PROJECT DOCUMENTATION

EXPLORATORY DATA ANALYSIS USING PYTHON

AND MACHINE LEARNING

|  |
| --- |
| Title: IPL Match Records |
| Name: Harish P |
| Course: DA/DS, Offline |
| Batch: June 2025 |

TABLE OF CONTENT

|  |  |
| --- | --- |
| 1. | **Introduction** |
| 2. | **Aim** |
| 3. | **Problem Statements** |
| 4. | **Project Workflow** |
| 5. | **Data Understanding** |
| 6. | **Data Cleaning** |
| 7. | **Exploratory Data Analysis (EDA) - Univariate Analysis** |
| 8. | **Bivariate Analysis** |
| 9. | **Multivariate Analysis** |
| 10. | **Overall Insights from Analysis** |
| 11. | **Model Training** |
| 12. | **Conclusion** |

**1.Introduction**

The Indian Premier League (IPL) is one of the most exciting and popular cricket tournaments in the world. It brings together top players from different countries and creates a fast-paced, high-scoring format that attracts millions of fans each year.

This project focuses on analyzing IPL match records to understand different patterns and trends in the data. By studying past match information such as teams, venues, toss results, winners, and player performances, we can gain useful insights about what factors influence the outcome of matches.

Data analysis helps us explore team strengths, key players, venue advantages, and seasonal changes in performance. These insights can be useful for teams, analysts, and cricket fans to better understand the game.

**2.Aim**

The main aim of this project is to analyze IPL match data to find meaningful patterns and insights.  
Through data cleaning and exploratory data analysis, we aim to:

* Understand the performance of different teams and players.
* Identify factors that influence match outcomes, such as toss decisions and venues.
* Explore seasonal trends and winning patterns.
* Gain insights that can help in making better predictions and understanding the game more deeply.

**3.Problem Statements**

The IPL dataset contains a large amount of information about matches, teams, players, and venues. However, without proper analysis, it is difficult to understand the hidden patterns and factors that affect match outcomes.

There is a need to:

* Identify key factors that influence winning and performance.
* Understand team and player trends over different seasons.
* Analyze how toss decisions and venues impact match results.
* Organize and clean the data to make it suitable for meaningful analysis.

By solving these problems, we can gain valuable insights from the IPL data.

**4.PROJECT WORKFLOW**

1. **Introduction**
   * Notebook starts with a markdown header: *"IPL Dataset"*.
2. **Library Setup**
   * Imports required libraries: pandas, numpy, matplotlib, seaborn, warnings.
3. **Data Loading**
   * Reads the dataset from ipl\_matches.csv.
   * Displays the dataset (df, head(), tail()).
4. **Initial Data Exploration**
   * Checks dataset info (info(), shape, columns).
   * Inspects null values and data types.
   * Describes numerical features (describe()).
5. **Data Cleaning & Preprocessing**
   * Handles missing values.
   * Drops/renames irrelevant columns (if any).
   * Encodes categorical columns (like teams, venues, etc.).
6. **Exploratory Data Analysis (EDA)**
   * Visualizations using matplotlib and seaborn:
     + Distribution of matches per season.
     + Toss decisions analysis.
     + Most successful teams/players.
     + Winning by runs/wickets trends.
     + Venue-wise match statistics.
7. **Advanced Analysis**
   * Player performance (e.g., top batsmen, bowlers).
   * Team performance trends across seasons.
   * Toss vs. match outcome correlation.
8. **Visual Insights**
   * Heatmaps, bar plots, and line charts for comparative insights.
   * Correlation matrix for numerical features.
9. **Summary & Conclusions**
   * Key findings from IPL data trends.
   * Insights on teams, players, and match outcomes.

5.**Data Understanding**

**Dataset Size**

* Typically includes **thousands of rows** (each row = one match or delivery, depending on dataset version).
* Columns describe match details (teams, players, venue, outcomes).

**Common Columns**

* id: Match ID (unique identifier).
* season: IPL season/year.
* city: City where the match was played.
* date: Match date.
* team1, team2: Competing teams.
* toss\_winner: Team winning the toss.
* toss\_decision: Bat or field decision.
* result: Type of result (win, tie, no result).
* winner: Winning team.
* win\_by\_runs, win\_by\_wickets: Margin of victory.
* player\_of\_match: Best performer.
* venue: Stadium/ground.
* umpire1, umpire2: Match officials.

**Data Types**

* season, win\_by\_runs, win\_by\_wickets: Numeric (integer).
* date: Date/time.
* teams, venue, winner, player\_of\_match: Categorical (string).

**Key Insights Possible**

* Which teams/players dominate across seasons.
* Toss impact on match outcomes.
* Venue-specific performance trends.
* Winning patterns (by runs vs. by wickets).

**6. Data Cleaning**

**Remove Duplicates**

* **Drop duplicate rows if any:**
* **df.drop\_duplicates(inplace=True)**

**Handle Missing Values**

* **Check null values:**
* **df.isnull().sum()**
* **Fill/drop depending on column:**
  + **umpire1, umpire2: If missing, can be filled with "Unknown".**
  + **winner: If missing → mark as "No Result".**
  + **player\_of\_match: If missing → "NA".**
  + **Drop columns with too many missing values (e.g., umpire3 in some datasets).**

**Standardize Team Names**

* **Old teams sometimes appear with different spellings (e.g., "Delhi Daredevils" vs. "Delhi Capitals").**
* **Use a dictionary to map:**
* **team\_map = {**
* **"Delhi Daredevils": "Delhi Capitals",**
* **"Deccan Chargers": "Sunrisers Hyderabad",**
* **# add more if needed**
* **}**
* **df['team1'] = df['team1'].replace(team\_map)**
* **df['team2'] = df['team2'].replace(team\_map)**
* **df['winner'] = df['winner'].replace(team\_map)**
* **df['toss\_winner'] = df['toss\_winner'].replace(team\_map)**

**Convert Data Types**

* **Convert date column to datetime:**
* **df['date'] = pd.to\_datetime(df['date'])**
* **Ensure numerical columns (win\_by\_runs, win\_by\_wickets, season) are integers.**

**Feature Engineering (Optional)**

* **Create new features for analysis:**
  + **win\_type: "Runs" if win\_by\_runs > 0, "Wickets" if win\_by\_wickets > 0.**
  + **is\_home\_team: whether the winner played in home city.**
  + **match\_decision: combine toss + match outcome.**

**Drop Irrelevant Columns**

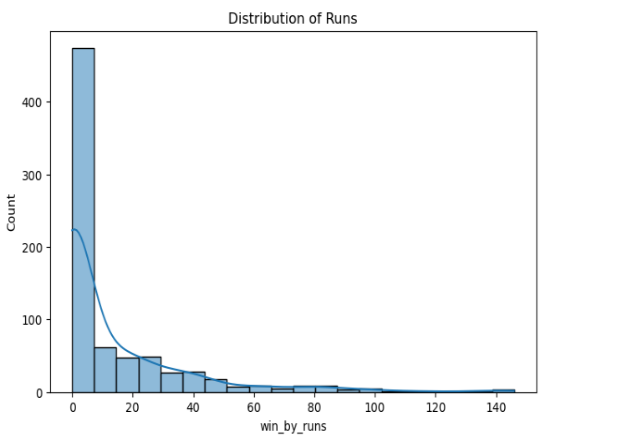
* **Columns like id (if not needed), or extra umpire columns with mostly nulls.**

**7.EXPLORATORY DATA ANALYSIS (EDA)**

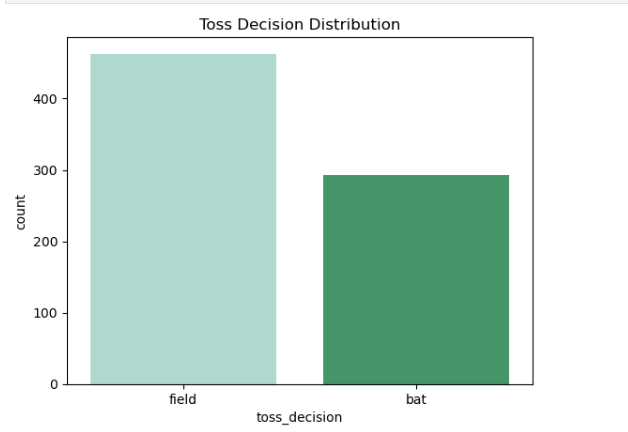
**UNIVARIATE ANALYSIS**

Univariate analysis focuses on exploring individual variables to understand their distribution, patterns, and anomalies.

|  |  |  |
| --- | --- | --- |
| **Variable** | **Chart Type** | **Key Insights** |
| **Win\_by\_runs** | **Hist plot** | **Most matches are won by small runs** |
| **Team 1** | **Count plot** | **Mumbai Indians played the most matches** |
| **Toss-Decesion** | **Count plot** | **Most teams choose to field after winning the toss** |
| **Toss-**  **winner** | **Pie plot** | **Mumbai Indians won the most tosses** |

** A graph of different colored bars

AI-generated content may be incorrect.**

** A colorful pie chart with text

AI-generated content may be incorrect.**

**Insights Gained:**

* Most matches are won by small runs
* Big-run wins are rare
* Mumbai Indians played the most matches
* Kings XI Punjab, CSK, and RCB also played many matches
* Mumbai Indians won the most tosses
* KKR and CSK also won many tosses

**8. BIVARIATE ANALYSIS**

Bivariate analysis helps explore relationships between two variables, uncovering trends, correlations, and dependencies that impact housing prices

|  |  |  |
| --- | --- | --- |
| Variable | Chart Type | Key Insights |
| Toss Decessiont by Result Type | Count Plot | Most matches are normal when teams choose to field |
| Toss Winner by Match Winner | Count Plot | Mumbai Indians won the most matches |
| Total by Team | Count Plot | Sometimes the toss winner also wins the match |

A graph of a game

AI-generated content may be incorrect. A graph with text and numbers

AI-generated content may be incorrect.

A graph with pink bars and black text

AI-generated content may be incorrect.

Insights Gained:

* Most matches are normal when teams choose to field
* Fewer matches are normal when teams choose to bat
* Mumbai Indians won the most matches
* CSK and KKR also won many matches
* CSK and KKR also won many matches
* Small teams like Kochi and Supergiants won very few matches

9. **MULTIVARAIATE ANALYSIS**

**Insights:**

* Runs and wickets have a strong negative link (If one goes up, the other goes down)
* Other values have almost no link (Very weak or no connection)
* Chris Gayle got the most awards
* AB de Villiers, Dhoni, and Rohit also got many awards
* Others got less but still good

A screenshot of a graph

AI-generated content may be incorrect.A graph of a match

AI-generated content may be incorrect.

**10.OVERALL INSIGHTS FROM ANALYSIS**

1.Univariate Analysis

* Most matches are won by small runs
* Big-run wins are rare
* Mumbai Indians played the most matches
* Kings XI Punjab, CSK, and RCB also played many matches
* Mumbai Indians won the most tosses
* KKR and CSK also won many tosses

2. Bivariate Analysis

* Most matches are normal when teams choose to field
* Fewer matches are normal when teams choose to bat
* Mumbai Indians won the most matches
* CSK and KKR also won many matches
* CSK and KKR also won many matches
* Small teams like Kochi and Supergiants won very few matches

3. Multivariate Analysis

* Runs and wickets have a strong negative link (If one goes up, the other goes down)
* Other values have almost no link (Very weak or no connection)
* Chris Gayle got the most awards
* AB de Villiers, Dhoni, and Rohit also got many awards
* Others got less but still good

**12. Model Training**

**1. Logistic Regression (Multiclass)**

* Encode categorical columns using LabelEncoder.
* Split data into train/test sets (stratified).
* Scale features with StandardScaler.
* Train using LogisticRegression(max\_iter=1000, multi\_class="multinomial").
* Evaluate with **accuracy, classification report, and confusion matrix**.

A screenshot of a computer program

AI-generated content may be incorrect.

**2. Decision Tree Classifier**

* Drop irrelevant features (date, player\_of\_match, venue).
* Encode categorical features, label encode target.
* Train/test split.
* Train using DecisionTreeClassifier(random\_state=42).
* Evaluate with **accuracy, classification report, confusion matrix**.
* Extract **feature importances**.

A screenshot of a computer program

AI-generated content may be incorrect.

**3. Random Forest Classifier**

* Encode features and target labels.
* Train/test split.
* Train using RandomForestClassifier(n\_estimators=200, random\_state=42).
* Evaluate with **accuracy, classification report, confusion matrix**.
* Extract **top feature importances**.

A screenshot of a computer code

AI-generated content may be incorrect.

**4. Support Vector Machine (SVC)**

* Encode categorical features, label encode target.
* Split into train/test.
* Scale features with StandardScaler.
* Train using SVC(kernel="rbf", probability=True).
* Evaluate with **accuracy, classification report, confusion matrix**.

A screenshot of a computer program

AI-generated content may be incorrect.

**5. K-Nearest Neighbors (KNN)**

* Encode categorical features, label encode target.
* Train/test split.
* Scale features with StandardScaler.
* Train using KNeighborsClassifier(n\_neighbors=5).
* Evaluate with **accuracy, classification report, confusion matrix**.

A screenshot of a computer code

AI-generated content may be incorrect.

**6. Gradient Boosting Classifier**

* Encode categorical features and target.
* Train/test split.
* Train using GradientBoostingClassifier(n\_estimators=200, learning\_rate=0.1, random\_state=42).
* Evaluate with **accuracy, classification report, confusion matrix**.

A screenshot of a computer program

AI-generated content may be incorrect.

12. Conclusion

This project shows how IPL match data can be analyzed to find meaningful patterns. By using data cleaning and EDA techniques, we can understand team performance, player contributions, and other key factors. This type of analysis can help teams, fans, and analysts make better decisions and predictions. In the future, this project can be extended with machine learning to predict match winners or player performance.