

In [11]:

```
import numpy as np
import pandas as pd
import plotly
from sklearn import linear_model
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.preprocessing import StandardScaler
from sklearn.preprocessing import MinMaxScaler
from plotly.offline import download_plotlyjs, init_notebook_mode, plot, iplot
init_notebook_mode(connected=True)
```

In [5]:

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

In [6]:

```
data.head()
```

Out[6]:

	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

In [7]:

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

Out[7]:

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

In [8]:

```
data.std()
```

Out[8]:

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

In [9]:

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

(200, 3)

(200,)

In [16]:

```
clf=linear_model.SGDClassifier(loss='log')
clf.fit(X,Y)
clf.coef_
```

Out[16]:

```
array([[ 4380.92814545, -10058.24797372,  11130.4604972 ]])
```

**more importance is given to feature with least variance and least to maximum variance**

In [17]:

```
clf=linear_model.SGDClassifier(loss='hinge')
clf.fit(X,Y)
clf.coef_
```

Out[17]:

```
array([[14328.56804778,  3909.56905184, 10740.96010545]])
```

**least importance given to feature with max variance and more to features with less variance**

In [25]:

```
from sklearn import preprocessing
Standardisation = preprocessing.StandardScaler()
x = Standardisation.fit_transform(X)
```

In [22]:

```
clf=linear_model.SGDClassifier(loss='log')
clf.fit(x,Y)
clf.coef_
```

Out[22]:

```
array([[ -2.18805229,  -1.45007284,  15.08819855]])
```

**more importance given to features with least variance**

In [24]:

```
clf=linear_model.SGDClassifier(loss='hinge')
clf.fit(x,Y)
clf.coef_
```

Out[24]:

```
array([[ 2.30827856,  0.69218093, 17.00314076]])
```

In [ ]:

```
least importance given to features with max variance
```