```
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.preprocessing import StandardScaler
from sklearn.preprocessing import MinMaxScaler
from plotly.offline import download_plotlyjs, init_notebook_mode, plot, iplot
init_notebook_mode(connected=True)
```

```
from google.colab import files
files=files.upload()
```

Choose Files No file chosen Upload widget is only available when the cell has been executed in the current browser session. Please rerun this cell to enable.

```
Saving task_b.csv to task_b (1).csv
```

```
data = pd.read_csv('task_b (1).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 fe**
*** 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 fea
2. Apply SVM(SGDClassifier with hinge) on 'data' and check the feature importance

> **Task2:**
1. Apply Logistic regression(SGDClassifier with logloss) on 'data' after standardiza
   i.e standardization(data, column wise): (column-mean(column))/std(column) and che
2. Apply SVM(SGDClassifier with hinge) on 'data' after standardization
   i.e standardization(data, column wise): (column-mean(column))/std(column) and che

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

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

```
#Apply Logistic regression(SGDClassifier with logloss) on 'data' and check the feature imp
columns = ['f1', 'f2', 'f3', 'y']
features = data.columns.drop("y").values
features
```

```
array(['f1', 'f2', 'f3'], dtype=object)
```

```
from sklearn.linear_model import LogisticRegression
from sklearn.linear_model import SGDClassifier

clf_lr = SGDClassifier(loss ="log",random_state = 15) #Applying Logistic regression (SGDCl
```

```
clf_lr.fit(X,Y)

feature_importance = abs(clf_lr.coef_[0])
feature_importance
```

```
array([ 3925.14601273, 16033.05764291, 10502.94022174])
```

```
feature_importance_lr = np.argsort(feature_importance)[::-1]
feature_importance_lr
```

```
array([1, 2, 0])
```

```
for i in feature_importance_lr:
  print(features[i], feature_importance[i])
```

```
f2 16033.057642911668
f3 10502.940221741319
f1 3925.1460127265855
```

```
#Apply SVM(SGDClassifier with hinge) on 'data' and check the feature importance
from sklearn.linear_model import LogisticRegression
from sklearn.linear_model import SGDClassifier

clf_svm = SGDClassifier(random_state = 15)
clf_svm.fit(X,Y)

feature_importance = abs(clf_svm.coef_[0])
feature_importance
```

```
array([ 1441.65036452,  3083.88512888, 10638.5348658 ])
```

```
feature_importance_svm = np.argsort(feature_importance)[::-1]
feature_importance_svm
```

```
array([2, 1, 0])
```

```
for i in feature_importance_svm:
  print(features[i], feature_importance[i])
```

```
f3 10638.5348658014
f2 3083.88512887846
f1 1441.6503645194891
```

Observations:

1. When we are using Logistic Regression on high variance and low correlation features(F2), then imporatnce of that particular features becomes very high as compared to other features.
2. But, when we are using SVM, low variance and high correlation feature (F3) has maximum feature importance score.

3. It states, the variance of a feature leads to how much its impacting on the dependent variable.
4. Also, feature importance is completely depends on which algorithms you are using to solve real world problem.

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

```
#Apply Logistic regression(SGDClassifier with logloss) on 'data' after standardization
#i.e standardization(data, column wise): (column-mean(column))/std(column) and check the f
scaling = StandardScaler()
feature_std_X = scaling.fit_transform(data[['f1', 'f2', 'f3']])


clf_lr = SGDClassifier(loss ="log",random_state = 15)
clf_lr.fit(feature_std_X,Y)

feature_importance_T2 = abs(clf_lr.coef_[0])
feature_importance_T2
```

```
array([ 0.29741788,  0.66973479, 10.35436789])
```

```
feature_importance_std_lr_T2 = np.argsort(feature_importance)[::-1]
feature_importance_std_lr_T2
```

```
array([2, 1, 0])
```

```
for i in feature_importance_std_lr_T2:
  print(features[i], feature_importance_T2[i])
```

```
f3 10.354367890267731
f2 0.669734794120101
f1 0.297417884183002
```

```
clf_svm = SGDClassifier(random_state = 15)
clf_svm.fit(feature_std_X,Y)

feature_importance_T2 = abs(clf_svm.coef_[0])
feature_importance_T2
```

```
array([ 2.23347737,  0.46842383, 22.39791493])
```

```
feature_importance_std_svm = np.argsort(feature_importance_T2)[::-1]
```

```
feature_importance_std_svm
```

```
array([2, 0, 1])
```

```
for i in feature_importance_std_svm:
  print(features[i], feature_importance_T2[i])
```

```
f3 22.397914928450113
f1 2.2334773740281184
f2 0.46842383180192654
```

Observations:

1. After Standardization, using Logistic Regression & SVM, low standard deviation & high correlation feature (F3) has maximum feature importance.
2. Standardization do not add much information here, it is very difficult to interpret the output results and sometimes, it misleads the information.
3. Features with higher coeffiecient means they are more important.