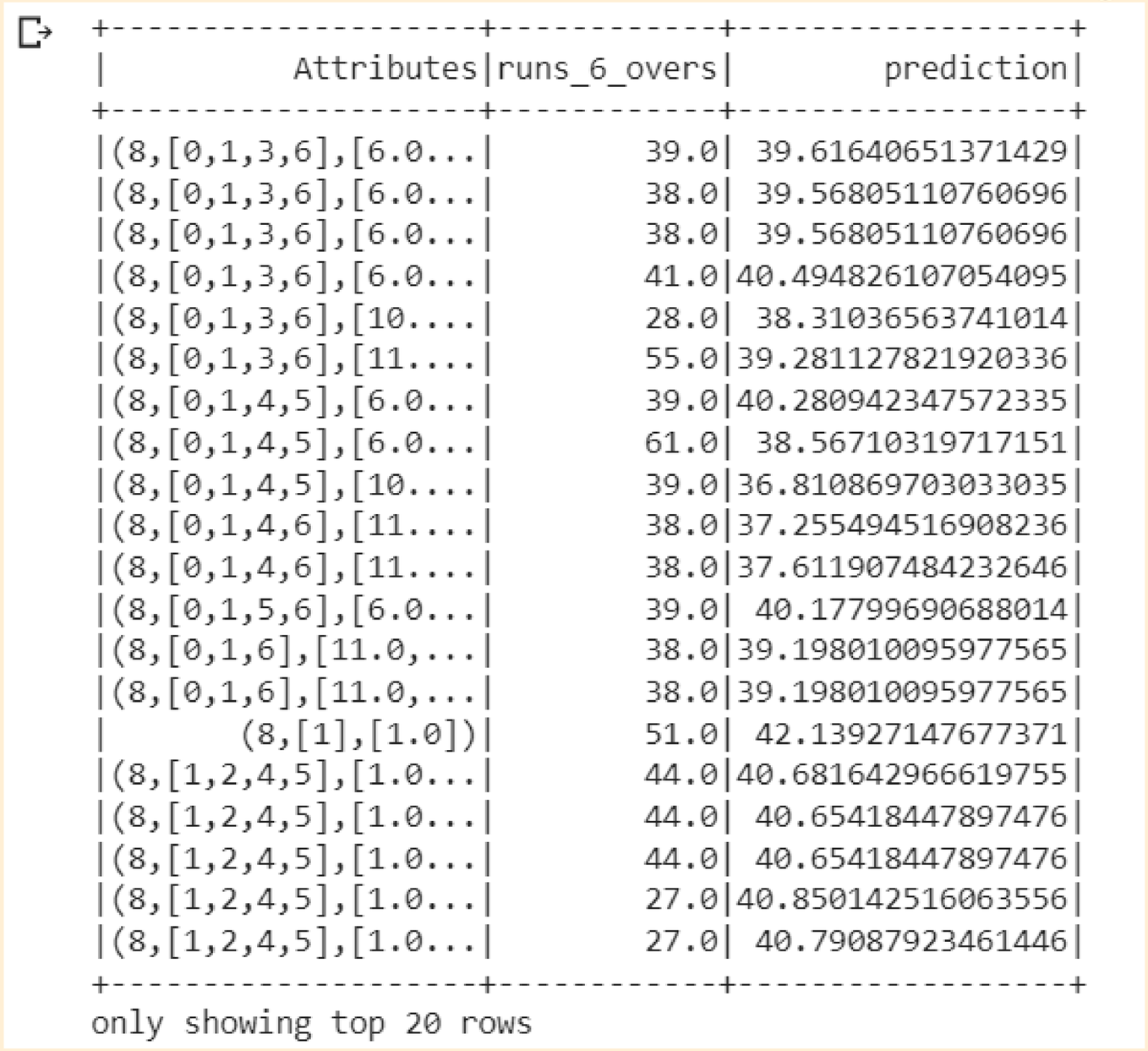
After comparing and analyzing the RMSE, MAE and R2 of all 4 regression models we have come to a conclusion that Random Forest Regression produces the best results showing low RMSE and high accuracy

Model Evaluation

We have used Root mean squared error (RMSE) to evaluate the model. RMSE score of 11.109 implies low prediction error.

```
RMSE: 11.109
MAE: 8.818
r2: 0.106
```

Fig 6. Predictions with Random Forest Regression



Attributes	runs_6_overs	prediction
(8,[0,1,3,6],[6.0...	39.0	39.61640651371429
(8,[0,1,3,6],[6.0...	38.0	39.56805110760696
(8,[0,1,3,6],[6.0...	38.0	39.56805110760696
(8,[0,1,3,6],[6.0...	41.0	40.494826107054095
(8,[0,1,3,6],[10....	28.0	38.31036563741014
(8,[0,1,3,6],[11....	55.0	39.281127821920336
(8,[0,1,4,5],[6.0...	39.0	40.280942347572335
(8,[0,1,4,5],[6.0...	61.0	38.56710319717151
(8,[0,1,4,5],[10....	39.0	36.810869703033035
(8,[0,1,4,6],[11....	38.0	37.255494516908236
(8,[0,1,4,6],[11....	38.0	37.611907484232646
(8,[0,1,5,6],[6.0...	39.0	40.17799690688014
(8,[0,1,6],[11.0,...	38.0	39.198010095977565
(8,[0,1,6],[11.0,...	38.0	39.198010095977565
(8,[1],[1.0])	51.0	42.13927147677371
(8,[1,2,4,5],[1.0...	44.0	40.681642966619755
(8,[1,2,4,5],[1.0...	44.0	40.65418447897476
(8,[1,2,4,5],[1.0...	44.0	40.65418447897476
(8,[1,2,4,5],[1.0...	27.0	40.850142516063556
(8,[1,2,4,5],[1.0...	27.0	40.79087923461446

only showing top 20 rows

CHALLENGES FACED

- It was difficult to gather data for quite a large period of time i.e., approximately 8 years.
- Since during the period of 8 years, many teams have dropped their name from IPL tournament and many teams have replaced their team's name, so we had to clean the dataset accordingly.

LEARNINGS

- We learnt how to use Apache Spark which is an open-source, distributed processing system used for big data workloads.
- Learnt about four different models namely Linear regression, Logistic regression, Decision Tree and Random forest regression for prediction.
- Learnt how to calculate RMSE, MAE and r2 which are important factors for determining accuracy.

CONCLUSION

- Thus, we successfully predicted the score of IPL matches using random forest model. The data was historic, with emphasis given to pitch conditions, player form, position in points table and more.
- We achieved RMSE of 11.109 and studied the impact of various attributes on cricket.

REFERENCES

1. A. A. Aburas, M. Mehtab and Y. Mehtab, "ICC World Cup Prediction Based Data Analytics and Business Intelligent (BI) Techniques," 2018 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC), 2018, pp. 273-2736, doi: 10.1109/CyberC.2018.00056.
2. T. Singh, V. Singla and P. Bhatia, "Score and winning prediction in cricket through data mining," 2015 International Conference on Soft Computing Techniques and Implementations (ICSCTI), 2015, pp. 60-66, doi: 10.1109/ICSCTI.2015.7489605.
3. M. J. Awan et al., “Cricket Match Analytics Using the Big Data Approach,” Electronics, vol. 10, no. 19, p. 2350, Sep. 2021, doi: 10.3390/electronics10192350.
4. Abdurazzag A. Aburas, Muhammed Mehtab, and Yusuf Mehtab. 2018. Cricket World Cup Predictions Using KNN Intelligent Bigdata Approach. *In Proceedings of the 2018 International Conference on Computing and Big Data (ICCBD '18)*. Association for Computing Machinery, New York, NY, USA, 18–22. DOI: https://doi.org/10.1145/3277104.3277117
5. https://spark.apache.org/docs/latest/ml-classification-regression.html#random-forest-regression
6. https://spark.apache.org/docs/latest/ml-classification-regression.html#linear-regression
7. https://spark.apache.org/docs/latest/ml-classification-regression.html#decision-tree-regression
8. https://spark.apache.org/docs/latest/ml-classification-regression.html#logistic-regression

Thank you!