```
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 SGDClassifier

from sklearn.linear_model import LogisticRegression
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)
```

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

```
data.head()
```

	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

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

```
f1    0.067172
```

```
f2   -0.017944
f3    0.839060
y     1.000000
Name: y, dtype: float64
```

```
data.std()
```

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

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

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

What if our features are with different variance

* As part of this task you will observe how linear models work in case of data having feautres with different variance

* from the output of the above cells you can observe that var(F2)>>var(F1)>>Var(F3)

> Task1:

1. Apply Logistic regression(SGDClassifier with logloss) on 'data' and check the feature importance

2. Apply SVM(SGDClassifier with hinge) on 'data' and check the feature importance

> Task2:

1. Apply Logistic regression(SGDClassifier with logloss) on 'data' after standardization

i.e standardization(data, column wise): (column-mean(column))/std(column) and check the feature importance

2. Apply SVM(SGDClassifier with hinge) on 'data' after standardization

i.e standardization(data, column wise): (column-mean(column))/std(column) and check the feature importance

TASK-1

```
# SGD CLASSIER WITH LOG LOSS

clf = SGDClassifier(loss = 'log',random_state = 40)
```

```
# FITTING THE DATA

clf.fit(X,Y)
```

```
SGDClassifier(loss='log', random_state=40)
```

```
coef_dict = {}

# CHECKING COFFICIENT FEATURE IN ALL THREE FEATURES
for coef, feature in zip(clf.coef_[0,:],['f1','f2','f3']): # PRINTING THE WEIGHT COEFFICENT LOGISTIC REGRESSION
    coef_dict[feature] = coef
```

```
coef_dict
```

```
{'f1': 6982.256767938532, 'f2': -2771.299563022494, 'f3': 11010.407795163874}
```

```
# SGD CLASSIFIER WITH HINGE LOSS

clf = SGDClassifier(loss = 'hinge',random_state = 40)
```

```
clf.fit(X,Y)
```

```
SGDClassifier(random_state=40)
```

```
coef_dict = {}

# for cofficient of feature in all three features
for coef, feature in zip(clf.coef_[0,:],['f1','f2','f3']): # PRINTING THE WEIGHT COEFFICENT LOGISTIC REGRESSION
    coef_dict[feature] = coef
```

```
coef_dict
```

```
{'f1': 9181.53524288595, 'f2': -3053.349870421418, 'f3': 10897.419417418072}
```

feature-3 is most imp feature with high variance

OBSERVATION

. **1. AS WE CAN SEE THAT F3>F1>F2 SO F3 IS MORE IMP FEATURE BECAUSE IT GIVING HIGH VARIANCE THAN F2 AND F1**

2. YES IT GIVING VARIANCE IN NEGATIVE VALUE BUT THAT NOT PROBLEM BECAUSE WE HAVE NOT STANDERIZED THE DATA YET

3 SO AT CONCLUSION F3 IS MORE IMP FEATURE THAN F1 AND F2 IN SGD CLASSIFIER WITH LOG LOSS AND HINGE LOSS

TASK-2

SO AS PER INSTRUCTION ABOVE PERFORMING TASK 2 AFTER STANDERIZED THE DATA

```
# STANDERIZING THE DATASET

df = StandardScaler().fit_transform(data[['f1','f2','f3']])
```

```
# SGD CLASSIFIER WITH LOG LOSS

clf = SGDClassifier(loss = 'log',random_state = 40)
```

```
# FITTING THE DATA

clf.fit(df,Y)
```

```
SGDClassifier(loss='log', random_state=40)
```

```
coef_dict = {}

# cofficient feature
for coef, feature in zip(clf.coef_[0,:],['f1','f2','f3']):
    coef_dict[feature] = coef
```

```
coef_dict
```

```
{'f1': -0.7591517430653005,
 'f2': -1.7950410095356744,
 'f3': 17.265182664963948}
```

FEATURE 3 IS MOST IMP FEATURE HERE

```
clf = SGDClassifier(loss = 'hinge',random_state = 40)
```

```
clf.fit(df,Y)
```

```
SGDClassifier(random_state=40)
```

```
coef_dict = {}

# for cofficient feature in all three featues

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

```
coef_dict
```

```
{'f1': -2.0026595276079937, 'f2': 0.9754735153716872, 'f3': 13.00240336239749}
```

HERE ALSO FEATURE-3 S MOST IMP WITH NO VARIANCE

OBSERVATION

1. AFTER HAVING STANDERIZATION OF DATASET NOW F2 IS MOST IMP FEATURE F3>F2F1

2.SO WE NEED TO KEEP IN MIND THE NEGATIVE VALUE BECAUSE DATA IS NOT STANDERIZED

3. SO AT CONCLUSION F3 IS MOST IMP FEATURE

Make sure you write the observations for each task, why a particular feautre got more importance than others

FINAL OBSERVATION

1. F3 IS MOST IMP FEATURE

2. WHEN DATA WAS NOT STANDARDRIZED VARIANCE WAS VERY HIGH

3. VARIANCE IS REMOVED AFTER STADRIZATION

4. IN BOTH THE CLASSIFIER LOGESTIC REG AND SVM F3 IS MOST IMP FEATURE

5. SGD CLASSIFIER TEND TO PERORM WELL THAN LOG LOSS