MACHINE LEARNING LABORATORY

1818128_JAYASREE T

☆ EXP 4 : Write a python program to implement, visualize and evaluate the performance metrics of simple linear regression using weather conditions in world war two datset

Importing the packages

```
[27] import pandas as pd
     import numpy as np
     import matplotlib.pyplot as plt
     import seaborn as seabornInstance
     from sklearn.model_selection import train_test_split
     from sklearn.linear_model import LinearRegression
     from sklearn import metrics
     %matplotlib inline
```

Preparing Data

```
[28] dataset = pd.read_csv('dataset.csv')
     dataset.describe()
```

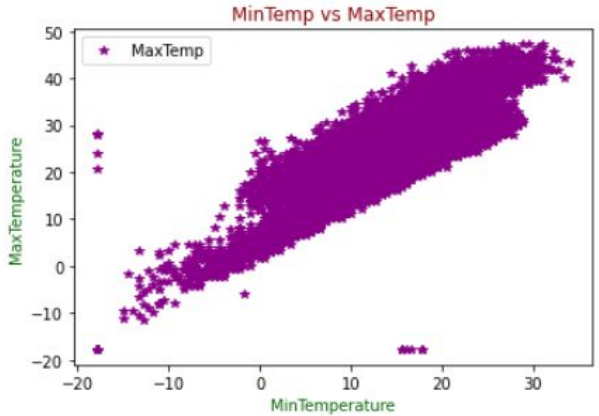
```
/usr/local/lib/python3.6/dist-packages/IPython/core/interactiveshell.py:2718: DtypeWarning: Columns (8,25) have mixed types.Specify dtype option c
  interactivity=interactivity, compiler=compiler, result=result)
```

	STA	WindGustSpd	MaxTemp	MinTemp	MeanTemp	Snowfall	YR	MO	DA	DR	SPD	
count	44109.000000	0.0	44109.000000	44109.000000	44109.000000	43635.000000	44108.000000	44108.000000	44108.000000	0.0	0.0	43722.0
mean	11624.114466	NaN	28.461450	19.860648	24.156685	0.015659	43.718804	6.729664	15.793530	NaN	NaN	83.9
std	1243.277768	NaN	7.148130	6.489605	6.597677	0.512345	1.136852	3.429048	8.800034	NaN	NaN	10.2
min	10001.000000	NaN	-17.777778	-17.777778	-17.777778	0.000000	40.000000	1.000000	1.000000	NaN	NaN	11.0
25%	10705.000000	NaN	27.777778	18.888889	24.444444	0.000000	43.000000	4.000000	8.000000	NaN	NaN	82.0
50%	11604.000000	NaN	30.000000	22.222222	26.111111	0.000000	44.000000	7.000000	16.000000	NaN	NaN	86.0
75%	12001.000000	NaN	31.666667	23.333333	27.222222	0.000000	45.000000	10.000000	23.000000	NaN	NaN	89.0
max	16202.000000	NaN	47.222222	33.888889	38.888889	33.020000	45.000000	12.000000	31.000000	NaN	NaN	117.0

[28]

max	16202.000000	NaN	47.222222	33.888889	38.888889	33.020000	45.000000	12.000000	31.000000	NaN	NaN	117.(

[77]
```
dataset.plot(x='MinTemp', y='MaxTemp', style='*',color='DarkMagenta')
plt.title('MinTemp vs MaxTemp',color='DarkRed')
plt.xlabel('MinTemperature',color='DarkGreen')
plt.ylabel('MaxTemperature',color='DarkGreen')
plt.show()
```

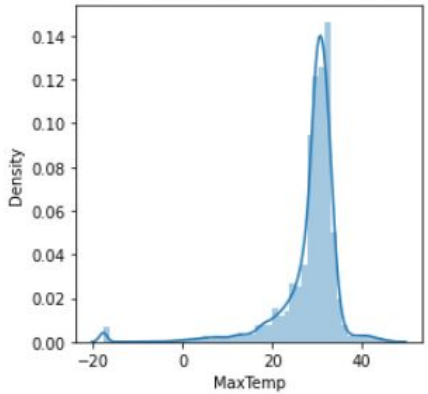


[66] pl.figure(figsize=(4,4))

MinTemperature

```
[66] pl.figure(figsize=(4,4))
     pl.tight_layout()
     seabornInstance.distplot(dataset['MaxTemp'])
     X = dataset['MinTemp'].values.reshape(-1,1)
     y = dataset['MaxTemp'].values.reshape(-1,1)
```

```
/usr/local/lib/python3.6/dist-packages/seaborn/distributions.py:2557: FutureWarning: `distplot` is a deprecated function and will be removed in a
  warnings.warn(msg, FutureWarning)
```



Splitting Dataset into training and testing sets

```
[31] X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=0)
     regressor = LinearRegression()
     regressor.fit(X_train, y_train)
```

```
LinearRegression(copy_X=True, fit_intercept=True, n_jobs=None, normalize=False)
```

```
[32] LinearRegression(copy_X=True, fit_intercept=True, n_jobs=None, normalize=False)
```

```
LinearRegression(copy_X=True, fit_intercept=True, n_jobs=None, normalize=False)
```

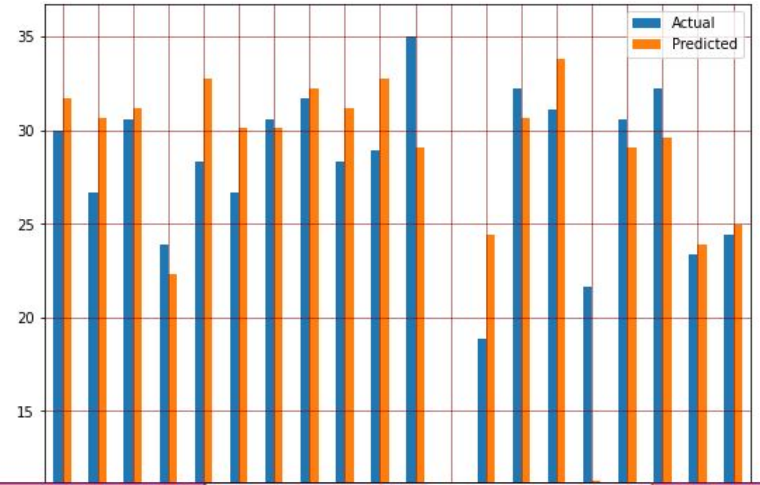
```
[33] print(regressor.intercept_)
     print(regressor.coef_)
```

```
[9.70563356]
[[0.94399232]]
```

Making predictions

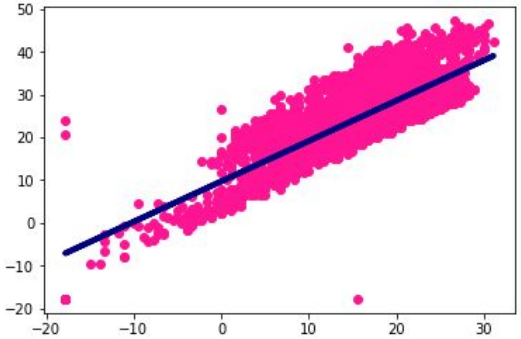
Making predictions

```
[73] y_pred = regressor.predict(X_test)
     df = pd.DataFrame({'Actual': y_test.flatten(), 'Predicted': y_pred.flatten()})
     df1 = df.head(20)
     df1.plot(kind='bar',figsize=(9,9))
     pl.grid(which='major', linestyle='-', linewidth='0.5', color='DarkRed')
     pl.grid(which='minor', linestyle=':', linewidth='0.5', color='blue')
     pl.show()
```



[73]

[75]
```
pl.scatter(X_test, y_test,  color='DeepPink')
pl.plot(X_test, y_pred, color='navy', linewidth=4)
pl.show()
```



[76]
```
print('Mean absolute error:', round(metrics.mean_absolute_error(y_test, y_pred),3))
print('Mean squared error:', round(metrics.mean_squared_error(y_test, y_pred),3))
print('Root mean squared error:', round(np.sqrt(metrics.mean_squared_error(y_test, y_pred)),4))
```

```
Mean absolute error: 2.835
Mean squared error: 13.664
Root mean squared error: 3.6964
```