1. INTRODUCTION

1.1 PROJECT DESCRIPTION

The system is to predict the rate of a batsman, there are a few key facts in mind. Runs: Runs are one of the most important parameter which gives an insight in regards to what is expected from a first class player. This in-turn determines the player reputation with fans. Batting average: The total number of runs a batsman has scored divided by a number of times they have been out [3] This tells us how consistent a batsman is, that's important because the team should score within 10 wickets of it Thatis vital as, but aside from the truth that not all of them are batting speclialistas (proper sentence eror) the community has to bring about those 10 wickets. A batting strike rate in cricket is the frequency with which a batsman achieves the primary goal of batting, which is to score runs [4]. The reason why it is important is that the longer the high scoring games go on for, the important rate at which the runs are coming are. Balls faced : The range is the number of balls that the batsman has faced in the whole season. A batsman can bat inside the first Inning or 2d Inning, which is determined type the toss release by using the captain of prevailing that Institution. Here, the output of our model is Price.

1.1.1 Problem Scenario:

Imagine a scenario wherein an IPL franchise group is getting ready for the imminent public sale. The user have a restrained price range and need to build a aggressive team inside that constraint. However, they lack insights into participant valuations and group compositions, making it challenging to formulate a triumphing approach for the public sale.

In the time of IPL all of us tries to are expecting the value of the players visitors can not find the charge for a way tons players can get with the help of this model visitors also can get to know approximately anticipated charge for the gamers primarily based on their performance inside the tournament. It offers more accuracy approximately the players how a great deal they should get paid.

1.1.2 Proposed Solution:

The proposed answer for IPL auction prediction entails utilising system learning algorithms and data analytics to research beyond auction facts and player performances, extract relevant features, educate predictive models, and make knowledgeable predictions about player fees and team compositions for upcoming auctions.

1.1.3 Purpose:

The purpose of challenge IPL auction prediction the use of device gaining knowledge of is to revolutionize the choice making manner by using of advanced analytics and gadget getting to know algorithms, the purpose is to offer stakeholders—franchise owners, coaches, and lovers—with correct ,This initiative seeks to beautify the strategic planning and useful resource allocation of IPL franchises, optimizing their possibilities of assembling aggressive and balanced groups while navigating the unpredictable dynamics of the auction.

Team Composition: Identifying most fulfilling group combinations that complement every different's strengths. This extends to predicting the impact of a new player on the general group dynamic and overall performance.

1.1.4 Project Scope:

Player Valuation: The number one consciousness is on predicting the fair marketplace price of gamers based on historic performances, information, gambling situations, and numerous contextual elements..

2. LITERATURE SURVEY

2.1 DOMAIN SURVEY

- Existing machine learning models or algorithms used for sports predictions, particularly in cricket or IPL.
- Data sources commonly ued for training machine learning models in sports prediction.
- Performance metrics used to evaluate the accuracy and effectiveness of machine learning models in sports prediction.
- Challenges or limitations specific to predicting IPL auction outcomes compared to other sports prdiction tasks.
- Any relevant research papers, articles, or case studies related to IPL auction prediction or sports prediction in general.
- Feedback or insights from experts in cricket analytics or sports prediction.

2.2 RELATED WORK

Paper 1

● Machine Learning Techniques for Predicting IPL Players Cost Pay

● Author: Nagaraj P, Muneeswaran V, Raja M, M C Prabhu, B Meghana

● Appeared in: 2023 Second International Conference on Augmented Intelligence and Sustainable Systems (ICAISS)

● Year: 23 August 2023

● summery: The fulfillment of an IPL player fee prediction device lies in its capacity to are expecting participant public sale fees. Accuracy of the machine can be measured using different metric such as Mean absolute errors, mean square errors and root mean square errors. These metrics show how closely forecast fees correspond to true auction prices.

Paper 2

IPL Players Price Prediction Using Machine Learning Techniques.

● Author: Nagaraj P, Muneeswaran V, Raja M, M C Prabhu, B Meghana

● Appears in: 2023 Second International Conference on Augmented Intelligence and Sustainable Systems (ICAISS)

● Date: 23 August 2023

summery: On the other hand the determinate to attend a fee (IPL player cost prediction machine) based on an regression compute is one of the fulfillment societies. The correctness of the machine implemented can be measured using different metrics like mean absolute error (MAE), mean squared errors (MSE) and root mean square errors (RMSE). These measures show how accurate the forecasted Fees are to the ground truth Sale Payments.

Paper 3

Predict and Evaluate the Performance of a Cricket Player Using Machine Learning Algos

● Authors: M Sumathi, S Prabu, M Rajkamal

Year : 05 April 2023

○ summery: A paper studies various ML methods in order to predict whether or not a certain player will perform well. Uses linear regression, K-Means and random woodland models will are expecting the overall performance of male cricket player. The functionality of cricket players are predicted and regressed with linear strains the usage of linear regression to pick the appropriate feature for performance research..

Paper 4

Predicting Results of Indian Premier League T-20 Matches the usage of Machine Learning

- Authors: Shilpi Agrawal, Suraj Pal Singh, Jayash Kumar Sharma
- Published in: 2018 8th International Conference on Communication Systems and Network Technologies (CSNT)
- Year: 24 November 2018
- summery : In this paintings, historic statistics has been gathered from actual IPL cricket suits and useful capabilities have been extracted after pre-processing of data. Further, suitable records is transformed to a numeric form and scale it on three parameters win, loss, and tie. This information is skilled and classified with three classifier SVM, CTree and Naïve Bayes using R Tool.

Paper 5

Selection of gamers and the teamfor an indian gold standard leaguecricket suit usingensemblof classifier

- author:-j jhansi raniaditya vidyadhara kamath,aditya menon
- Published in: 2020 IEEE International Conference on Electronics, Computing and Communication Technologies (CONECCT)
- Year: 02 July 2020
- summery : This paper explores the capacity of Support Vector Machines and Random Forests for predicting participant costs inside the IPL auction. It analyzes each traditional overall performance metrics and superior capabilities like form index and

player effect rating. The authors spotlight the significance of Realtime records integration and model variation to address the dynamic nature of the auction environment.

2.3 EXISTING SYSTEMS

At the time of my last update in January 2022, there is no one system that is popular for IPL auction prediction across the circles. But a number of platforms and companies have made their own algorithms and engines that predict how the IPL auction can proceed. Using different types of machine learning techniques, statistical analysis, and data mining combinatorially, these systems analyse data and predict outcomes of the matches by applying massive amounts of data on historical data, player performances, team needs, and other key aspects. A few popular data analytics companies, sports data providers, and fantasy sports platforms provide IPL auction prediction service or tools. Systems and methodologiesFor most market makers, the details of their systems and methodologies are proprietary and are not publicly disclosed.

It must be remembered, however, that while these data-based systems for decision-making provide very useful tools for team owners and analysts going into the IPL auctions, given the uncertain environment of the auctions and the fluctuations in player performances, no system can predict the future with full accuracy.

2.4 TECHNOLOGY SURVEY

Python

Python is a highlevel programing language developed by Guido Van Rossum. Originally launched in 1991 Today Python interpreters are available for lots running Systems like Window and Linux. Free to use and distribute, Open Source, Community supported Python. The Python interpreter is available on many important systems.. It is superior to traditional and backing language due to the backing of advance reuse mechanism. A lot smaller language, and far more expressive than statically typed languages Lesser to Type Less to Debug Less to Maintain No Long

Compile and Link Types It is compatible as it runs on almost all platforms, python programs run across nearly every major platform in use. Python has a wide range individual frameworks from text pattern matching to networking.

Pycaret

First released as an alpha version in May 2018, PyCaret is an open-source, low-code machine learning library in Python that enables automation of machine learning workflows. It is an all-in-one from model management to machine learning and which drastically reduces the time it takes to perform trials and makes you more productive.

Basic Information about PyCaret: As for the difference with other open-source libraries, PyCaret is an alternative low-code library that is primarily used for replacing hundreds of lines of code with few lines only. So experiments are going to be 10^-3 or even 10^-2 times faster and hence it makes our life faster in turn. PyCaret is a Python library for performing an end to end model evaluation process including data loading, preprocessing, and visualization. It is essentially a low-code machine learning library that combines best-in-suit ML frameworks to make it easy for developers to learn about their models and improve them by using out-of-the-box solutions.

PyCaret is designed to work well with Citizen Data Scientists and this is also reflected in its design and ease of use. a term first used by Gartner. A Citizen Data Scientist is a power user who can do easy and moderately difficult analytics, something that earlier required much more technical hands.

CatBoost:

A mutation of gradient boosting, Catboost is able to handle both categorical and numerical features. It take now not any function encodings methods like One-Hot Encoder or Label Encoder to convert categorical functions into numbers. It additionally implements an algorithm called symmetric weighted quantile sketch (SWQS) which implicitly manages the missing values in the dataset in order to

maximize accuracy while minimizing overfit, leading to superior performance of the resulting model.

Features of CatBoost

Method for handling categorical features: CatBoost effectively handle categorial features without preprocessing or shuffle. Top- of-the-line result without parameter tuning : CatBoost aims to give the best solution even when we did not play around much with parameters. This is an effective way for saving time and effort as the user can get competitive performance with default parameters.

Handling missing values using built-in methods: CatBoost can deal with lacking values in the enter data while not having to impute - unlike in other Models.

Auto function scaling: CatBoost internal scales all the columns to equal scaling whilst in different methods we want to convert columns substantially.

Resistance to Overfitting: CatBoost features strong Tree Boosting algorithms and Ordered Boosting and using random permutations for combinations of attributes to help prevent overfitting These techniques help in building models that generalize well to unseen data.
In conjunction w/awsensors Matching its features list, CatBoost uses an internal go-validation method to make sure it chooses the quality hyperparameters for your model.

Fast Scalable GPU version : CatBoost provides a GPU- version of its algorithm.

2.5.FEASIBILITY STUDY

2.5.1 Technical Feasibility

The project is technically possible because player data is available and machine learning algorithms such as CatBoost could be used, considering that the problem is actually a regression task. There are some of the challenges which are data cleaning is a messy task and it becomes tedious in the case of larger data sizes, choosing the relevant features is also a concern, we will use Cloud Services like GCP for this since they are efficient and scalable. In sum, it was a well-planned and well executed project and probably a successful application of machine learning in the IPL domain.

2.5.2 Operational Feasibility

IPL player price prediction project will likely be operationally feasible. The model itself doesn't need to incur much in the way of operational costs beyond potentially the cloud storage for the data and trained model. But, integrating the model into the IPL franchises' existing workflows still remains a challenge. It should be implemented successfully might be:

Create systems that prompt franchise personnel to interact with the model and understand its predictions.

Getting decision-makers within franchises that are full of tradition to buy in.

Keeping the model up to date with fresh data in order to keep the accuracy high, and the model relevant as player performance or the market dynamics change.

2.5.3 Economic Feasibility

The project is economically feasible as long as franchises find it more valuable than its cost. Advantages can encompass more efficient auctions, which both save money and increase team performance, but costs can include the time to acquire and process data, the compute power needed, and features development. On balance, if whatever savings and revenues can be accrued from the project offset the expense of developing the thing, then it sounds like a goer. Testing the actual result can be the other way of showing the built model is working correctly which will help to lift the economical lift value.

2.5.4 Market Feasibility

IPL player price prediction model is good market Most franchises do have analytics teams, but for a wider penetration of user-friendly tools is an inevitable gap, especially among the smaller franchises. Be prepared to research competition, promote your model's unique value (i.e., accuracy, proprietary features), and rollout an attractive pricing model (subscriptions, pay-per-use, tiers of features) to appeal to franchise buyers and driving revenues.

3. HARDWARE AND SOFTWARE REQUIREMENTS

3.1Hardware Requirements:

- Computer or Laptop: A computer with sufficient processing power is required for training and running the deep learning model. A machine with a dedicated GPU (Graphics Processing Unit) is recommended to faster model training.
- RAM 8GB: Sufficient RAM is crucial, especially when dealing with large datasets. A minimum of 8 GB of RAM is recommended for efficient model training.
- GPU: I have used NVIDIA GeForce MX250 4gb graphic card, Graphic card in mandated since it takes longer time for larger epochs
- Processor: I have used Intel Core i7-10210U CPU running at 1.60 GHz. These processor helps in faster training, testing and prediction.

3.2Software Requirements:

- Operating System: Windows 10
- Python Libraries: Image Processing Libraries like OpenCV. NumPy, Pandas, Seaborn, Pycaret and scikit-learn, for efficient data manipulation and model evaluation.
- Visual Studio Code (VSCode) as the Integrated Development Environment (IDE) (Version: 2023.3.1).
- Development Tools: Python Flask (Version 3.11) for the application, Jupyter Notebooks

4. SOFTWARE REQUIREMENTS SPECIFICATION

4.1Team Management

The user can pass the players parameter such Raa, Efscores, Wins, Salary
And the our system will generate the predicted value in money(dollars)
By looking at the budget the user can bid the player and add the player to the team

4.2Functional Requirements:

- **Team Management**

  SEARCH PLAYER: This characteristic lets in you to look up records about a selected player. You may use it to discover records, performance records, or other relevant facts approximately gamers you are inquisitive about.

  PREDICT VALUE: This characteristic in all likelihood allows you estimate the destiny overall performance or price of a player. It might also use historical records and different metrics to provide a prediction on how treasured a player might be in the destiny.

  ADD/DISCARD PLAYER: This characteristic permits you to add a participant on your team or discard a participant out of your crew. It is beneficial for handling your roster and making strategic decisions about which gamers to hold or let cross.

  CREATE TEAM: This function allows you to form a new team by means of selecting players. You can build your crew from scratch, selecting players based totally on their performance, price, and different standards.

  VIEW PURSE: This function helps you to see the whole amount of cash or finances you have available. It enables you manage your price range, ensuring that you stay within your price range while acquiring new players.

  VIEW TEAM: This feature displays the modern roster of your group. You can see all the gamers which can be part of your team, at the side of their statistics and other applicable records.

5. SYSTEM DESIGN

5.1ARCHITECTURE DIAGRAM

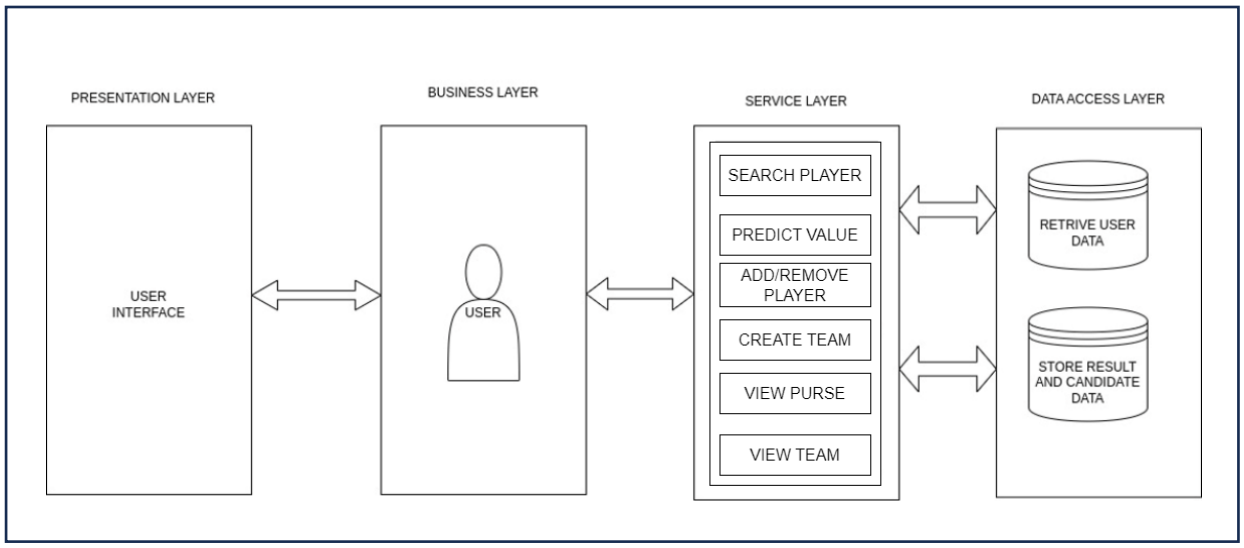


Fig 1: Architecture Diagram

This Diagram shows how the user interacts with the web interface to search for the player value to know its prediction, when the players name is inserted through presentation layer, the data is passed to service layer where the data is sent to augment, preprocess and apply the trained model to find the predicted value of the player.

5.2CONTEXT DIAGRAM

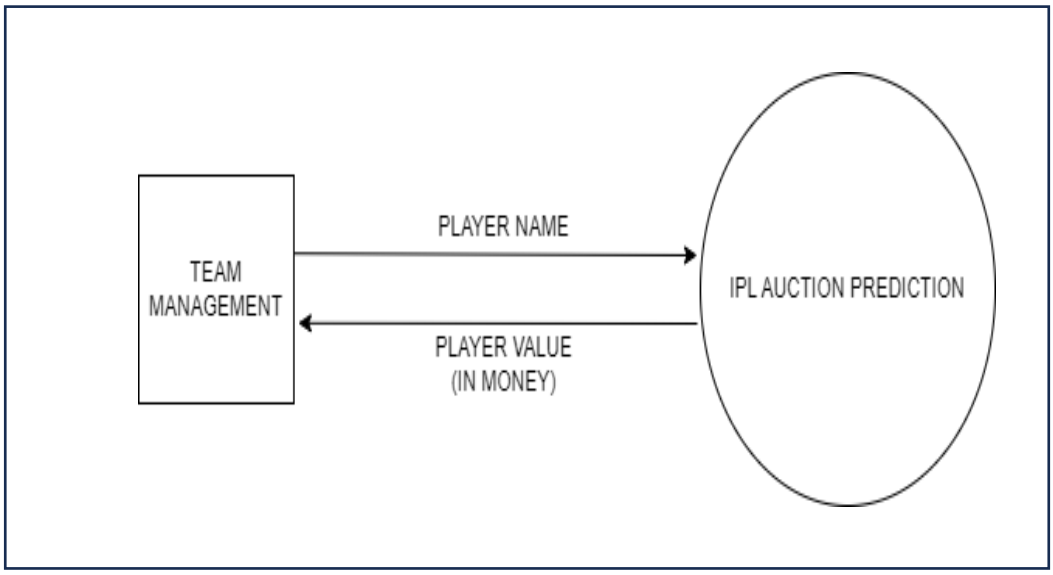


Fig2: Context Diagram

A context diagram provides an overview of a system and its interactions with external entities. For an IPL auction prediction project, the context diagram would illustrate the system's components and how it interacts with users and external data sources.

The context diagram provides a high-level view of how the IPL auction prediction system interacts with its environment, including users, external data sources, and feedback mechanisms. It helps stakeholders understand the system's scope, functionality, and dependencies, laying the groundwork for more detailed system design and implementation.

6. DETAILED DESIGN

6.1PROCESS FLOW

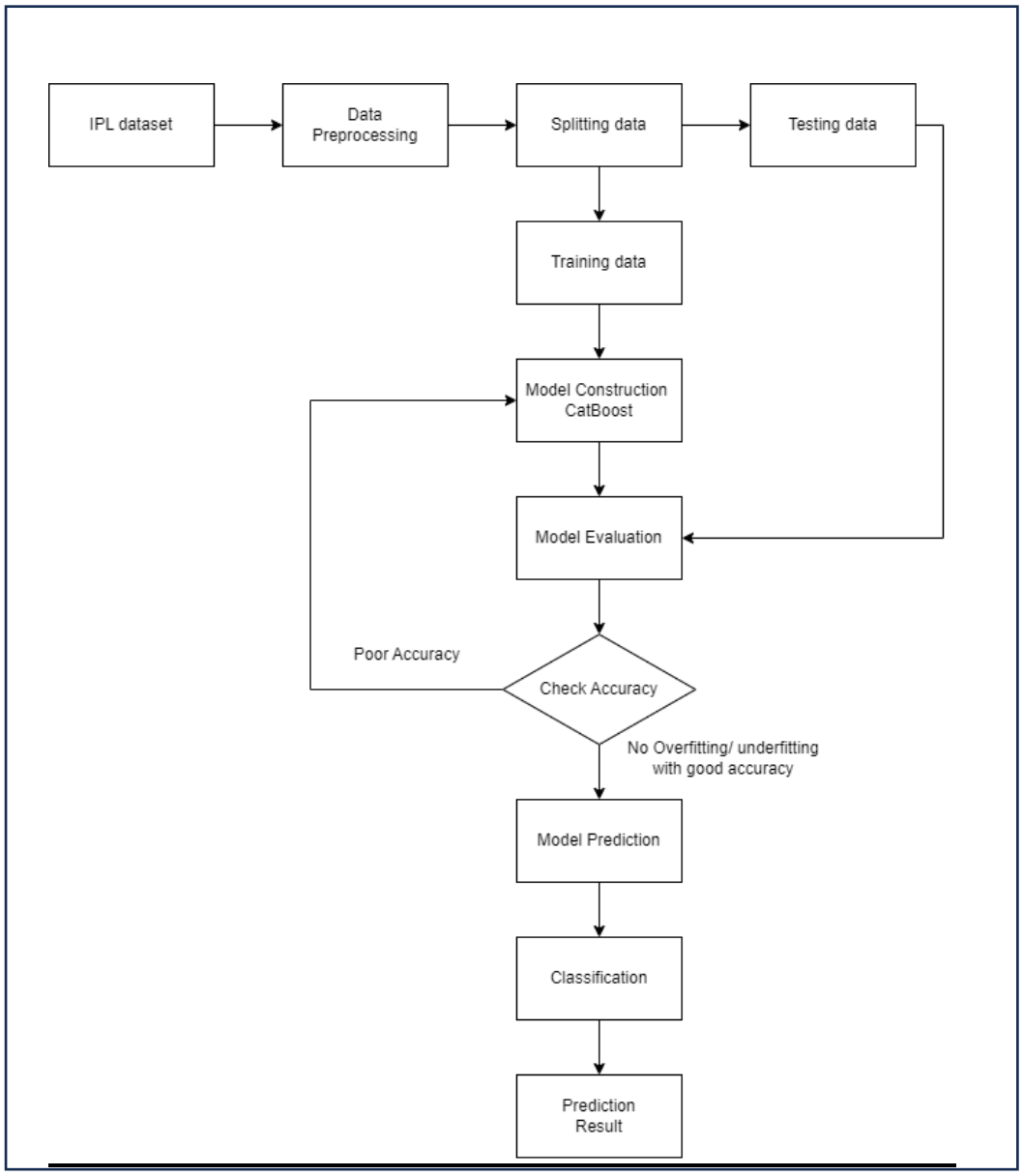


Fig 3: Process Flow Diagram

IPL Dataset: These datasets usually include a wide range of information such as match results, player statistics, team performances, venue details, and more, spanning multiple seasons of the Indian Premier League.

Data Preprocessing: Clean and preprocess the data to ensure it is ready for analysis. This includes removing duplicates, filling missing data, and standardizing the format of the data. This step is crucial to ensure the accuracy and reliability of the results.

Splitting Data: Divide the data into training and testing sets. The training set will be used to train the machine learning models, while the testing set will be used to evaluate the performance of the models. This step is important for ensuring that the models are able to generalize well to new, unseen data.

Model Training: Train machine learning models, CATBOOST model, on the training data. These models will learn patterns in the data that can be used to make predictions about the value of the player

Model Evaluation: Evaluate the performance of the models using the testing set. This will allow you to assess the accuracy of the models and identify areas for improvement.

Generating Models: Select the best-performing models to make predictions about the value of the player. These models can be integrated into the user-friendly platform to provide accurate and reliable predictions to users.

Result: Present the predictions to users through clear and easy-to-understand charts, graphs This will allow users to easily understand the predictions.

6.2 USE CASE DIAGRAM

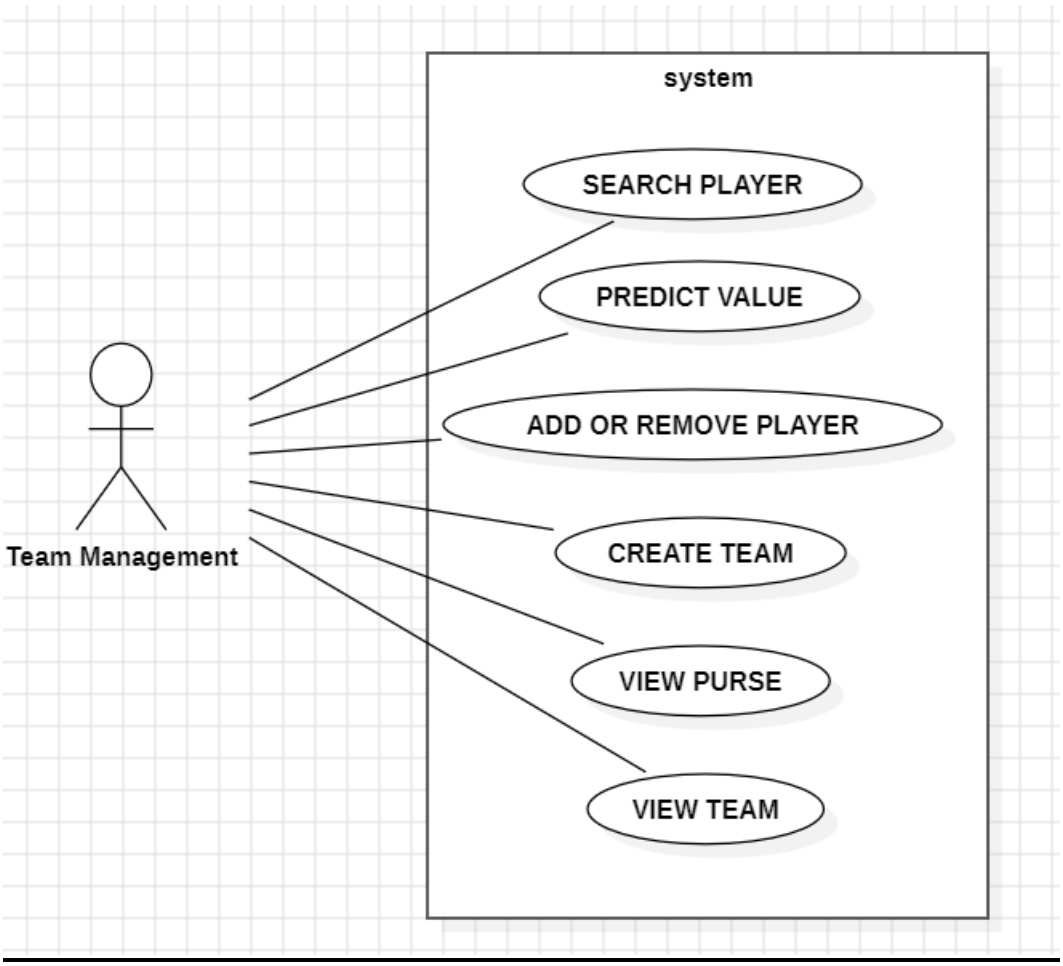


Fig 4: Use case diagram

7. IMPLEMENTATION

7.1 PSEUDO CODE

1. **Model preparation:**

Step 1:

```
Import Libraries
import pandas as pd
import warnings
warnings.simplefilter(action='ignore')
import matplotlib.pyplot as plt
from wordcloud import WordCloud
import seaborn as sns
from pycaret.regression import *
import pickle
```

Step 2:

```
Data Loading
ipl_08 = pd.read_csv("2008.csv")
ipl_09 = pd.read_csv("2009.csv")
ipl_10 = pd.read_csv("2010.csv")
ipl_11 = pd.read_csv("2011.csv")
ipl_12 = pd.read_csv("2012.csv")
ipl_13 = pd.read_csv("2013.csv")
ipl_14 = pd.read_csv("2014.csv")
ipl_15 = pd.read_csv("2015.csv")
ipl_16 = pd.read_csv("2016.csv")
ipl_17 = pd.read_csv("2017.csv")
ipl_18 = pd.read_csv("2018.csv")
ipl_19 = pd.read_csv("2019.csv")
ipl_20 = pd.read_csv("2020.csv")
ipl_21 = pd.read_csv("2021.csv")
ipl_22 = pd.read_csv("2022.csv")
ipl_23 = pd.read_csv("2023.csv")

Concatenate all yearly DataFrames
ipl_full = pd.concat([ipl_08, ipl_09, ipl_10, ipl_11, ipl_12, ipl_13,
ipl_14, ipl_15, ipl_16, ipl_17, ipl_18, ipl_19, ipl_20, ipl_21],
ignore_index=True)
```

```
Step 3: Data Preprocessing
Drop unnecessary columns
ipl_full.drop(['Rank'], axis=1, inplace=True)
Check for duplicates
duplicates = ipl_full[ipl_full.duplicated(keep=False)]

Generate and display word cloud for player names
player = ipl_full.Player.dropna()
text = player.explode().to_list()
wc = WordCloud(width=1500, height=1000).generate(' '.join(text))
plt.figure(figsize=(15, 15))
plt.axis('off')
plt.imshow(wc)
plt.show()
Drop 'Player' and 'Team' columns
ipl_full = ipl_full.drop(["Player", "Team"], axis=1)
Clean and convert 'Salary' and 'Value' columns to numeric types
ipl_full['Salary'] = ipl_full['Salary'].str.replace(',',
'').str.replace('$', '').astype(float)
ipl_full['Value'] = ipl_full['Value'].str.replace(',',
'').str.replace('$',    '').astype(float)

Convert 'RAA' column to numeric type
ipl_full["RAA"] = ipl_full["RAA"].astype(float)
```

Step 4:

```
Data Visualization
sns.pairplot(data=ipl_full)
plt.show()
```

Step 5:

```
Model Training
Set up PyCaret regression environment
reg = setup(data=ipl_full,
      target='Value',
      normalize=True,
      session_id=123)
Compare models to find the best one
best_model = compare_models()
Create and tune CatBoost model
catboost = create_model('catboost')
catboost_tune = tune_model(catboost)
Evaluate the tuned CatBoost model
evaluate_model(catboost_tune)
predict_model(catboost_tune)
```

Step 6:

```
Model Saving
Finalize the tuned CatBoost model
final_catboost = finalize_model(catboost_tune)
Save the finalized model using pickle
pickle.dump(final_catboost, open('ipl.pkl', 'wb'))
```

1. Search player:

```
Function search_player(player_name):
Load player data from the database or session
If player_name exists in player data:
        Return player details
Else:
        Return message "Player not found"
```

2. Predict Value:

```
Function predict_value(player_features):
```

```
Load the trained model
Convert player_features to the appropriate format
Predict player value using the model
Return the predicted value
```

3. Add or Remove Player:

```
Function add_player(player_name, player_value):
Load current team data from session
If player_count >= 25:
        Return message "Player count limit reached"
If current budget < player_value:
      Return message "Insufficient budget"
If player_name in team data:
        Return message "Player already exists"
Add player_name and player_value to team data
Decrease current budget by player_value
Increase player_count by 1
Save updated team data to session
Return success message "Player added successfully"

Function remove_player(player_name):
Load current team data from session
If player_name not in team data:
   Return message "Player not in team"
Remove player_name from team data
Increase current budget by player_value of removed player
Decrease player_count by 1
Save updated team data to session
Return success message "Player removed successfully"
```

4. Create Team:

```
Function create_team():
Initialize empty team data structure
Set initial budget
Set player_count to 0
Save initial team data to session
Return success message "Team created successfully"
```

5. View Purse:

```
Function view_purse():
Load current budget from session
Return current budget
```

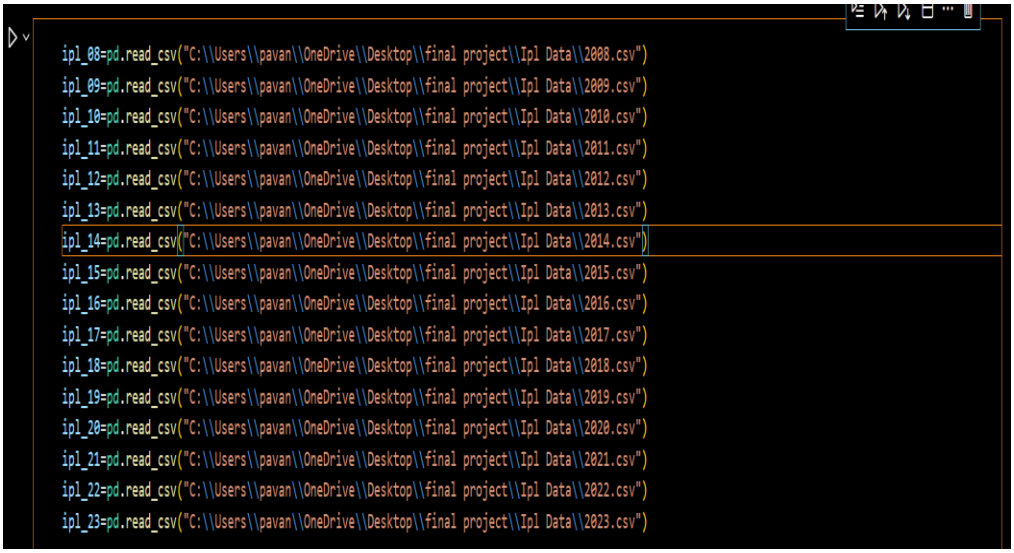
6. View Team:

```
Function view_team():
Load team data from session
Return team data including player names and their values
```

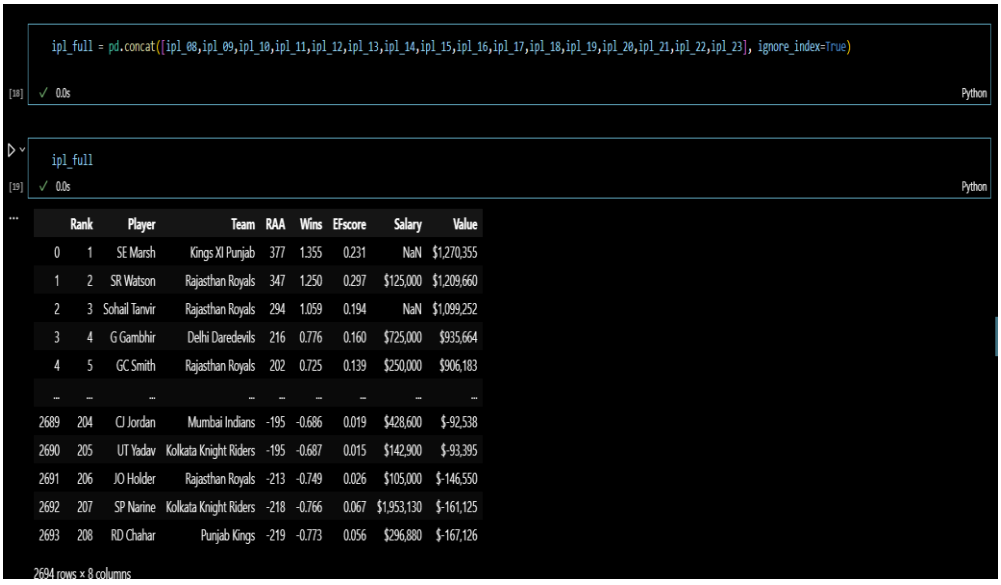
7.2Screenshots

Importing all the necessary libraries to import the dataset

Reading all of the dataset from the yr of 2008 to 2023 which incorporates gamers performance from past 15 years

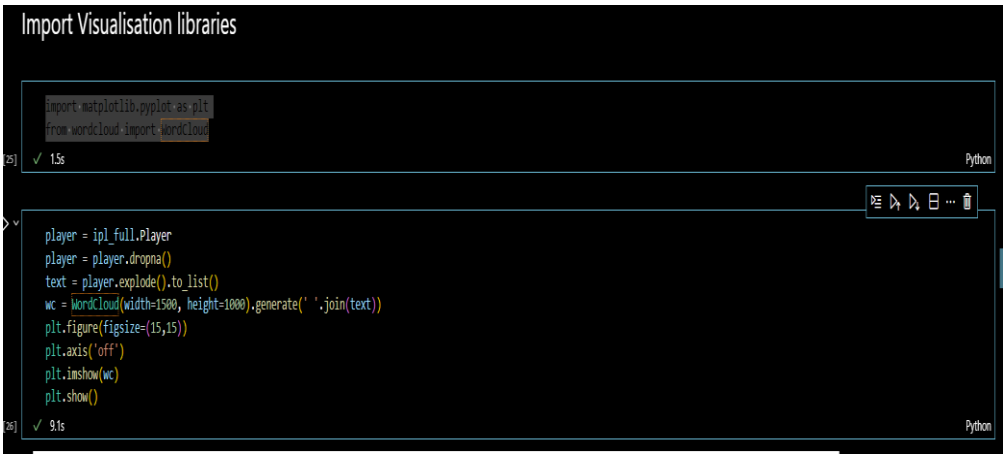


Concatenating all the dataset

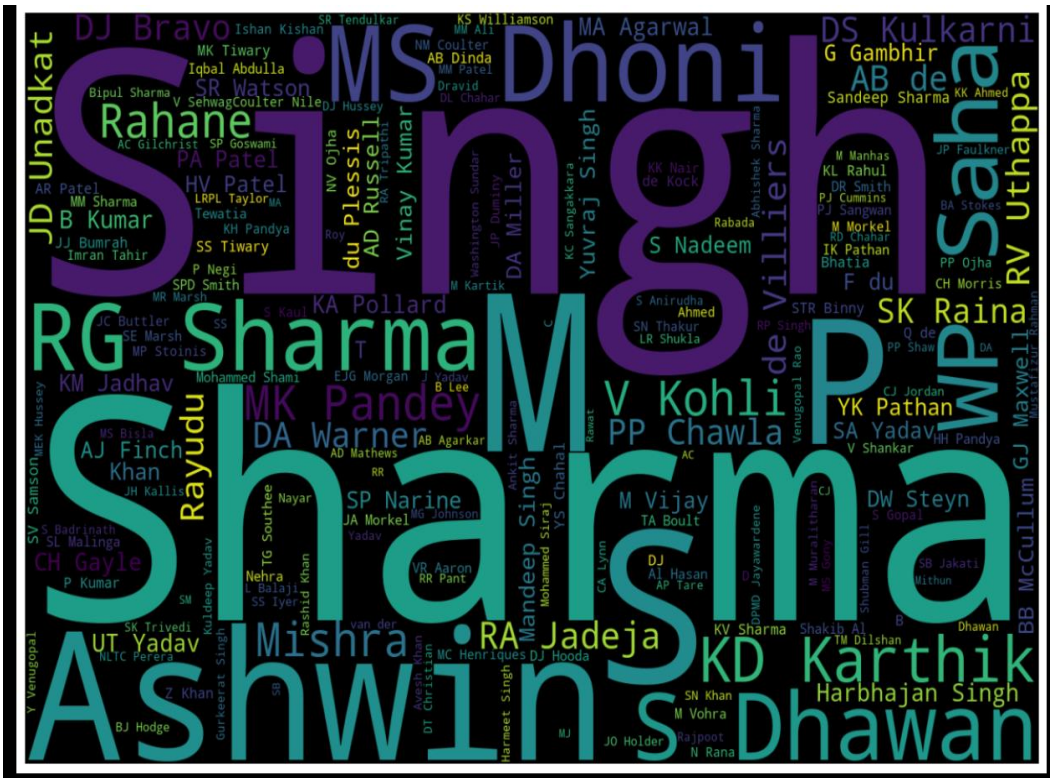


	Rank	Player	Team	RAA	Wins	EFscore	Salary	Value
0	1	SE Marsh	Kings XI Punjab	377	1.355	0.231	NaN	$1,270,355
1	2	SR Watson	Rajasthan Royals	347	1.250	0.297	$125,000	$1,209,660
2	3	Sohail Tanvir	Rajasthan Royals	294	1.059	0.194	NaN	$1,099,252
3	4	G Gambhir	Delhi Daredevils	216	0.776	0.160	$725,000	$935,664
4	5	GC Smith	Rajasthan Royals	202	0.725	0.139	$250,000	$906,183
...	...	...	...	...	...	...	...	...
2689	204	CJ Jordan	Mumbai Indians	-195	-0.686	0.019	$428,600	$-92,538
2690	205	UT Yadav	Kolkata Knight Riders	-195	-0.687	0.015	$142,900	$-93,395
2691	206	JO Holder	Rajasthan Royals	-213	-0.749	0.026	$105,000	$-146,550
2692	207	SP Narine	Kolkata Knight Riders	-218	-0.766	0.067	$1,953,130	$-161,125
2693	208	RD Chahar	Punjab Kings	-219	-0.773	0.056	$296,880	$-167,126

2694 rows × 8 columns



These lines of code installation the environment for growing word clouds the use of the WordCloud library and visualizing them using Matplotlib. Once imported, you may use the functionalities supplied by those libraries to create and personalize phrase clouds based on your textual content information and visualize them the use of Matplotlib's plotting talents.

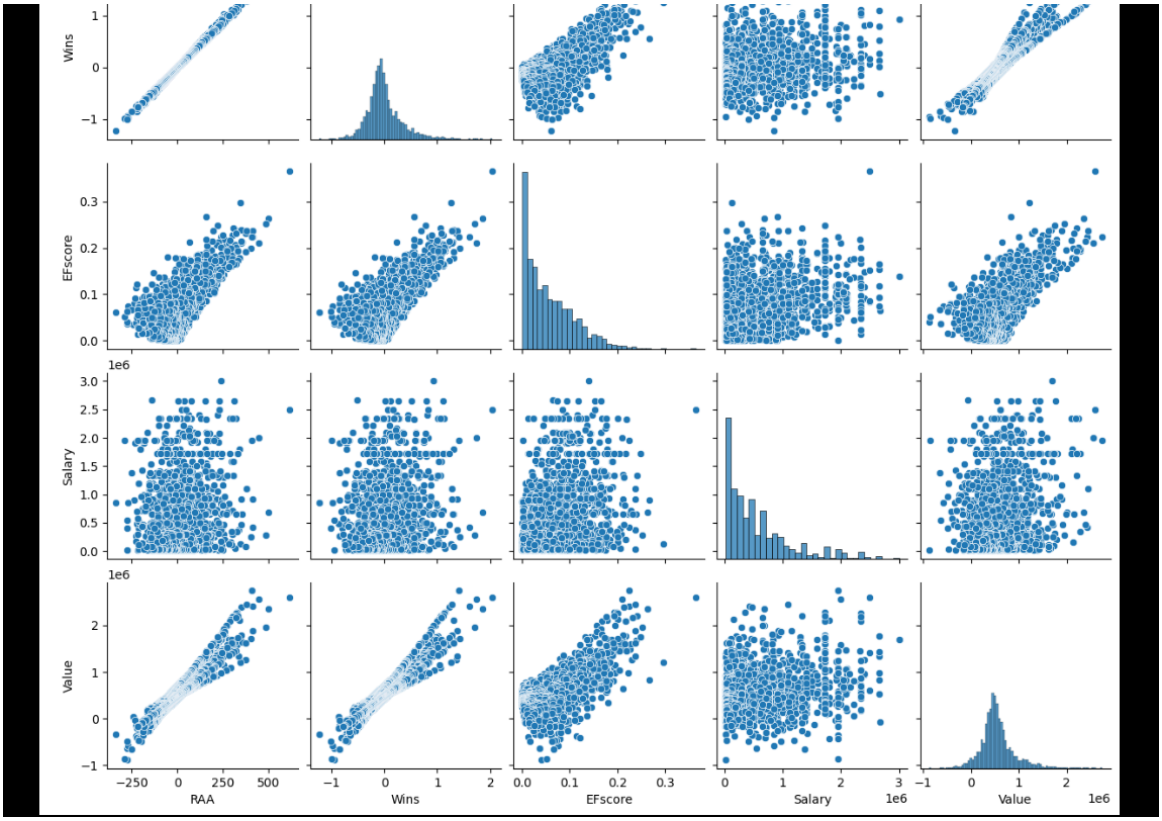


Use seaborn library for visualisation of the whether the data is normally distributed

```
[32]  import seaborn as sns
      ✓ 1.4s

[33]  sns.pairplot(data=ipl_full)
      ✓ 6.1s

...   <seaborn.axisgrid.PairGrid at 0x1ae94a9a0d0>
```

The Seaborn library to create a couple plot visualization based totally at the ipl_full Data Frame. The pair plot will incorporate scatterplots alongside the diagonal, displaying the distribution of each character variable. The off-diagonal plots can be scatterplots displaying the relationship between pairs of variables.

This visualization is particularly beneficial for exploring relationships between more than one variables in a dataset, identifying patterns, and detecting capacity correlations. It's a effective tool for preliminary exploratory statistics evaluation (EDA) and gaining insights into the structure of the data.



```
from pycaret.regression import *
```
✓ 3.4s

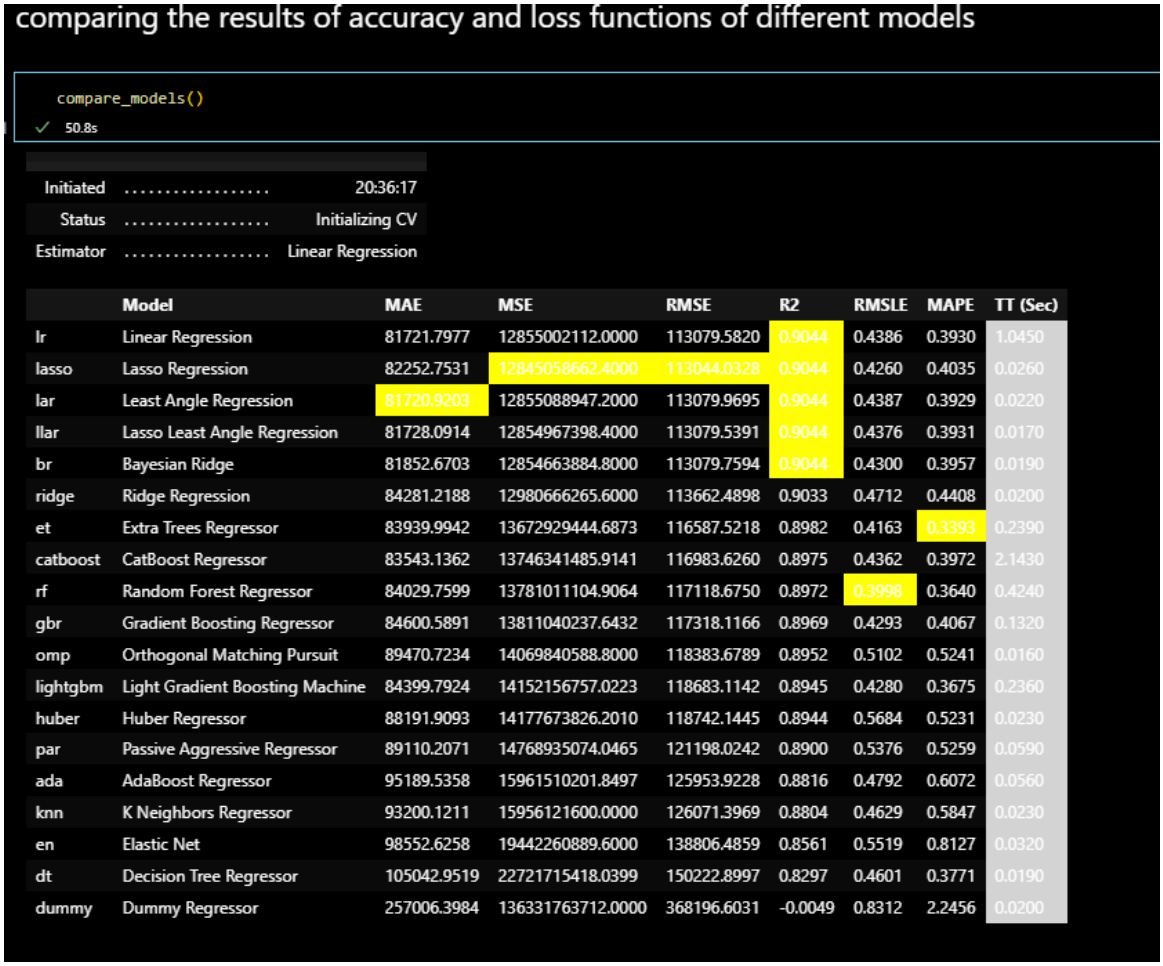
Pass the data and target feature to the pycaret model and specify required features for tuning

```
reg = setup(data = ipl_full,
            target = 'Value',
            normalize = True,
            session_id=123)
```
✓ 2.8s

	Description	Value
0	Session id	123
1	Target	Value
2	Target type	Regression
3	Original data shape	(2694, 5)
4	Transformed data shape	(2694, 5)
5	Transformed train set shape	(1885, 5)
6	Transformed test set shape	(809, 5)
7	Numeric features	4
8	Rows with missing values	28.6%
9	Preprocess	True
10	Imputation type	simple
11	Numeric imputation	mean
12	Categorical imputation	mode
13	Normalize	True
14	Normalize method	zscore
15	Fold Generator	KFold
16	Fold Number	10
17	CPU Jobs	-1
18	Use GPU	False
19	Log Experiment	False
20	Experiment Name	reg-default-name
21	USI	76c7

From pycaret.Regression import *: This line imports the necessary capabilities and instructions from the PyCaret library for regression tasks. PyCaret is a low-code gadget mastering library in Python that automates diverse steps inside the system mastering workflow, making it less complicated to carry out obligations consisting of facts preprocessing, version training, and evaluation.

After jogging this code, the reg object will include all the preprocessed records, together with the transformed features, the goal variable, and different information needed for regression modeling. This preprocessed information can then be used to educate and examine regression fashions using PyCaret's automatic workflow.

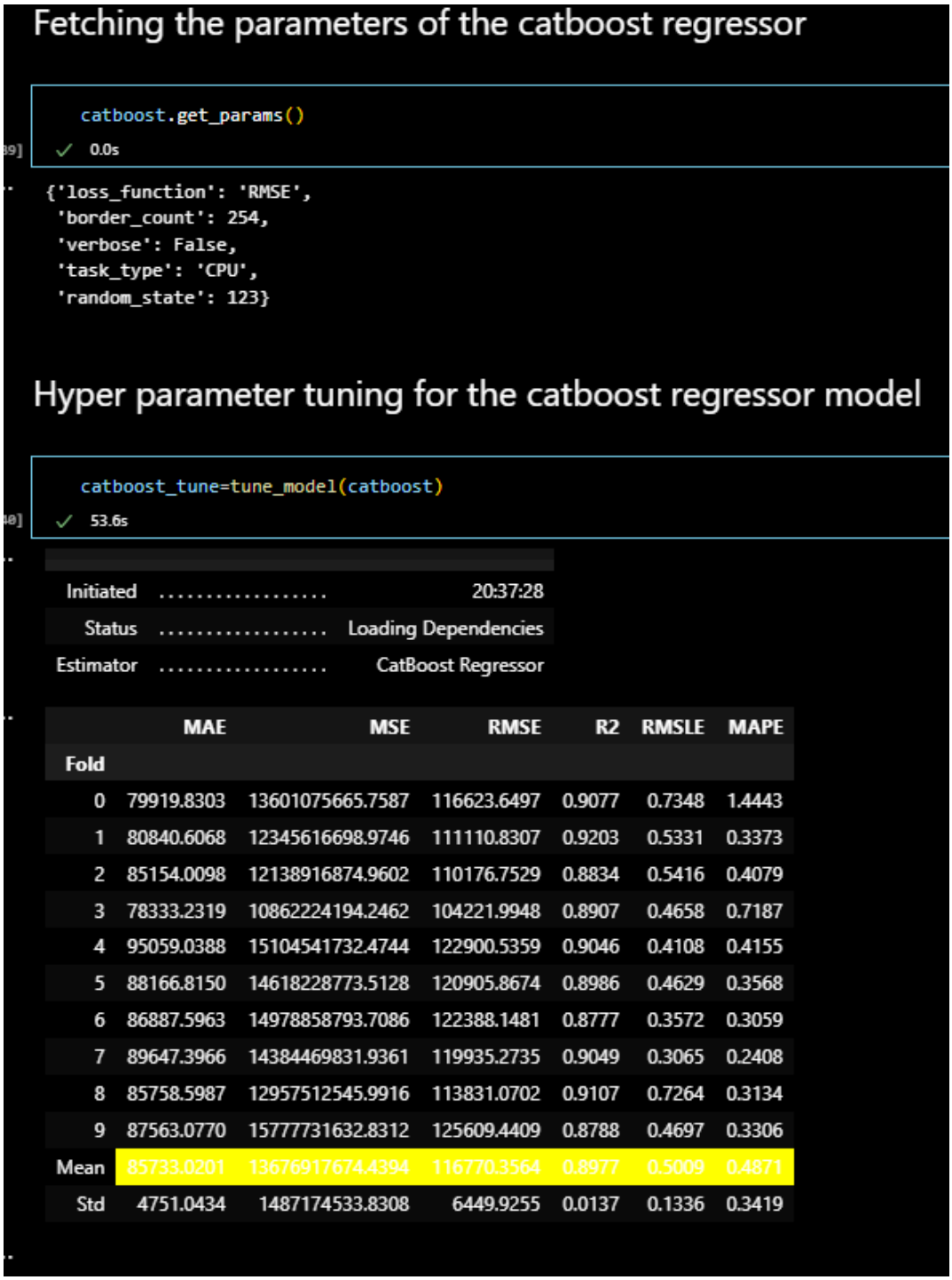


	Model	MAE	MSE	RMSE	R2	RMSLE	MAPE	TT (Sec)
lr	Linear Regression	81721.7977	12855002112.0000	113079.5820	0.9044	0.4386	0.3930	1.0450
lasso	Lasso Regression	82252.7531	12845058662.4000	113044.0328	0.9044	0.4260	0.4035	0.0260
lar	Least Angle Regression	81720.9203	12855088947.2000	113079.9695	0.9044	0.4387	0.3929	0.0220
llar	Lasso Least Angle Regression	81728.0914	12854967398.4000	113079.5391	0.9044	0.4376	0.3931	0.0170
br	Bayesian Ridge	81852.6703	12854663884.8000	113079.7594	0.9044	0.4300	0.3957	0.0190
ridge	Ridge Regression	84281.2188	12980666265.6000	113662.4898	0.9033	0.4712	0.4408	0.0200
et	Extra Trees Regressor	83939.9942	13672929444.6873	116587.5218	0.8982	0.4163	0.3393	0.2390
catboost	CatBoost Regressor	83543.1362	13746341485.9141	116983.6260	0.8975	0.4362	0.3972	2.1430
rf	Random Forest Regressor	84029.7599	13781011104.9064	117118.6750	0.8972	0.3998	0.3640	0.4240
gbr	Gradient Boosting Regressor	84600.5891	13811040237.6432	117318.1166	0.8969	0.4293	0.4067	0.1320
omp	Orthogonal Matching Pursuit	89470.7234	14069840588.8000	118383.6789	0.8952	0.5102	0.5241	0.0160
lightgbm	Light Gradient Boosting Machine	84399.7924	14152156757.0223	118683.1142	0.8945	0.4280	0.3675	0.2360
huber	Huber Regressor	88191.9093	14177673826.2010	118742.1445	0.8944	0.5684	0.5231	0.0230
par	Passive Aggressive Regressor	89110.2071	14768935074.0465	121198.0242	0.8900	0.5376	0.5259	0.0590
ada	AdaBoost Regressor	95189.5358	15961510201.8497	125953.9228	0.8816	0.4792	0.6072	0.0560
knn	K Neighbors Regressor	93200.1211	15956121600.0000	126071.3969	0.8804	0.4629	0.5847	0.0230
en	Elastic Net	98552.6258	19442260889.6000	138806.4859	0.8561	0.5519	0.8127	0.0320
dt	Decision Tree Regressor	105042.9519	22721715418.0399	150222.8997	0.8297	0.4601	0.3771	0.0190
dummy	Dummy Regressor	257006.3984	136331763712.0000	368196.6031	-0.0049	0.8312	2.2456	0.0200

The characteristic compares the overall performance of different regression models using default hyperparameters and evaluates their performance the usage of move-validation.

After execution, PyCaret generates a desk showing diverse performance metrics for each model evaluated, permitting you to compare their overall performance and become aware of the pinnacle-acting ones based on the chosen metric.

Overall, this code snippet simplifies the technique of evaluating regression fashions with the aid of automating the schooling and evaluation system, making it simpler for customers to discover the maximum appropriate model for their regression challenge.



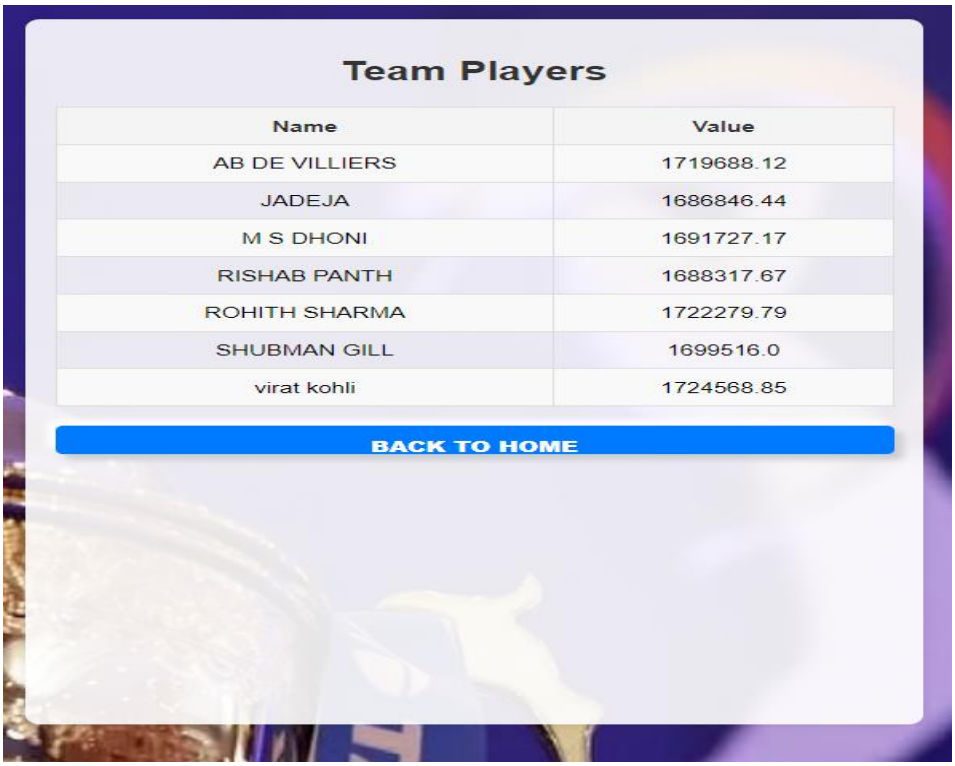
the process of creating and training a CatBoost regression model by automating the training process and providing a trained model object that can be used for various tasks, such as making predictions or evaluating model performance.

```python
import numpy as np
import pickle
import catboost
from catboost import CatBoostRegressor

from flask import Flask, request, render_template, session, redirect, url_for

app = Flask(__name__, template_folder='template')
app.secret_key = 'key'
model = pickle.load(open('ipl.pkl', 'rb'))

@app.route('/')
@app.route('/')
def home():
    session['nameVal'] = {}
    session['curVal'] = "${:,.2f}".format(12030111.84)
    session['playerCount'] = 0
    return render_template("index.html", nameVal=session['nameVal'], curVal=session['curVal'], playerCount=session['playerCount'])

@app.route('/predict', methods=['POST'])
def predict():
    int_features = list(request.form.values())
    name = int_features[0]
    int_features = int_features[1:]
    int_features = [float(x) for x in int_features]
    final_features = [np.array(int_features)]
    prediction = model.predict(final_features)
    output = round(prediction[0], 2)

    session['name'] = name
    session['output'] = output
    return render_template('index.html', name='{}'.format(name), prediction_value='Prediction value : ${}'.format(output), curVal=session['curVal'], playerCount=session['

@app.route('/discard', methods=['GET', 'POST'])
def discard():
```

```python
@app.route('/discard', methods=['GET', 'POST'])
def discard():
    return render_template('index.html', name='', prediction_value='Prediction value : $', playerCount=session['playerCount'], curVal=session['curVal'])

@app.route('/addPlayer', methods=['GET', 'POST'])
def addPlayer():
    output = session.get('output', 0)
    name = session.get('name')

    nameVal = session.get('nameVal', {})

    if session['playerCount'] >= 25:
        flag = "Player count limit reached. Cannot add more players."
        return render_template('index.html', name='{}'.format(name), prediction_value='Prediction value : ${}'.format(output), nameVal=session['nameVal'], playerCount=ses

    curVal_str = session.get('curVal', '0').replace('$', '').replace(',', '')
    try:
        curVal = float(curVal_str)
    except ValueError:
        curVal = 0.0

    if curVal - output < 0:
        flag = "Insufficient budget to add this player."
        return render_template('index.html', name='{}'.format(name), prediction_value='Prediction value : ${}'.format(output), nameVal=session['nameVal'], playerCount=ses

    if name not in nameVal:
        session['playerCount'] = session.get('playerCount', 0) + 1
        curVal -= output
        session['curVal'] = "${:,.2f}".format(curVal)
    else:
        flag = f"{name} already exists!"
        return render_template('index.html', name='{}'.format(name), prediction_value='Prediction value : ${}'.format(output), nameVal=session['nameVal'], playerCount=ses

    nameVal[name] = output
```

(import) render_template: Any

```python
    if name not in nameVal:
        session['playerCount'] = session.get('playerCount', 0) + 1
        curVal -= output
        session['curVal'] = "${:,.2f}".format(curVal)
    else:
        flag = f"{name} already exists!"
        return render_template('index.html', name='{}'.format(name), prediction_value='Prediction value : ${}'.format(output), nameVal=session['nameVal'], playerCount=ses

    nameVal[name] = output
    session['nameVal'] = nameVal

    return render_template('index.html', name='{}'.format(name), prediction_value='Prediction value : ${}'.format(output), nameVal=session['nameVal'], playerCount=session

@app.route('/reset', methods=['GET', 'POST'])
def reset():
    session['nameVal'] = {}
    session['curVal'] = "${:,.2f}".format(12030111.84)
    session['playerCount'] = 0
    return redirect(url_for('home'))

@app.route('/view_team')
def view_team():
    nameVal = session.get('nameVal', {})
    curVal = session.get('curVal', 0)
    return render_template('view_team.html', nameVal=nameVal, curVal=curVal)

@app.route('/home_view')
def home_view():
    nameVal = session.get('nameVal', {})
    curVal = session.get('curVal', 0)
    return render_template('index.html', nameVal=nameVal, curVal=curVal, playerCount=session['playerCount'])

if __name__ == "__main__":
    app.run(debug=True)
```



8. SOFTWARE TESTING

8.1MANUAL TEST CASES

Test Case ID	Step Details	Expected Action	Actual Action	Pass/Fail
TC 01	Checking for the values In range	Alerts should be displayed if any value is out of the acceptable range.	Alerts is displayed	Pass
TC 02	Adding a Player above Player Limit	A flag message "Player count limit reached. Cannot add more players." Should be displayed.	Flag message is displayed	Pass
TC 03	Add a new player when the budget is insufficient	A flag message "Insufficient budget to add this player." Should be displayed	Flag message is displayed	Pass
TC 04	Add a player who is already in the team	A flag message "{name} already exists!" should be displayed.	Flag message is displayed	Pass

9. CONCLUSION

In this study, a machine is evolved and evaluated numerous device learning fashions to are expecting the promoting fees of gamers in the Indian Premier League (IPL) auction, leveraging past overall performance metrics along with

runs, balls confronted, innings, wickets, and fits performed. Our evaluation encompassed quite a few algorithms, inclusive of Decision Tree Regressor, K-Nearest Neighbors (KNN), Linear Regression, Random Forest Regressor, and Catboost. Among those, Catboost and Linear Regression emerged as the best models for predicting the public sale expenses of batsmen and bowlers, respectively. These models provide auctioneers with treasured insights, enabling them to make informed and rapid selections at some stage in the public sale process. Furthermore, by means of incorporating an inflation thing and mapping it to the price range during model schooling, we ensured that our predictions remain relevant and accurate in the context of converting monetary situations. Overall, our technique demonstrates the capacity of machine getting to know in enhancing the efficiency and accuracy of participant valuation in sports activities auctions.

10. FUTURE ENHANCEMENTS

Advanced Statistics: Incorporate advanced cricket metrics including strike rate, financial system fee, and participant health facts.
Contextual Performance Data: Include context-based totally overall performance metrics like overall performance in stress situations, consistency

throughout exceptional pitches and weather conditions.
Player Form and Fitness: Add latest shape, health reviews, and injury records to higher expect a player's future performance.
Market Sentiment Analysis: Incorporate market sentiment analysis the usage of social media and information to seize the public perception and call for for gamers.
Sponsor Influence: Analyze the impact of sponsorship deals and endorsements on participant price.
Generalization: Adapt and generalize the model to are expecting participant auction prices in other sports activities leagues (NBA, NFL) via incorporating recreation-unique overall performance metrics.

APPENDIX:

Appendix A: Bibliography

Paper 1: https://ieeexplore.ieee.org/document/9198668

Paper 2: https://ieeexplore.ieee.org/document/10250755

Paper 3: https://ieeexplore.ieee.org/document/10127503

Paper 4: https://ieeexplore.ieee.org/document/8820235

Paper 5: https://ieeexplore.ieee.org/document/9198371

Appendix C: Plagiarism Report

Appendix D: Poster

IPL AUCTION PREDICTION

BY: PAVAN CHAKRASALI SRN: PES1PG22CA139

GUIDE: PROF SANTOSH KATTI

PES UNIVERSITY

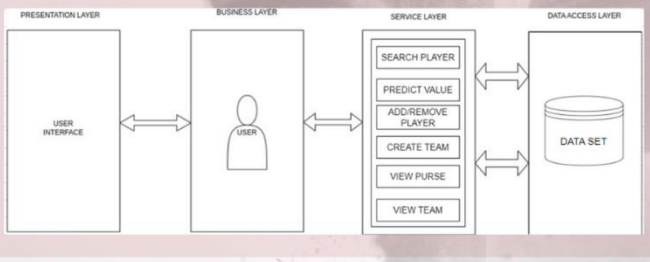
CELEBRATING 50 YEARS

IPL

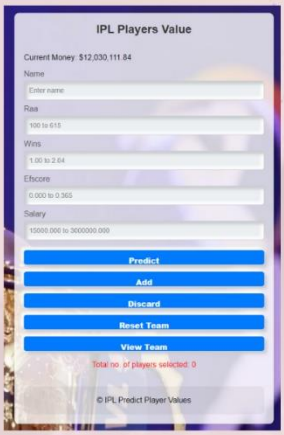
ABSTRACT

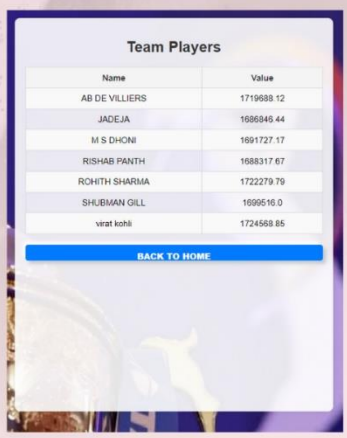
This work applies machine learning algorithms to predict player auction prices in the Indian Premier League based on past performance metrics such as runs, balls, innings, wickets, and matches played. We tested various models, including Decision Tree Regressor, K-Nearest Neighbors, Linear Regression, Random Forest Regressor, and Support Vector Regression. CatBoost and Linear Regression yielded the best results for batsmen and bowlers, respectively, aiding quick decision-making for auctioneers. The model also accounts for inflation and budget mapping.

ARCHITECTURE



RESULTS





TOOLS AND TECHNOLOGIES

Visual Studio Code

CONCLUSION

The integration of machine learning and data analytics in IPL auction prediction is set to revolutionize player valuation and team composition