8B_LR_SVM

November 7, 2020

[1]:

```
import numpy as np
import pandas as pd
import plotly
import plotly.figure_factory as ff
import plotly.graph_objs as go
from sklearn.svm import SVC
from sklearn.linear_model import LogisticRegression, SGDClassifier
from sklearn.preprocessing import StandardScaler
from sklearn.preprocessing import MinMaxScaler
from plotly.offline import download_plotlyjs, init_notebook_mode, plot, iplot
init_notebook_mode(connected=True)
```

[2]:

```
data = pd.read_csv('task_b.csv')
data=data.iloc[:,1:]
```

[3]:

```
data.head()
```

[3]:

```
            f1            f2        f3    y
0  -195.871045 -14843.084171  5.532140  1.0
1 -1217.183964  -4068.124621  4.416082  1.0
2     9.138451   4413.412028  0.425317  0.0
3   363.824242  15474.760647  1.094119  0.0
4  -768.812047  -7963.932192  1.870536  0.0
```

[4]:

```
data.var()
```

[4]:

```
f1    2.383344e+05
f2    1.082311e+08
f3    8.565349e+00
y     2.512563e-01
dtype: float64
```

[5]:

```
data.corr()['y']
```

[5]:

```
f1    0.067172
f2   -0.017944
f3    0.839060
y     1.000000
```

```
Name: y, dtype: float64
```

```
[6]: data.std()
```

```
[6]: f1      488.195035
     f2    10403.417325
     f3        2.926662
     y         0.501255
     dtype: float64
```

```
[7]: X=data[['f1','f2','f3']].values
     Y=data['y'].values
     print(X.shape)
     print(Y.shape)
```

```
(200, 3)
(200,)
```

1 What if our features are with different variance

```
[8]: clf1 = SGDClassifier(tol=1e-3, loss='log', random_state=2, early_stopping=True)
     clf1.fit(X, Y)
     print(clf1.coef_, "\n")
     print("As per clf coeff_ Feature {} is more important : ".format(np.argmax(clf1.
      ↪coef_)+1), np.max(clf1.coef_))
```

```
[[17775.79389064 -3030.39642824  6137.25731743]]

As per clf coeff_ Feature 1 is more important :  17775.793890644018
```

```
[9]: clf2 = SGDClassifier(tol=1e-3, random_state=2, early_stopping=True)
     clf2.fit(X, Y)
     print(clf2.coef_, "\n")
     print("As per clf coeff_ Feature {} is more important : ".format(np.argmax(clf2.
      ↪coef_)+1), np.max(clf2.coef_))
```

```
[[ 15212.9889028  -10153.69088156   6120.07374827]]

As per clf coeff_ Feature 1 is more important :  15212.988902799994
```

1.0.1 OBSERVATION:

```
1. Feature 1 is important than other 2 without standardizing the data
```

1.0.2 Task2:

[10]:

```
scaler = StandardScaler()
X = scaler.fit_transform(X)
```

[11]:

```
clf1 = SGDClassifier(max_iter=50, tol=1e-3, loss='log', random_state=2,␣
 ↪early_stopping=True)
clf1.fit(X, Y)
print(clf1.coef_, "\n")
print("As per clf coeff_ Feature {} is more important : ".format(np.argmax(clf1.
 ↪coef_)+1), np.max(clf1.coef_))
```

```
[[ 3.99830555  3.73532394 29.76712453]]

As per clf coeff_ Feature 3 is more important :  29.767124527248654
```

[12]:

```
clf2 = SGDClassifier(tol=1e-3, random_state=2, early_stopping=True)
clf2.fit(X, Y)
print(clf2.coef_, "\n")
print("As per clf coeff_ Feature {} is more important : ".format(np.argmax(clf2.
 ↪coef_)+1), np.max(clf2.coef_))
```

```
[[ 6.79991736  2.1136678  33.56134459]]

As per clf coeff_ Feature 3 is more important :  33.561344587338574
```

1.0.3 OBSERVATION :

1. Feature 3 is important than other 2 with standardizing the data

1.0.4 OVERALL OBSERVATION :

1. SVM and LogisticRegression co-eff also nearly same.

2. Standardization impacts much on feature selection.

3. Higher variance features may be a least important feature.