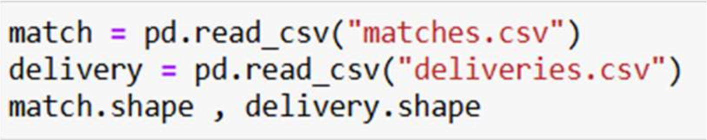
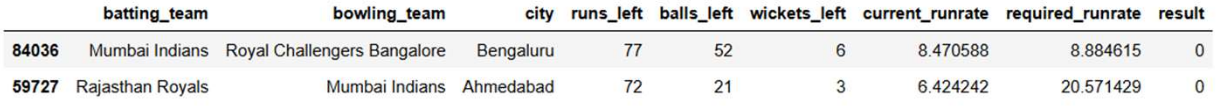
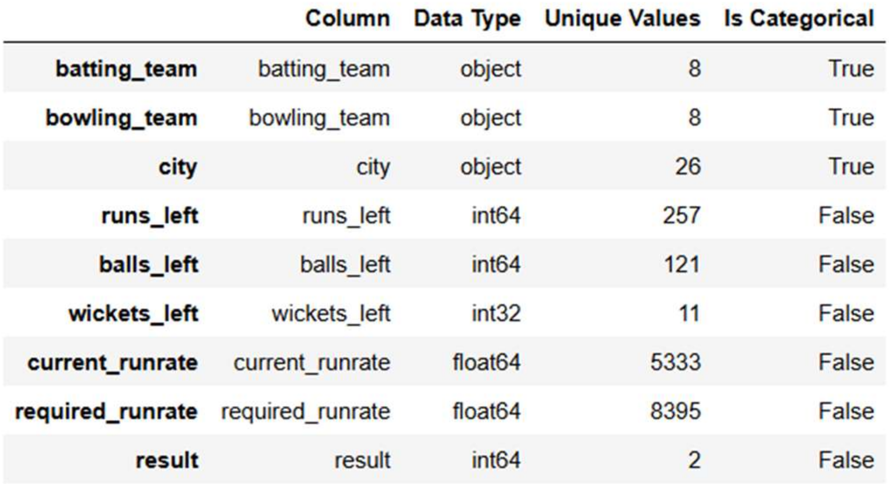


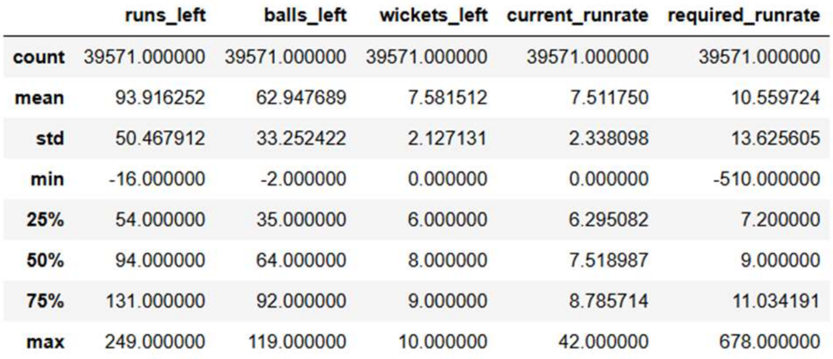
```
match = pd.read_csv("matches.csv")
delivery = pd.read_csv("deliveries.csv")
match.shape , delivery.shape
```



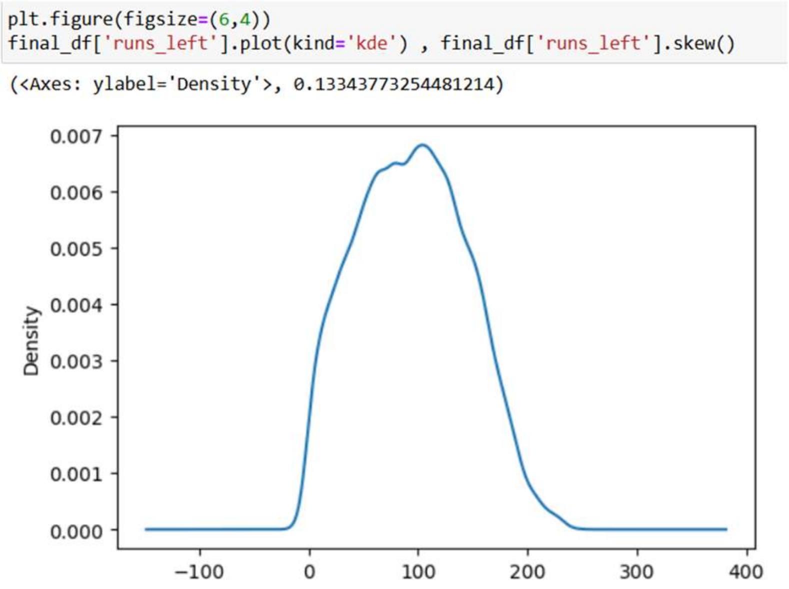
	batting_team	bowling_team	city	runs_left	balls_left	wickets_left	current_runrate	required_runrate	result
84036	Mumbai Indians	Royal Challengers Bangalore	Bengaluru	77	52	6	8.470588	8.884615	0
59727	Rajasthan Royals	Mumbai Indians	Ahmedabad	72	21	3	6.424242	20.571429	0



	Column	Data Type	Unique Values	Is Categorical
batting_team	batting_team	object	8	True
bowling_team	bowling_team	object	8	True
city	city	object	26	True
runs_left	runs_left	int64	257	False
balls_left	balls_left	int64	121	False
wickets_left	wickets_left	int32	11	False
current_runrate	current_runrate	float64	5333	False
required_runrate	required_runrate	float64	8395	False
result	result	int64	2	False

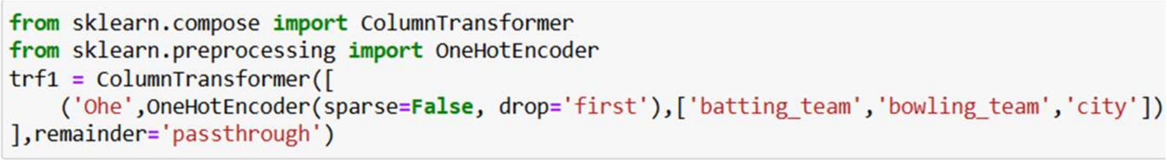


	runs_left	balls_left	wickets_left	current_runrate	required_runrate
count	39571.000000	39571.000000	39571.000000	39571.000000	39571.000000
mean	93.916252	62.947689	7.581512	7.511750	10.559724
std	50.467912	33.252422	2.127131	2.338098	13.625605
min	-16.000000	-2.000000	0.000000	0.000000	-510.000000
25%	54.000000	35.000000	6.000000	6.295082	7.200000
50%	94.000000	64.000000	8.000000	7.518987	9.000000
75%	131.000000	92.000000	9.000000	8.785714	11.034191
max	249.000000	119.000000	10.000000	42.000000	678.000000

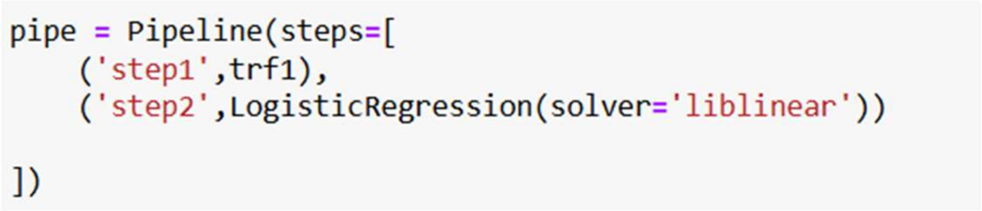


```
plt.figure(figsize=(6,4))
final_df['runs_left'].plot(kind='kde') , final_df['runs_left'].skew()
```

```
(<Axes: ylabel='Density'>, 0.13343773254481214)
```

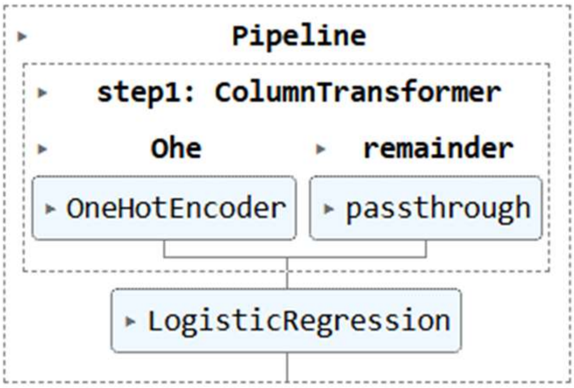


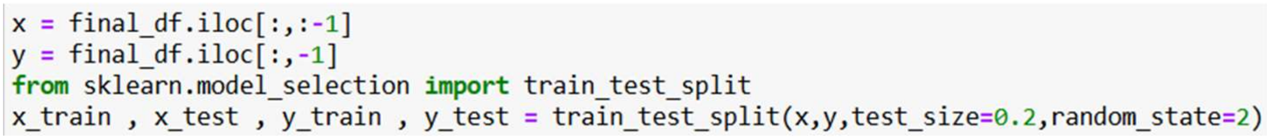
```
from sklearn.compose import ColumnTransformer
from sklearn.preprocessing import OneHotEncoder
trf1 = ColumnTransformer([
    ('Ohe',OneHotEncoder(sparse=False, drop='first'),['batting_team','bowling_team','city'])
],remainder='passthrough')
```



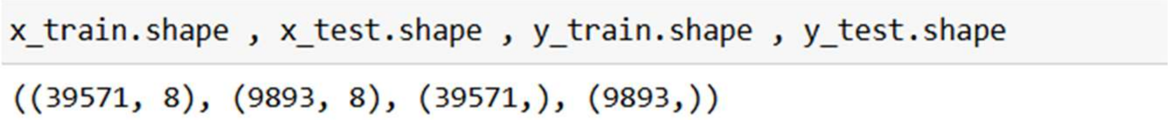
```
pipe = Pipeline(steps=[
    ('step1',trf1),
    ('step2',LogisticRegression(solver='liblinear'))

])
```



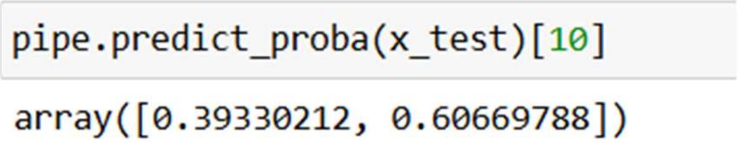


```
x = final_df.iloc[:,:-1]
y = final_df.iloc[:,-1]
from sklearn.model_selection import train_test_split
x_train , x_test , y_train , y_test = train_test_split(x,y,test_size=0.2,random_state=2)
```



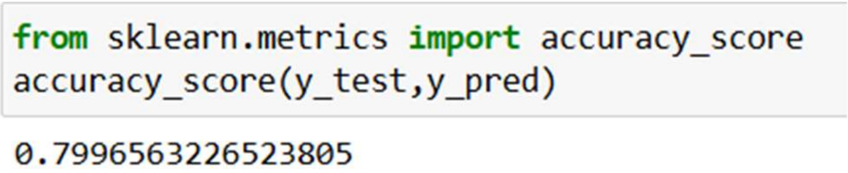
```
x_train.shape , x_test.shape , y_train.shape , y_test.shape
```

```
((39571, 8), (9893, 8), (39571,), (9893,))
```



```
pipe.predict_proba(x_test)[10]
```

```
array([0.39330212, 0.60669788])
```



```
from sklearn.metrics import accuracy_score
accuracy_score(y_test,y_pred)
```

```
0.7996563226523805
```