FIRST INNINGS SCORE PREDICTION

# OBJECTIVE :

### \*The Project I have done is First Innings Score Prediction using Machine Learning.

\*This will Predict the Score of the First Innings by taking the overs, after

### 5 in order get the Overview of the Match.

In [1]:

*# Importing essential libraries*

**import** pandas **as** pd

**import** pickle

*# Loading the dataset*

df **=** pd**.**read\_csv(r'C:\Users\Admin\Downloads\ipl11.csv')

In [2]:

df**.**head()

Out[2]:

**mid date venue bat\_team bowl\_team batsman bowler runs wickets overs runs\_last\_5 wickets\_la**

2008-

**0** 1 04-18

**1** 1 2008-

04-18

**2** 1 2008-

04-18

**3** 1 2008-

04-18

**4** 1 2008-

04-18

M

Chinnaswamy

Stadium

M

Chinnaswamy

Stadium

M

Chinnaswamy

Stadium

M

Chinnaswamy

Stadium

M

Chinnaswamy

Stadium

Kolkata Knight Riders

Kolkata Knight Riders

Kolkata Knight Riders

Kolkata Knight Riders

Kolkata Knight Riders

Royal Challengers Bangalore

Royal Challengers Bangalore

Royal Challengers Bangalore

Royal Challengers Bangalore

Royal Challengers Bangalore

SC

Ganguly

BB

McCullum

BB

McCullum

BB

McCullum

BB

McCullum

P

Kumar

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 1 | 0 | 0.1 | 1 |  |
| 1 | 0 | 0.2 | 1 |  |
| 2 | 0 | 0.2 | 2 |  |
| 2 | 0 | 0.3 | 2 |  |
| 2 | 0 | 0.4 | 2 |  |

P

Kumar

P

Kumar

P

Kumar

P

Kumar

In [3]:

*# --- Data Cleaning ---*

*# Removing unwanted columns*

columns\_to\_remove **=** ['mid', 'batsman', 'bowler', 'striker', 'non-striker'] df**.**drop(labels**=**columns\_to\_remove, axis**=**1, inplace**=True**)

In [4]:

df**.**head()

Out[4]:

**date venue bat\_team bowl\_team runs wickets overs runs\_last\_5 wickets\_last\_5 total**

2008-

**0** 04-18

M Chinnaswamy

Stadium

Kolkata Knight Riders

Royal Challengers Bangalore

1 0 0.1 1 0 222

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **date venue bat\_team** | **bowl\_team** | **runs** | **wickets** | **overs** | **runs\_last\_5** | **wickets\_last\_5** | **total** |
| 2008- M Chinnaswamy Kolkata  **1** 04-18 Stadium Knight | Royal Challengers | 1 | 0 | 0.2 | 1 | 0 | 222 |
| Riders | Bangalore |  |  |  |  |  |  |
| 2008- M Chinnaswamy Kolkata  **2** 04-18 Stadium Knight | Royal Challengers | 2 | 0 | 0.2 | 2 | 0 | 222 |
| Riders | Bangalore |  |  |  |  |  |  |
| 2008- M Chinnaswamy Kolkata  **3** 04-18 Stadium Knight | Royal Challengers | 2 | 0 | 0.3 | 2 | 0 | 222 |
| Riders | Bangalore |  |  |  |  |  |  |
| 2008- M Chinnaswamy Kolkata  **4** 04-18 Stadium Knight | Royal Challengers | 2 | 0 | 0.4 | 2 | 0 | 222 |
| Riders | Bangalore |  |  |  |  |  |  |

In [5]:

Out[5]:

In [6]:

array(['Kolkata Knight Riders', 'Chennai Super Kings', 'Rajasthan Royals', 'Mumbai Indians', 'Deccan Chargers', 'Kings XI Punjab',

df['bat\_team']**.**unique()

'Royal Challengers Bangalore', 'Delhi Daredevils',

'Kochi Tuskers Kerala', 'Pune Warriors', 'Sunrisers Hyderabad', 'Rising Pune Supergiants', 'Gujarat Lions',

'Rising Pune Supergiant'], dtype=object)

*# Keeping only consistent teams*

consistent\_teams **=** ['Kolkata Knight Riders', 'Chennai Super Kings', 'Rajasthan Royals', 'Mumbai Indians', 'Kings XI Punjab', 'Royal Challengers Bangalore', 'Delhi Daredevils', 'Sunrisers Hyderabad']

In [7]:

df **=** df[(df['bat\_team']**.**isin(consistent\_teams)) **&** (df['bowl\_team']**.**isin(consistent\_teams)

In [8]:

*# Removing the first 5 overs data in every match*

df **=** df[df['overs']**>=**5.0]

In [9]:

df**.**head()

Out[9]:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **date venue** | **bat\_team** | **bowl\_team** | **runs** | **wickets** | **overs** | **runs\_last\_5** | **wickets\_last\_5** | **total** |
| 2008- M  **32** 04-18 Chinnaswamy | Kolkata Knight | Royal Challengers | 61 | 0 | 5.1 | 59 | 0 | 222 |
| Stadium | Riders | Bangalore |  |  |  |  |  |  |
| 2008- M  **33** 04-18 Chinnaswamy | Kolkata Knight | Royal Challengers | 61 | 1 | 5.2 | 59 | 1 | 222 |
| Stadium | Riders | Bangalore |  |  |  |  |  |  |
| 2008- M  **34** 04-18 Chinnaswamy | Kolkata Knight | Royal Challengers | 61 | 1 | 5.3 | 59 | 1 | 222 |
| Stadium | Riders | Bangalore |  |  |  |  |  |  |
| 2008- M  **35** 04-18 Chinnaswamy | Kolkata Knight | Royal Challengers | 61 | 1 | 5.4 | 59 | 1 | 222 |
| Stadium | Riders | Bangalore |  |  |  |  |  |  |
| 2008- M  **36** 04-18 Chinnaswamy | Kolkata Knight | Royal Challengers | 61 | 1 | 5.5 | 58 | 1 | 222 |
| Stadium | Riders | Bangalore |  |  |  |  |  |  |

In [10]:

print(df['bat\_team']**.**unique()) print(df['bowl\_team']**.**unique())

['Kolkata Knight Riders' 'Chennai Super Kings' 'Rajasthan Royals' 'Mumbai Indians' 'Kings XI Punjab' 'Royal Challengers Bangalore' 'Delhi Daredevils' 'Sunrisers Hyderabad']

['Royal Challengers Bangalore' 'Kings XI Punjab' 'Delhi Daredevils' 'Rajasthan Royals' 'Mumbai Indians' 'Chennai Super Kings'

'Kolkata Knight Riders' 'Sunrisers Hyderabad']

In [11]:

*# Converting the column 'date' from string into datetime object*

**from** datetime **import** datetime

df['date'] **=** df['date']**.**apply(**lambda** x: datetime**.**strptime(x, '%Y-%m-%d'))

In [12]:

*# --- Data Preprocessing ---*

*# Converting categorical features using OneHotEncoding method*

encoded\_df **=** pd**.**get\_dummies(data**=**df, columns**=**['bat\_team', 'bowl\_team'])

In [13]:

encoded\_df

Out[13]:

**date venue runs wickets overs runs\_last\_5 wickets\_last\_5 total bat\_team\_Chennai**

**Super Kings**

**bat\_team\_Del**

**Daredev**

2008-

**32** 04-18

**33** 2008-

04-18

**34** 2008-

04-18

**35** 2008-

04-18

**36** 2008-

04-18

M

Chinnaswamy

Stadium

M

Chinnaswamy

Stadium

M

Chinnaswamy

Stadium

M

Chinnaswamy

Stadium

M

Chinnaswamy

Stadium

61 0 5.1 59 0 222 0

61 1 5.2 59 1 222 0

61 1 5.3 59 1 222 0

61 1 5.4 59 1 222 0

61 1 5.5 58 1 222 0

**...** ... ... ... ... ... ... ... ... ...

**75884**

**75885**

**75886**

**75887**

2017-

05-19

2017-

05-19

2017-

05-19

2017-

05-19

M

Chinnaswamy

Stadium

M

Chinnaswamy

Stadium

M

Chinnaswamy

Stadium

M

Chinnaswamy

Stadium

106 9 18.1 29 4 107 0

107 9 18.2 29 4 107 0

107 9 18.3 28 4 107 0

107 9 18.4 24 4 107 0

**date venue runs wickets overs runs\_last\_5 wickets\_last\_5 total bat\_team\_Chennai**

**Super Kings**

**bat\_team\_Del**

**Daredev**

**75888**

2017-

05-19

M

Chinnaswamy

Stadium

107 10 18.5 23 5 107 0

### 40108 rows × 24 columns

In [14]:

Out[14]:

In [15]:

encoded\_df**.**columns

Index(['date', 'venue', 'runs', 'wickets', 'overs', 'runs\_last\_5', 'wickets\_last\_5', 'total', 'bat\_team\_Chennai Super Kings', 'bat\_team\_Delhi Daredevils', 'bat\_team\_Kings XI Punjab', 'bat\_team\_Kolkata Knight Riders', 'bat\_team\_Mumbai Indians', 'bat\_team\_Rajasthan Royals', 'bat\_team\_Royal Challengers Bangalore', 'bat\_team\_Sunrisers Hyderabad', 'bowl\_team\_Chennai Super Kings', 'bowl\_team\_Delhi Daredevils', 'bowl\_team\_Kings XI Punjab', 'bowl\_team\_Kolkata Knight Riders', 'bowl\_team\_Mumbai Indians', 'bowl\_team\_Rajasthan Royals', 'bowl\_team\_Royal Challengers Bangalore', 'bowl\_team\_Sunrisers Hyderabad'],

dtype='object')

*# Rearranging the columns*

encoded\_df **=** encoded\_df[['date', 'bat\_team\_Chennai Super Kings', 'bat\_team\_Delhi Daredevi 'bat\_team\_Kolkata Knight Riders', 'bat\_team\_Mumbai Indians', 'bat\_team\_Raja 'bat\_team\_Royal Challengers Bangalore', 'bat\_team\_Sunrisers Hyderabad', 'bowl\_team\_Chennai Super Kings', 'bowl\_team\_Delhi Daredevils', 'bowl\_team\_K 'bowl\_team\_Kolkata Knight Riders', 'bowl\_team\_Mumbai Indians', 'bowl\_team\_R 'bowl\_team\_Royal Challengers Bangalore', 'bowl\_team\_Sunrisers Hyderabad', 'overs', 'runs', 'wickets', 'runs\_last\_5', 'wickets\_last\_5', 'total']]

In [16]:

*# Splitting the data into train and test set*

X\_train **=** encoded\_df**.**drop(labels**=**'total', axis**=**1)[encoded\_df['date']**.**dt**.**year **<=** 2016] X\_test **=** encoded\_df**.**drop(labels**=**'total', axis**=**1)[encoded\_df['date']**.**dt**.**year **>=** 2017]

In [17]:

y\_train **=** encoded\_df[encoded\_df['date']**.**dt**.**year **<=** 2016]['total']**.**values y\_test **=** encoded\_df[encoded\_df['date']**.**dt**.**year **>=** 2017]['total']**.**values

In [18]:

Out[18]:

In [19]:

Out[19]:

In [20]:

Out[20]:

In [21]:

array([222, 222, 222, ..., 107, 107, 107], dtype=int64)

encoded\_df['total']**.**values

y\_train

array([222, 222, 222, ..., 208, 208, 208], dtype=int64)

y\_test

array([207, 207, 207, ..., 107, 107, 107], dtype=int64)

*# Removing the 'date' column*

X\_train**.**drop(labels**=**'date', axis**=True**, inplace**=True**)

X\_test**.**drop(labels**=**'date', axis**=True**, inplace**=True**)

In [22]:

*# --- Model Building --- # Linear Regression Model*

**from** sklearn.linear\_model **import** LinearRegression regressor **=** LinearRegression() regressor**.**fit(X\_train,y\_train)

Out[22]:

In [23]:

Out[23]:

In [24]:

LinearRegression()

regressor**.**predict(X\_test)

array([172.07093429, 175.2197967 , 174.61607874, ..., 100.37504751,

99.80473879, 93.14382211])

*# Creating a pickle file for the classifier* filename **=** 'first-innings-score-lr-model.pkl' pickle**.**dump(regressor, open(filename, 'wb'))

In [25]:

*## Ridge Regression*

**from** sklearn.linear\_model **import** Ridge

**from** sklearn.model\_selection **import** RandomizedSearchCV

Ridge Regression

In [26]:

ridge**=**Ridge()

parameters**=**{'alpha':[1e-13,1e-11,1e-9,1e-5,1e-2,1,7,17,24,31,37,40]}

ridge\_regressor**=**RandomizedSearchCV(ridge,parameters,scoring**=**'neg\_mean\_squared\_error',cv**=**1 ridge\_regressor**.**fit(X\_train,y\_train)

C:\Users\Admin\anaconda3\lib\site-packages\sklearn\linear\_model\\_ridge.py:147: LinAlgWarni ng: Ill-conditioned matrix (rcond=1.62441e-19): result may not be accurate.

return linalg.solve(A, Xy, sym\_pos=True,

C:\Users\Admin\anaconda3\lib\site-packages\sklearn\linear\_model\\_ridge.py:147: LinAlgWarni ng: Ill-conditioned matrix (rcond=9.18025e-20): result may not be accurate.

return linalg.solve(A, Xy, sym\_pos=True,

C:\Users\Admin\anaconda3\lib\site-packages\sklearn\linear\_model\\_ridge.py:147: LinAlgWarni ng: Ill-conditioned matrix (rcond=9.91852e-21): result may not be accurate.

return linalg.solve(A, Xy, sym\_pos=True,

C:\Users\Admin\anaconda3\lib\site-packages\sklearn\linear\_model\\_ridge.py:147: LinAlgWarni ng: Ill-conditioned matrix (rcond=8.61947e-20): result may not be accurate.

return linalg.solve(A, Xy, sym\_pos=True,

C:\Users\Admin\anaconda3\lib\site-packages\sklearn\linear\_model\\_ridge.py:147: LinAlgWarni ng: Ill-conditioned matrix (rcond=1.32973e-19): result may not be accurate.

return linalg.solve(A, Xy, sym\_pos=True,

C:\Users\Admin\anaconda3\lib\site-packages\sklearn\linear\_model\\_ridge.py:147: LinAlgWarni ng: Ill-conditioned matrix (rcond=9.38175e-20): result may not be accurate.

return linalg.solve(A, Xy, sym\_pos=True,

C:\Users\Admin\anaconda3\lib\site-packages\sklearn\linear\_model\\_ridge.py:147: LinAlgWarni ng: Ill-conditioned matrix (rcond=1.16909e-19): result may not be accurate.

return linalg.solve(A, Xy, sym\_pos=True,

C:\Users\Admin\anaconda3\lib\site-packages\sklearn\linear\_model\\_ridge.py:147: LinAlgWarni ng: Ill-conditioned matrix (rcond=1.34244e-19): result may not be accurate.

return linalg.solve(A, Xy, sym\_pos=True,

C:\Users\Admin\anaconda3\lib\site-packages\sklearn\linear\_model\\_ridge.py:147: LinAlgWarni ng: Ill-conditioned matrix (rcond=1.02863e-19): result may not be accurate.

return linalg.solve(A, Xy, sym\_pos=True,

C:\Users\Admin\anaconda3\lib\site-packages\sklearn\linear\_model\\_ridge.py:147: LinAlgWarni

Out[26]:

In [27]:

ng: Ill-conditioned matrix (rcond=7.46664e-20): result may not be accurate. return linalg.solve(A, Xy, sym\_pos=True,

C:\Users\Admin\anaconda3\lib\site-packages\sklearn\linear\_model\\_ridge.py:147: LinAlgWarni ng: Ill-conditioned matrix (rcond=2.60168e-20): result may not be accurate.

return linalg.solve(A, Xy, sym\_pos=True,

C:\Users\Admin\anaconda3\lib\site-packages\sklearn\linear\_model\\_ridge.py:147: LinAlgWarni ng: Ill-conditioned matrix (rcond=2.23702e-21): result may not be accurate.

return linalg.solve(A, Xy, sym\_pos=True, RandomizedSearchCV(cv=10, estimator=Ridge(),

param\_distributions={'alpha': [1e-13, 1e-11, 1e-09, 1e-05,

0.01, 1, 7, 17, 24, 31, 37,

40]},

scoring='neg\_mean\_squared\_error')

print(ridge\_regressor**.**best\_params\_) print(ridge\_regressor**.**best\_score\_)

{'alpha': 40}

-326.17628966626864

In [28]:

prediction**=**ridge\_regressor**.**predict(X\_test)

In [29]:

**import** seaborn **as** sns sns**.**distplot(y\_test**-**prediction)

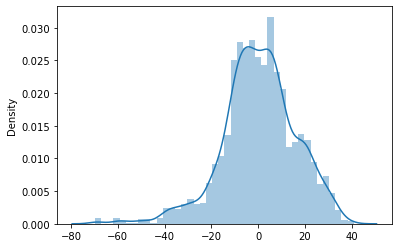
Out[29]:

In [30]:

C:\Users\Admin\anaconda3\lib\site-packages\seaborn\distributions.py:2619: FutureWarning: ` distplot` is a deprecated function and will be removed in a future version. Please adapt y our code to use either `displot` (a figure-level function with similar flexibility) or `hi stplot` (an axes-level function for histograms).

warnings.warn(msg, FutureWarning)

<AxesSubplot:ylabel='Density'>



MAE: 12.117294527005019

**from** sklearn **import** metrics

**import** numpy **as** np

print('MAE:', metrics**.**mean\_absolute\_error(y\_test, prediction)) print('MSE:', metrics**.**mean\_squared\_error(y\_test, prediction)) print('RMSE:', np**.**sqrt(metrics**.**mean\_squared\_error(y\_test, prediction)))

MSE: 251.03172964112724

RMSE: 15.843980864704655

## Lasso Regression

In [31]:

**from** sklearn.linear\_model **import** Lasso

**from** sklearn.model\_selection **import** RandomizedSearchCV

In [32]:

lasso**=**Lasso()

parameters**=**{'alpha':[1e-13,1e-11,1e-9,1e-5,1e-2,1,7,17,24,31,37,40]}

lasso\_regressor**=**RandomizedSearchCV(lasso,parameters,scoring**=**'neg\_mean\_squared\_error',cv**=**1

lasso\_regressor**.**fit(X\_train,y\_train) print(lasso\_regressor**.**best\_params\_) print(lasso\_regressor**.**best\_score\_)

C:\Users\Admin\anaconda3\lib\site-packages\sklearn\linear\_model\\_coordinate\_descent.py:53 0: ConvergenceWarning: Objective did not converge. You might want to increase the number o f iterations. Duality gap: 545236.0799651463, tolerance: 2832.403069708603

model = cd\_fast.enet\_coordinate\_descent(

C:\Users\Admin\anaconda3\lib\site-packages\sklearn\linear\_model\\_coordinate\_descent.py:53 0: ConvergenceWarning: Objective did not converge. You might want to increase the number o f iterations. Duality gap: 577545.7196925934, tolerance: 2976.5025190463425

model = cd\_fast.enet\_coordinate\_descent(

C:\Users\Admin\anaconda3\lib\site-packages\sklearn\linear\_model\\_coordinate\_descent.py:53 0: ConvergenceWarning: Objective did not converge. You might want to increase the number o f iterations. Duality gap: 773440.5980806481, tolerance: 2947.0448237759324

model = cd\_fast.enet\_coordinate\_descent(

C:\Users\Admin\anaconda3\lib\site-packages\sklearn\linear\_model\\_coordinate\_descent.py:53 0: ConvergenceWarning: Objective did not converge. You might want to increase the number o f iterations. Duality gap: 520412.4500997169, tolerance: 2880.11191421258

model = cd\_fast.enet\_coordinate\_descent(

C:\Users\Admin\anaconda3\lib\site-packages\sklearn\linear\_model\\_coordinate\_descent.py:53 0: ConvergenceWarning: Objective did not converge. You might want to increase the number o f iterations. Duality gap: 738261.2767869541, tolerance: 2938.929818864777

model = cd\_fast.enet\_coordinate\_descent(

C:\Users\Admin\anaconda3\lib\site-packages\sklearn\linear\_model\\_coordinate\_descent.py:53 0: ConvergenceWarning: Objective did not converge. You might want to increase the number o f iterations. Duality gap: 507345.1055719815, tolerance: 3014.2874176265714

model = cd\_fast.enet\_coordinate\_descent(

C:\Users\Admin\anaconda3\lib\site-packages\sklearn\linear\_model\\_coordinate\_descent.py:53 0: ConvergenceWarning: Objective did not converge. You might want to increase the number o f iterations. Duality gap: 371684.0302372668, tolerance: 2977.5869558591507

model = cd\_fast.enet\_coordinate\_descent(

C:\Users\Admin\anaconda3\lib\site-packages\sklearn\linear\_model\\_coordinate\_descent.py:53 0: ConvergenceWarning: Objective did not converge. You might want to increase the number o f iterations. Duality gap: 726987.6906137262, tolerance: 2951.0758268416835

model = cd\_fast.enet\_coordinate\_descent(

C:\Users\Admin\anaconda3\lib\site-packages\sklearn\linear\_model\\_coordinate\_descent.py:53 0: ConvergenceWarning: Objective did not converge. You might want to increase the number o f iterations. Duality gap: 2700897.8211471234, tolerance: 2832.403069708603

model = cd\_fast.enet\_coordinate\_descent(

C:\Users\Admin\anaconda3\lib\site-packages\sklearn\linear\_model\\_coordinate\_descent.py:53 0: ConvergenceWarning: Objective did not converge. You might want to increase the number o f iterations. Duality gap: 943781.3289563945, tolerance: 2976.5025190463425

model = cd\_fast.enet\_coordinate\_descent(

C:\Users\Admin\anaconda3\lib\site-packages\sklearn\linear\_model\\_coordinate\_descent.py:53 0: ConvergenceWarning: Objective did not converge. You might want to increase the number o f iterations. Duality gap: 3132297.812621603, tolerance: 2947.0448237759324

model = cd\_fast.enet\_coordinate\_descent(

C:\Users\Admin\anaconda3\lib\site-packages\sklearn\linear\_model\\_coordinate\_descent.py:53 0: ConvergenceWarning: Objective did not converge. You might want to increase the number o f iterations. Duality gap: 525628.877674765, tolerance: 2880.11191421258

model = cd\_fast.enet\_coordinate\_descent(

C:\Users\Admin\anaconda3\lib\site-packages\sklearn\linear\_model\\_coordinate\_descent.py:53 0: ConvergenceWarning: Objective did not converge. You might want to increase the number o f iterations. Duality gap: 1851069.0615584152, tolerance: 2938.929818864777

model = cd\_fast.enet\_coordinate\_descent(

C:\Users\Admin\anaconda3\lib\site-packages\sklearn\linear\_model\\_coordinate\_descent.py:53

0: ConvergenceWarning: Objective did not converge. You might want to increase the number o f iterations. Duality gap: 503731.2836086452, tolerance: 3014.2874176265714

model = cd\_fast.enet\_coordinate\_descent(

C:\Users\Admin\anaconda3\lib\site-packages\sklearn\linear\_model\\_coordinate\_descent.py:53 0: ConvergenceWarning: Objective did not converge. You might want to increase the number o f iterations. Duality gap: 2332787.0161870336, tolerance: 2977.5869558591507

model = cd\_fast.enet\_coordinate\_descent(

C:\Users\Admin\anaconda3\lib\site-packages\sklearn\linear\_model\\_coordinate\_descent.py:53 0: ConvergenceWarning: Objective did not converge. You might want to increase the number o f iterations. Duality gap: 731365.5823141662, tolerance: 2951.0758268416835

model = cd\_fast.enet\_coordinate\_descent(

C:\Users\Admin\anaconda3\lib\site-packages\sklearn\linear\_model\\_coordinate\_descent.py:53 0: ConvergenceWarning: Objective did not converge. You might want to increase the number o f iterations. Duality gap: 545214.1557764141, tolerance: 2832.403069708603

model = cd\_fast.enet\_coordinate\_descent(

C:\Users\Admin\anaconda3\lib\site-packages\sklearn\linear\_model\\_coordinate\_descent.py:53 0: ConvergenceWarning: Objective did not converge. You might want to increase the number o f iterations. Duality gap: 577579.5106445365, tolerance: 2976.5025190463425

model = cd\_fast.enet\_coordinate\_descent(

C:\Users\Admin\anaconda3\lib\site-packages\sklearn\linear\_model\\_coordinate\_descent.py:53 0: ConvergenceWarning: Objective did not converge. You might want to increase the number o f iterations. Duality gap: 773440.741889664, tolerance: 2947.0448237759324

model = cd\_fast.enet\_coordinate\_descent(

C:\Users\Admin\anaconda3\lib\site-packages\sklearn\linear\_model\\_coordinate\_descent.py:53 0: ConvergenceWarning: Objective did not converge. You might want to increase the number o f iterations. Duality gap: 520403.0371814277, tolerance: 2880.11191421258

model = cd\_fast.enet\_coordinate\_descent(

C:\Users\Admin\anaconda3\lib\site-packages\sklearn\linear\_model\\_coordinate\_descent.py:53 0: ConvergenceWarning: Objective did not converge. You might want to increase the number o f iterations. Duality gap: 738305.9291754775, tolerance: 2938.929818864777

model = cd\_fast.enet\_coordinate\_descent(

C:\Users\Admin\anaconda3\lib\site-packages\sklearn\linear\_model\\_coordinate\_descent.py:53 0: ConvergenceWarning: Objective did not converge. You might want to increase the number o f iterations. Duality gap: 507337.54687731527, tolerance: 3014.2874176265714

model = cd\_fast.enet\_coordinate\_descent(

C:\Users\Admin\anaconda3\lib\site-packages\sklearn\linear\_model\\_coordinate\_descent.py:53 0: ConvergenceWarning: Objective did not converge. You might want to increase the number o f iterations. Duality gap: 371620.6075416049, tolerance: 2977.5869558591507

model = cd\_fast.enet\_coordinate\_descent(

C:\Users\Admin\anaconda3\lib\site-packages\sklearn\linear\_model\\_coordinate\_descent.py:53 0: ConvergenceWarning: Objective did not converge. You might want to increase the number o f iterations. Duality gap: 726984.5948459059, tolerance: 2951.0758268416835

model = cd\_fast.enet\_coordinate\_descent(

{'alpha': 1}

-320.9085017331346

In [33]:

prediction1**=**lasso\_regressor**.**predict(X\_test)

In [34]:

Out[34]:

In [35]:

array([170.85983934, 174.24824974, 173.46897392, ..., 105.86186741,

prediction1

104.87877323, 98.9709836 ])

**from** sklearn **import** metrics

**import** numpy **as** np

print('MAE:', metrics**.**mean\_absolute\_error(y\_test, prediction1)) print('MSE:', metrics**.**mean\_squared\_error(y\_test, prediction1)) print('RMSE:', np**.**sqrt(metrics**.**mean\_squared\_error(y\_test, prediction1)))

MAE: 12.214053814850246

MSE: 262.3797366400714

RMSE: 16.19813991296752

# XGB Boost

In [36]:

**from** xgboost **import** XGBRegressor

In [37]:

xgboost**=**XGBRegressor()

parameters**=**{'alpha':[1e-13,1e-11,1e-9,1e-5,1e-2,1,7,17,24,31,37,40]}

xgboost\_regressor**=**RandomizedSearchCV(ridge,parameters,scoring**=**'neg\_mean\_squared\_error',cv xgboost\_regressor**.**fit(X\_train,y\_train)

C:\Users\Admin\anaconda3\lib\site-packages\sklearn\linear\_model\\_ridge.py:147: LinAlgWarni ng: Ill-conditioned matrix (rcond=1.40604e-17): result may not be accurate.

return linalg.solve(A, Xy, sym\_pos=True,

C:\Users\Admin\anaconda3\lib\site-packages\sklearn\linear\_model\\_ridge.py:147: LinAlgWarni ng: Ill-conditioned matrix (rcond=1.36584e-17): result may not be accurate.

return linalg.solve(A, Xy, sym\_pos=True,

C:\Users\Admin\anaconda3\lib\site-packages\sklearn\linear\_model\\_ridge.py:147: LinAlgWarni ng: Ill-conditioned matrix (rcond=1.37946e-17): result may not be accurate.

return linalg.solve(A, Xy, sym\_pos=True,

C:\Users\Admin\anaconda3\lib\site-packages\sklearn\linear\_model\\_ridge.py:147: LinAlgWarni ng: Ill-conditioned matrix (rcond=1.40733e-17): result may not be accurate.

return linalg.solve(A, Xy, sym\_pos=True,

C:\Users\Admin\anaconda3\lib\site-packages\sklearn\linear\_model\\_ridge.py:147: LinAlgWarni ng: Ill-conditioned matrix (rcond=1.40094e-17): result may not be accurate.

return linalg.solve(A, Xy, sym\_pos=True,

C:\Users\Admin\anaconda3\lib\site-packages\sklearn\linear\_model\\_ridge.py:147: LinAlgWarni ng: Ill-conditioned matrix (rcond=1.37684e-17): result may not be accurate.

return linalg.solve(A, Xy, sym\_pos=True,

C:\Users\Admin\anaconda3\lib\site-packages\sklearn\linear\_model\\_ridge.py:147: LinAlgWarni ng: Ill-conditioned matrix (rcond=1.3841e-17): result may not be accurate.

return linalg.solve(A, Xy, sym\_pos=True,

C:\Users\Admin\anaconda3\lib\site-packages\sklearn\linear\_model\\_ridge.py:147: LinAlgWarni ng: Ill-conditioned matrix (rcond=1.4027e-17): result may not be accurate.

return linalg.solve(A, Xy, sym\_pos=True,

C:\Users\Admin\anaconda3\lib\site-packages\sklearn\linear\_model\\_ridge.py:147: LinAlgWarni ng: Ill-conditioned matrix (rcond=1.40055e-17): result may not be accurate.

return linalg.solve(A, Xy, sym\_pos=True,

C:\Users\Admin\anaconda3\lib\site-packages\sklearn\linear\_model\\_ridge.py:147: LinAlgWarni ng: Ill-conditioned matrix (rcond=1.4006e-17): result may not be accurate.

return linalg.solve(A, Xy, sym\_pos=True,

C:\Users\Admin\anaconda3\lib\site-packages\sklearn\linear\_model\\_ridge.py:147: LinAlgWarni ng: Ill-conditioned matrix (rcond=1.62441e-19): result may not be accurate.

return linalg.solve(A, Xy, sym\_pos=True,

C:\Users\Admin\anaconda3\lib\site-packages\sklearn\linear\_model\\_ridge.py:147: LinAlgWarni ng: Ill-conditioned matrix (rcond=9.18025e-20): result may not be accurate.

return linalg.solve(A, Xy, sym\_pos=True,

C:\Users\Admin\anaconda3\lib\site-packages\sklearn\linear\_model\\_ridge.py:147: LinAlgWarni ng: Ill-conditioned matrix (rcond=9.91852e-21): result may not be accurate.

return linalg.solve(A, Xy, sym\_pos=True,

C:\Users\Admin\anaconda3\lib\site-packages\sklearn\linear\_model\\_ridge.py:147: LinAlgWarni ng: Ill-conditioned matrix (rcond=8.61947e-20): result may not be accurate.

return linalg.solve(A, Xy, sym\_pos=True,

C:\Users\Admin\anaconda3\lib\site-packages\sklearn\linear\_model\\_ridge.py:147: LinAlgWarni ng: Ill-conditioned matrix (rcond=1.32973e-19): result may not be accurate.

return linalg.solve(A, Xy, sym\_pos=True,

C:\Users\Admin\anaconda3\lib\site-packages\sklearn\linear\_model\\_ridge.py:147: LinAlgWarni ng: Ill-conditioned matrix (rcond=9.38175e-20): result may not be accurate.

return linalg.solve(A, Xy, sym\_pos=True,

C:\Users\Admin\anaconda3\lib\site-packages\sklearn\linear\_model\\_ridge.py:147: LinAlgWarni ng: Ill-conditioned matrix (rcond=1.16909e-19): result may not be accurate.

return linalg.solve(A, Xy, sym\_pos=True,

C:\Users\Admin\anaconda3\lib\site-packages\sklearn\linear\_model\\_ridge.py:147: LinAlgWarni ng: Ill-conditioned matrix (rcond=1.34244e-19): result may not be accurate.

Out[37]:

In [38]:

return linalg.solve(A, Xy, sym\_pos=True,

C:\Users\Admin\anaconda3\lib\site-packages\sklearn\linear\_model\\_ridge.py:147: LinAlgWarni ng: Ill-conditioned matrix (rcond=1.02863e-19): result may not be accurate.

return linalg.solve(A, Xy, sym\_pos=True,

C:\Users\Admin\anaconda3\lib\site-packages\sklearn\linear\_model\\_ridge.py:147: LinAlgWarni ng: Ill-conditioned matrix (rcond=7.46664e-20): result may not be accurate.

return linalg.solve(A, Xy, sym\_pos=True, RandomizedSearchCV(cv=10, estimator=Ridge(),

param\_distributions={'alpha': [1e-13, 1e-11, 1e-09, 1e-05,

0.01, 1, 7, 17, 24, 31, 37,

40]},

scoring='neg\_mean\_squared\_error')

prediction2**=**xgboost\_regressor**.**predict(X\_test)

In [39]:

**from** sklearn **import** metrics

**import** numpy **as** np

print('MAE:', metrics**.**mean\_absolute\_error(y\_test, prediction2)) print('MSE:', metrics**.**mean\_squared\_error(y\_test, prediction2)) print('RMSE:', np**.**sqrt(metrics**.**mean\_squared\_error(y\_test, prediction2)))

|  |  |  |  |
| --- | --- | --- | --- |
|  | | | MAE: 12.117294527005019  MSE: 251.03172964112724  RMSE: 15.843980864704655 |
| In | [ | ]: |  |
|  |  |  |  |
| In | [ | ]: |  |
|  |  |  |  |
| In | [ | ]: |  |
|  |  |  |  |
| In | [ | ]: |  |
|  |  |  |  |
| In | [ | ]: |  |