A2-Python

June 19, 2025

[27]:
```
# ## 1. Suppress Warnings
import warnings
warnings.filterwarnings("ignore", category=RuntimeWarning)
```

[28]:
```
# ## 2. Import Required Libraries
import pandas as pd
import os
import scipy.stats as st
from fuzzywuzzy import process
```

[29]:
```
# ## 3. Set Paths for Dataset Files
BASE = os.getcwd()
DATASETS_DIR = os.path.join(BASE, 'datasets')

PERF_DATASET_NAME = 'IPL_ball_by_ball_updated till 2024.csv'
PERF_DATASET_PATH = os.path.join(DATASETS_DIR, PERF_DATASET_NAME)

SALARY_DATASET_NAME = 'IPL SALARIES 2024.xlsx'
SALARY_DATASET_PATH = os.path.join(DATASETS_DIR, SALARY_DATASET_NAME)
```

[30]:
```
# ## 4. Read IPL Performance Data
df = pd.read_csv(PERF_DATASET_PATH, low_memory=False)
```

[31]:
```
df.head(3)  # Preview the first few rows
```

[31]:

	Match id	Date	Season	Batting team	Bowling team	Innings No	Ball No	Bowler	Striker	Non Striker	runs_scored	extras	type of extras	score	score/wicket
0	335982	18-04-2008	2007/08	Kolkata Knight Riders	Royal Challengers Bangalore	1	0.1	P Kumar	SC Ganguly	BB McCullum	0	1	legbyes	1	1/0
1	335982	18-04-2008	2007/08	Kolkata Knight Riders	Royal Challengers Bangalore	1	0.2	P Kumar	BB McCullum	SC Ganguly	0	0	NaN	1	1/0
2	335982	18-04-2008	2007/08	Kolkata Knight Riders	Royal Challengers Bangalore	1	0.2	P Kumar	BB McCullum	SC Ganguly	0	0	NaN	1	1/0

```
2   SC Ganguly            0       1          wides      2          2/0

   wicket_confirmation wicket_type fielders_involved Player Out
0                    0         NaN               NaN        NaN
1                    0         NaN               NaN        NaN
2                    0         NaN               NaN        NaN
```

[32]:
```
df.columns  # Check column names
```

[32]:
```
Index(['Match id', 'Date', 'Season', 'Batting team', 'Bowling team',
       'Innings No', 'Ball No', 'Bowler', 'Striker', 'Non Striker',
       'runs_scored', 'extras', 'type of extras', 'score', 'score/wicket',
       'wicket_confirmation', 'wicket_type', 'fielders_involved',
       'Player Out'],
      dtype='object')
```

[33]:
```
df.info()  # Summary info about columns and types
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 255759 entries, 0 to 255758
Data columns (total 19 columns):
 #   Column               Non-Null Count   Dtype
---  ------               --------------   -----
 0   Match id             255759 non-null  int64
 1   Date                 255759 non-null  object
 2   Season               255759 non-null  object
 3   Batting team         255759 non-null  object
 4   Bowling team         255759 non-null  object
 5   Innings No           255759 non-null  int64
 6   Ball No              255759 non-null  float64
 7   Bowler               255759 non-null  object
 8   Striker              255759 non-null  object
 9   Non Striker          255759 non-null  object
 10  runs_scored          255759 non-null  int64
 11  extras               255759 non-null  int64
 12  type of extras       13823 non-null   object
 13  score                255759 non-null  int64
 14  score/wicket         255759 non-null  object
 15  wicket_confirmation  255759 non-null  int64
 16  wicket_type          12651 non-null   object
 17  fielders_involved    9122 non-null    object
 18  Player Out           12651 non-null   object
dtypes: float64(1), int64(6), object(12)
memory usage: 37.1+ MB
```

[34]:
```
# ## 5. Drop Unnecessary Columns
df.drop([
    "Batting team", "Bowling team", "Ball No", "Non Striker", "extras", "score",
```

```
    "score/wicket", "type of extras", "wicket_type", "fielders_involved",␣
 ↪"Player Out"
], axis=1, inplace=True)
```

[35]:
```
# ## 6. Extract Year from Date Column
df["Year"] = pd.to_datetime(df["Date"], format="%d-%m-%Y").dt.year
```

[36]:
```
# ## 7. Aggregate Runs and Wickets by Year, Innings, Player, Match
a1 = df.groupby(['Year', 'Innings No', 'Striker', 'Bowler']) \
        .agg({"runs_scored": "sum", "wicket_confirmation": "sum"}) \
        .reset_index()
```

[37]:
```
runs = a1.groupby(['Year', 'Innings No', 'Striker']) \
         .agg({"runs_scored": "sum"}).reset_index()
wickets = a1.groupby(['Year', 'Innings No', 'Bowler']) \
            .agg({"wicket_confirmation": "sum"}).reset_index()
```

[38]:
```
# ## 8. Identify Top 3 Players Each Year
seasons = runs["Year"].unique()
print(seasons)
```

```
[2008 2009 2010 2011 2012 2013 2014 2015 2016 2017 2018 2019 2020 2021
 2022 2023 2024]
```

[39]:
```
for season in seasons:
    runs_season = runs[runs["Year"] == season]
    wickets_season = wickets[wickets["Year"] == season]

    print(f"Year: {season}\n")
    print("Top 3 Run Scorers:")
    print(runs_season.sort_values(by="runs_scored", ascending=False).
 ↪head(3)[['Striker', 'runs_scored']])

    print("\nTop 3 Wicket Takers:")
    print(wickets_season.sort_values(by="wicket_confirmation", ascending=False).
 ↪head(3)[['Bowler', 'wicket_confirmation']])
    print("\n" + "="*50 + "\n")
```

```
Year: 2008

Top 3 Run Scorers:
       Striker  runs_scored
31   G Gambhir          334
241  SR Watson          321
262  YK Pathan          318

Top 3 Wicket Takers:
           Bowler  wicket_confirmation
```

```
75  Sohail Tanvir                   15
82      VY Mahesh                   14
44    MF Maharoof                   14

================================================

Year: 2009

Top 3 Run Scorers:
            Striker  runs_scored
334       ML Hayden          363
376        SK Raina          352
278  AB de Villiers          343

Top 3 Wicket Takers:
        Bowler  wicket_confirmation
202  DP Nannes                   16
244   RP Singh                   15
181    A Nehra                   14

================================================

Year: 2010

Top 3 Run Scorers:
          Striker  runs_scored
644  SR Tendulkar          371
716     JH Kallis          368
646     SS Tiwary          342

Top 3 Wicket Takers:
              Bowler  wicket_confirmation
381         A Kumble                   16
411         DW Steyn                   16
518  Harbhajan Singh                   16

================================================

Year: 2011

Top 3 Run Scorers:
         Striker  runs_scored
882      M Vijay          381
1106     V Kohli          373
885   MEK Hussey          363

Top 3 Wicket Takers:
         Bowler  wicket_confirmation
```

```
782    R Ashwin                   18
809  SL Malinga                   15
693  SL Malinga                   15

================================================

Year: 2012

Top 3 Run Scorers:
        Striker  runs_scored
1227   S Dhawan          478
1286   CH Gayle          399
1129  AM Rahane          379

Top 3 Wicket Takers:
         Bowler  wicket_confirmation
889    M Morkel                   26
921  SL Malinga                   21
922   SP Narine                   17

================================================

Year: 2013

Top 3 Run Scorers:
       Striker  runs_scored
1436  CH Gayle          530
1549   V Kohli          439
1533  SK Raina          426

Top 3 Wicket Takers:
           Bowler  wicket_confirmation
1094  JP Faulkner                   22
1072     DJ Bravo                   20
1145    SP Narine                   18

================================================

Year: 2014

Top 3 Run Scorers:
         Striker  runs_scored
1912  RV Uthappa          457
1818    V Sehwag          364
1736   DA Warner          359

Top 3 Wicket Takers:
         Bowler  wicket_confirmation
```

```
1319  MM Sharma                   21
1368    B Kumar                   15
1397  KV Sharma                   13

================================================

Year: 2015

Top 3 Run Scorers:
          Striker  runs_scored
1965    DA Warner          411
1958  BB McCullum          372
1994  LMP Simmons          371

Top 3 Wicket Takers:
          Bowler  wicket_confirmation
1613  SL Malinga                   18
1558    DJ Bravo                   15
1545     A Nehra                   15

================================================

Year: 2016

Top 3 Run Scorers:
             Striker  runs_scored
2292         V Kohli          668
2177  AB de Villiers          490
2321       DA Warner          468

Top 3 Wicket Takers:
         Bowler  wicket_confirmation
1640    B Kumar                   16
1821  YS Chahal                   15
1649   DJ Bravo                   14

================================================

Year: 2017

Top 3 Run Scorers:
        Striker  runs_scored
2441  DA Warner          424
2575  G Gambhir          381
2507   S Dhawan          348

Top 3 Wicket Takers:
              Bowler  wicket_confirmation
```

```
1923         B Kumar                   18
1857       JJ Bumrah                   17
1867  MJ McClenaghan                   14

==================================================

Year: 2018

Top 3 Run Scorers:
            Striker  runs_scored
2745        RR Pant          491
2825       KL Rahul          482
2827  KS Williamson          426

Top 3 Wicket Takers:
         Bowler  wicket_confirmation
2016     AJ Tye                   19
2044  JC Archer                   13
2165     S Kaul                   13

==================================================

Year: 2019

Top 3 Run Scorers:
        Striker  runs_scored
2919  DA Warner          430
2976  Q de Kock          400
3065   KL Rahul          390

Top 3 Wicket Takers:
           Bowler  wicket_confirmation
2222     K Rabada                   18
2305    JJ Bumrah                   17
2213  Imran Tahir                   16

==================================================

Year: 2020

Top 3 Run Scorers:
           Striker  runs_scored
3225      S Dhawan          423
3235       SS Iyer          406
3274  F du Plessis          380

Top 3 Wicket Takers:
              Bowler  wicket_confirmation
```

```
2489        K Rabada                   23
2485       JJ Bumrah                   18
2418  Mohammed Shami                   17

================================================

Year: 2021

Top 3 Run Scorers:
           Striker  runs_scored
3603      S Dhawan          432
3412  F du Plessis          425
3474    RD Gaikwad          362

Top 3 Wicket Takers:
         Bowler  wicket_confirmation
2575   HV Patel                   19
2722  SN Thakur                   17
2667   HV Patel                   16

================================================

Year: 2022

Top 3 Run Scorers:
         Striker  runs_scored
3679  JC Buttler          671
3691    KL Rahul          425
3759   SV Samson          386

Top 3 Wicket Takers:
           Bowler  wicket_confirmation
2953    YS Chahal                   23
2835  T Natarajan                   19
2940     TA Boult                   16

================================================

Year: 2023

Top 3 Run Scorers:
           Striker  runs_scored
4069  Shubman Gill          503
3973  F du Plessis          484
3971     DP Conway          446

Top 3 Wicket Takers:
              Bowler  wicket_confirmation
```

```
3138       MM Sharma                    23
3043     Rashid Khan                    17
3024  Mohammed Shami                    17

================================================

Year: 2024

Top 3 Run Scorers:
         Striker  runs_scored
4326  RD Gaikwad          426
4344   SP Narine          304
4356     TM Head          293

Top 3 Wicket Takers:
              Bowler  wicket_confirmation
3212        HV Patel                    14
3200  Arshdeep Singh                    12
3239       MM Sharma                    10

================================================
```

[40]:
```
# ## 9. Prepare Data for Distribution Fitting
runs_per_year = df.groupby(['Year', 'Striker'])[['runs_scored']].sum().
 ↪reset_index()
wickets_per_year = df.groupby(['Year', 'Bowler'])[['wicket_confirmation']].
 ↪sum().reset_index()
runs_per_year.sort_values(['Year', 'runs_scored'], ascending=False,␣
 ↪inplace=True)
wickets_per_year.sort_values(['Year', 'wicket_confirmation'], ascending=False,␣
 ↪inplace=True)
```

[41]:
```
last_3_seasons = runs_per_year['Year'].unique().tolist()[:3]
```

[42]:
```
# ## 10. Dictionary for Top 3 Players (last 3 seasons)
top_3_dict = {}
for season in last_3_seasons:
    top_3_dict[season] = {
        "batsmen": {k: [] for k in runs_per_year[runs_per_year["Year"] ==␣
 ↪season]["Striker"].to_list()[:3]},
        "bowlers": {k: [] for k in wickets_per_year[wickets_per_year["Year"] ==␣
 ↪season]["Bowler"].to_list()[:3]}
    }
```

[43]:
```
# ## 11. List of Candidate Distributions for Fitting
distribs = [
```

```
    "alpha", "beta", "betaprime", "burr12", "crystalball", "dgamma",␣
 ↪"dweibull", "erlang",
    "exponnorm", "f", "fatiguelife", "gamma", "gengamma", "gumbel_l",␣
 ↪"johnsonsb",
    "kappa4", "lognorm", "nct", "norm", "norminvgauss", "powernorm", "rice",
    "recipinvgauss", "t", "trapezoid", "truncnorm"
]
```

[44]:
```
# ## 12. Function to Get Best Fit Distribution
def get_best_distrib(data):
    results = []
    params = {}
    for distrib in distribs:
        dist = getattr(st, distrib)
        param = dist.fit(data)
        params[distrib] = param
        _, p = st.kstest(data, distrib, args=param)
        results.append((distrib, p))

    best_distrib, best_p = max(results, key=lambda x: x[1])

    print(f"Best Fitting Distribution: {best_distrib}")
    print(f"Best P-Value: {best_p}")
    print(f"Params for Best Fit: {str(params[best_distrib])}")

    return [best_distrib, best_p, params[best_distrib]]
```

[45]:
```
# ## 13. Fit Distributions for Top 3 Players Each Season
runs = df.groupby(['Year', 'Striker','Match id'])[['runs_scored']].sum().
 ↪reset_index()
wickets = df.groupby(['Year', 'Bowler', 'Match id'])[['wicket_confirmation']].
 ↪sum().reset_index()

for year in top_3_dict.keys():
    for striker in top_3_dict[year]["batsmen"].keys():
        print("************************\n")
        print(f"Year: {year} | Batsman: {striker}\n")
        top_3_dict[year]["batsmen"][striker] =␣
 ↪get_best_distrib(runs[runs['Striker'] == striker]['runs_scored'])
        print('\n\n')

    for bowler in top_3_dict[year]["bowlers"].keys():
        print("************************\n")
        print(f"Year: {year} | Bowler: {striker}\n")
        top_3_dict[year]["bowlers"][bowler] =␣
 ↪get_best_distrib(wickets[wickets['Bowler'] == bowler]['wicket_confirmation'])
        print('\n\n')
```

****

Year: 2024 | Batsman: RD Gaikwad

Best Fitting Distribution: nct
Best P-Value: 0.5881570496217834
Params for Best Fit: (np.float64(5.718048022849898),
np.float64(9.399490726283615), np.float64(-54.25277343780452),
np.float64(8.497060689079994))

****

Year: 2024 | Batsman: V Kohli

Best Fitting Distribution: beta
Best P-Value: 0.7807091136830002
Params for Best Fit: (np.float64(0.8162762768683594),
np.float64(2.339175317753771), np.float64(-1.208864897132228e-31),
np.float64(130.79369265890057))

****

Year: 2024 | Batsman: B Sai Sudharsan

Best Fitting Distribution: f
Best P-Value: 0.9743698730235856
Params for Best Fit: (np.float64(7.230079510849502),
np.float64(94.80990591947705), np.float64(-0.4687012931969091),
np.float64(39.842021248481544))

****

Year: 2024 | Bowler: B Sai Sudharsan

Best Fitting Distribution: alpha
Best P-Value: 0.0002993252328930708
Params for Best Fit: (np.float64(5.200800514990576),
np.float64(-4.106246473111661), np.float64(27.580368990504883))

****

Year: 2024 | Bowler: B Sai Sudharsan

Best Fitting Distribution: alpha
Best P-Value: 0.6028771589628603
Params for Best Fit: (np.float64(6.113363581345144),
np.float64(-5.245777123804531), np.float64(39.57745263632695))

****

Year: 2024 | Bowler: B Sai Sudharsan

Best Fitting Distribution: t
Best P-Value: 0.004473243416688644
Params for Best Fit: (np.float64(4.822497644715119),
np.float64(1.1162819391895469), np.float64(0.9153269129308039))

****

Year: 2023 | Batsman: Shubman Gill

Best Fitting Distribution: johnsonsb
Best P-Value: 0.6214006077205521
Params for Best Fit: (np.float64(1.127462972555547),
np.float64(0.7082040622620326), np.float64(-1.0785135120264155),
np.float64(140.579464379851))

****

Year: 2023 | Batsman: F du Plessis

Best Fitting Distribution: beta
Best P-Value: 0.591353132429538
Params for Best Fit: (np.float64(0.9649241424447583),
np.float64(2.365566455697893), np.float64(-8.907903449602261e-30),
np.float64(110.45361094162868))

****

Year: 2023 | Batsman: DP Conway

```
Best Fitting Distribution: beta
Best P-Value: 0.9335739280635688
Params for Best Fit: (np.float64(0.6250316512826838),
np.float64(0.6786342050356671), np.float64(-3.4741633120498916),
np.float64(95.47416331204991))



**********

Year: 2023 | Bowler: DP Conway

Best Fitting Distribution: t
Best P-Value: 0.00012008020713636172
Params for Best Fit: (np.float64(29.05846643939152),
np.float64(1.2878076424619436), np.float64(1.197404368883093))



**********

Year: 2023 | Bowler: DP Conway

Best Fitting Distribution: alpha
Best P-Value: 0.0005609846480252997
Params for Best Fit: (np.float64(6.734843933630203),
np.float64(-5.500744811228249), np.float64(44.826257131250145))



**********

Year: 2023 | Bowler: DP Conway

Best Fitting Distribution: alpha
Best P-Value: 1.4259399000489277e-06
Params for Best Fit: (np.float64(5.783058438949956),
np.float64(-4.20986029264825), np.float64(30.878991656277478))



**********

Year: 2022 | Batsman: JC Buttler

Best Fitting Distribution: exponnorm
Best P-Value: 0.7137955109895673
```

```
Params for Best Fit: (np.float64(3054.885295608514),
np.float64(-0.031805252610631926), np.float64(0.01119090499814962))



**********

Year: 2022 | Batsman: KL Rahul

Best Fitting Distribution: johnsonsb
Best P-Value: 0.9402453631468675
Params for Best Fit: (np.float64(0.9331207997896902),
np.float64(0.7776389044559282), np.float64(-2.345202857963015),
np.float64(143.08331948370605))



**********

Year: 2022 | Batsman: Q de Kock

Best Fitting Distribution: burr12
Best P-Value: 0.4931279667432127
Params for Best Fit: (np.float64(590926023.7998527),
np.float64(0.05483081555360233), np.float64(-969803927.022117),
np.float64(969803927.160071))



**********

Year: 2022 | Bowler: Q de Kock

Best Fitting Distribution: alpha
Best P-Value: 1.1180274965710717e-05
Params for Best Fit: (np.float64(6.054854001673274),
np.float64(-4.898293043793716), np.float64(36.81747298117385))



**********

Year: 2022 | Bowler: Q de Kock

Best Fitting Distribution: exponnorm
Best P-Value: 0.3076424973571079
Params for Best Fit: (np.float64(1.5651879172672551),
np.float64(0.40254290759385924), np.float64(0.6274498232929551))
```

```
**********

Year: 2022 | Bowler: Q de Kock

Best Fitting Distribution: alpha
Best P-Value: 0.017666063432803525
Params for Best Fit: (np.float64(8.172744476082507),
np.float64(-7.746415964015842), np.float64(75.18055369544504))
```

[46]:
```
# ## 14. Assigned Player Distribution Fit (N Pooran)
my_player = "N Pooran"
my_data = runs[runs['Striker'] == my_player]['runs_scored']
get_best_distrib(my_data)
```

```
Best Fitting Distribution: dgamma
Best P-Value: 0.17382307012588938
Params for Best Fit: (np.float64(1.941154704943994),
np.float64(22.522643564002294), np.float64(8.834432758239476))
```

[46]:
```
['dgamma',
 np.float64(0.17382307012588938),
 (np.float64(1.941154704943994),
  np.float64(22.522643564002294),
  np.float64(8.834432758239476))]
```

[47]:
```
# ## 15. Total Stats by Year
total_runs_per_year = df.groupby(['Year', 'Striker'])[['runs_scored']].sum().
 ↪reset_index().sort_values(['Year', 'runs_scored'], ascending=False)
total_wickets_per_year = df.groupby(['Year',␣
 ↪'Bowler'])[['wicket_confirmation']].sum().reset_index().sort_values(['Year',␣
 ↪'wicket_confirmation'], ascending=False)
```

[48]:
```
# ## 16. Load Salary Dataset
salary_df = pd.read_excel(SALARY_DATASET_PATH)
```

[49]:
```
# ## 17. Filter 2024 Data
total_runs_2024 = total_runs_per_year[total_runs_per_year['Year'] == 2024]
total_wickets_2024 = total_wickets_per_year[total_wickets_per_year['Year'] ==␣
 ↪2024]
```

[50]:
```
# ## 18. Fuzzy Matching for Player Names
def match_names(name, names_list, threshold=80):
```

```
    if not name or not isinstance(name, str):
        return None

    result = process.extractOne(name, names_list)
    if result:
        match, score = result # type: ignore
        return match if score >= threshold else None
    return None
```

[51]:
```
# ## 19. Correlation for Batsmen
df_striker_salary = salary_df.copy()
df_striker_salary['Matched Player'] = df_striker_salary['Player'].apply(lambda␣
 ↪x: match_names(x, total_runs_2024['Striker'].tolist()))
df_striker_merged = pd.merge(df_striker_salary, total_runs_2024,␣
 ↪left_on='Matched Player', right_on='Striker')

corr_striker = df_striker_merged['Rs'].corr(df_striker_merged['runs_scored'])
print('Correlation between Salary and Runs in 2024:', corr_striker)
```

```
Correlation between Salary and Runs in 2024: 0.3061248376582167
```

[52]:
```
# ## 20. Correlation for Bowlers
df_bowler_salary = salary_df.copy()
df_bowler_salary['Matched Player'] = df_bowler_salary['Player'].apply(lambda x:␣
 ↪match_names(x, total_wickets_2024['Bowler'].tolist()))
df_bowler_merged = pd.merge(df_bowler_salary, total_wickets_2024,␣
 ↪left_on='Matched Player', right_on='Bowler')

corr_bowler = df_bowler_merged['Rs'].
 ↪corr(df_bowler_merged['wicket_confirmation'])
print('Correlation between Salary and Wickets in 2024:', corr_bowler)
```

```
Correlation between Salary and Wickets in 2024: 0.056932579421469245
```