```
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)
```

```
data = pd.read_csv('task_b.csv')
data.drop('index',axis=1,inplace=True)
```

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

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

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

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

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

Without Standardization

```
from sklearn.linear_model import SGDClassifier

sgd=SGDClassifier(loss='log')
sgd.fit(X,y)
sgd.coef_
```

```
array([[ 12350.15508145, -17054.04708804,   9931.5547418 ]])
```

With Standardization

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

sgd=SGDClassifier(loss='log')
sgd.fit(X,y)
sgd.coef_
```

```
array([[-7.00353603,  2.83144457, 16.14345001]])
```

when we scale the weights gets reduced

The importance of features get changed when scaled and when not scaled

SVM with and without Standardization

```
sgd=SGDClassifier(loss='hinge')
sgd.fit(X,y)
sgd.coef_
```

```
array([[-2.41808791,  1.29437362, 12.75283036]])
```

Double-click (or enter) to edit

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

sgd=SGDClassifier(loss='hinge')
sgd.fit(X,y)
sgd.coef_
```

```
array([[-2.06989625,  0.39792864, 20.6367455 ]])
```

```
SVM is much more accurate than Logistic Regres
```

SVM is much more accurate than Logistic Regression

✓ 0s completed at 12:34 PM