8B_Solution

April 12, 2022

[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.linear_model import LogisticRegression
from sklearn.linear_model import SGDClassifier
from sklearn.svm import LinearSVC
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.corr()['y']
```

[4]:
```
f1    0.067172
f2   -0.017944
f3    0.839060
y     1.000000
Name: y, dtype: float64
```

[5]:
```
data.std()
```

[5]:
```
f1      488.195035
f2    10403.417325
```

```
f3      2.926662
y       0.501255
dtype: float64
```

[6]:

```python
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

1.1 Task1

[9]:

```python
# Logistic Regression without standardization

clf_LR = SGDClassifier(loss = 'log',random_state = 0)
clf_LR.fit(X,Y)
coef_dict = {}

for coef, feat in zip(clf_LR.coef_[0,:],['f1','f2','f3']): # # PRINTING THE␣
 ↪WEIGHT COEFFICENT SVM
    coef_dict[feat] = coef

print("Coefficients of features are:",coef_dict)
s = clf_LR.score(X,Y)
print("Accuracy score is :",s)
```

```
Coefficients of features are: {'f1': -1481.8259519608932, 'f2':
14346.683837052784, 'f3': 10505.385694069439}
Accuracy score is : 0.475
```

[10]:

```python
# SVM without standardization

clf_SVM = SGDClassifier(loss = 'hinge',random_state = 0)
clf_SVM.fit(X,Y)
coef_dict = {}

for coef, feat in zip(clf_SVM.coef_[0,:],['f1','f2','f3']): # # PRINTING THE␣
 ↪WEIGHT COEFFICENT SVM
    coef_dict[feat] = coef

print("Coefficients of features are:",coef_dict)
s = clf_SVM.score(X,Y)
print("Accuracy score is :",s)
```

```
Coefficients of features are: {'f1': 10127.953228543716, 'f2':
14938.464404617735, 'f3': 10232.765491481385}
Accuracy score is : 0.47
```

1.1.1 Task 2

[11]:

```
df = StandardScaler().fit_transform(data)  # STANDARDIZING THE DATA
X=df[:,0:3]
Y=df[:,3]
print(X.shape)
print(Y.shape)

clf = SGDClassifier(loss = 'log',random_state = 0)
clf.fit(X,Y)

coef_dict = {}
for coef, feat in zip(clf.coef_[0,:],['f1','f2','f3']):
    coef_dict[feat] = coef

print("Coefficients of features are:",coef_dict)
s = clf.score(X,Y)
print("Accuracy score is :",s)
```

```
(200, 3)
(200,)
Coefficients of features are: {'f1': 1.6799267124700588, 'f2':
0.45235761381388767, 'f3': 9.618068818831835}
Accuracy score is : 0.91
```

[12]:

```
clf = SGDClassifier(loss = 'hinge',random_state = 0)
clf.fit(X,Y)
coef_dict = {}
for coef, feat in zip(clf.coef_[0,:],['f1','f2','f3']):
    coef_dict[feat] = coef

print("Coefficients of features are:",coef_dict)
s = clf.score(X,Y)
print("Accuracy score is :",s)
```

```
Coefficients of features are: {'f1': 0.08723915774337873, 'f2':
0.4659565548762608, 'f3': 9.980699843870536}
Accuracy score is : 0.91
```

1.1.2 Observations:

Before Standardization :

- feature importance for LR & SVM - F2>F3>F1
- we can see that accuracy is only 47.5% for LR and 47% for SVM

- Std Dev, correlation and weights of the features are:

Feature	sd	corr	weight in LR	weight in SVM
F1	488.195035	0.067172	-1481.8259519608932	10127.953228543716
F1	10403.4173	-0.017944	14346.683837052784	14938.464404617735
F1	2.926662	0.839060	10505.385694069439	10232.765491481385

As we can clearly see feature having low correlation and high variance(Standard deviation) i.e., F2 has the highest weight for both the models. Highly variance nature of data is affecting the classifier behavior and also on the accuracy of the model

After Standardization :

- After standardzing the data we can see that feature importance is changed to F3>F1>F2 from both LR and SVM.
- Also accuracy is significantly improved; 91% for both

we know after Standardization,std deviation will be 1 which improved the accuracy of the model.