In [11]:

```
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)
```

In [6]:

```
data = pd.read_csv('/content/drive/MyDrive/temp/Linear model/task_b.csv')
data=data.iloc[:,1:]
```

In [5]:

```
from google.colab import drive
drive.mount('/content/drive')
```

```
Mounted at /content/drive
```

In [7]:

```
data.head()
```

Out[7]:

	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 [8]:

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

Out[8]:

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

In [104]:

```
data.std()
```

Out[104]:

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

In [29]:

```
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 havi
ng 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 th
e feature importance
    2. Apply SVM(SGDClassifier with hinge) on 'data' and check the feature importan
ce

> Task2:
    1. Apply Logistic regression(SGDClassifier with logloss) on 'data' after stand
ardization
       i.e standardization(data, column wise): (column-mean(column))/std(column) a
nd 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) a
nd check the feature importance
```

In [154]:

```
from sklearn import  linear_model

clf=linear_model.SGDClassifier(loss='log', penalty='l2', alpha=0.0001, l1_ratio=0.15,
                           fit_intercept=True, max_iter=1000, tol=0.001, shuffle=True, v
erbose=0,
                            epsilon=0.1, n_jobs=None, random_state=None, learning_rate='
optimal',
                             eta0=0.0, power_t=0.5, early_stopping=False, validation_fra
ction=0.1,
                              n_iter_no_change=5, class_weight=None, warm_start=False, a
verage=False)
clf.fit(X,Y)
score=clf.score(X, Y)
print("LR_score:",score)
feature_imp=clf.coef_
print("LR_feature importance",feature_imp)
clf1=linear_model.SGDClassifier(loss='hinge',penalty='l2', alpha=0.0001, l1_ratio=0.15,
                                fit_intercept=True, max_iter=1000, tol=0.001, shuffle=Tr
ue,
                                verbose=0, epsilon=0.1, n_jobs=None, random_state=None,
                                learning_rate='optimal', eta0=0.0, power_t=0.5, early_st
opping=False,
                                validation_fraction=0.1, n_iter_no_change=5, class_weigh
t=None,
                                warm_start=False, average=False)

clf1.fit(X,Y)
score1=clf1.score(X, Y)
print("SVM_SCORE:",score1)
feature_imp1=clf1.coef_
print("SVM_feature importance",feature_imp1)
```

```
LR_score: 0.47
LR_feature importance [[11516.82549293 27592.38403388 10754.260146   ]]
```

```
SVM_SCORE: 0.525
SVM_feature importance [[ 8202.71523435 -8846.52371832 10112.34178885]]
```

obervation.

1.As the features are not standardized , its very difficult to analyse the most import feature which favours vaule of y.

2.As SGD is a randomized classifier , for each runtime we get different feature weights ,so its difficult to analyise feature importance.

3.Score of the model is also very less(approx between 0.4-05) as the data is not standardised .

In [155]:

```
data1=data.copy()

Y=data1['y']
X1 = data1.drop(['y'], axis=1)
standardised_X=(X1-X1.mean())/X1.std()

import  sklearn

clf3=linear_model.SGDClassifier(loss='log', penalty='l2', alpha=0.0001, l1_ratio=0.15,
                                fit_intercept=True, max_iter=1000, tol=0.001, shuffle=True, v
erbose=0,
                                 epsilon=0.1, n_jobs=None, random_state=None, learning_rate='
optimal',
                                  eta0=0.0, power_t=0.5, early_stopping=False, validation_fra
ction=0.1,
                                   n_iter_no_change=5, class_weight=None, warm_start=False, a
verage=False)
clf3.fit(standardised_X,Y)
score2=clf3.score(standardised_X, Y)
print("LR_score",score2)
feature_imp2=clf3.coef_
print("LR_feature _importance:",feature_imp2)
clf4=linear_model.SGDClassifier(loss='hinge',penalty='l2', alpha=0.0001, l1_ratio=0.15,
                                      fit_intercept=True, max_iter=1000, tol=0.001, shuffle=Tr
ue,
                                      verbose=0, epsilon=0.1, n_jobs=None, random_state=None,
                                      learning_rate='optimal', eta0=0.0, power_t=0.5, early_st
opping=False,
                                      validation_fraction=0.1, n_iter_no_change=5, class_weigh
t=None,
                                      warm_start=False, average=False)

clf4.fit(standardised_X,Y)
score3=clf4.score(standardised_X, Y)
print("SVM_SCore",score3)
feature_imp3=clf4.coef_
print("SVM_feature_importance",feature_imp3)
```

```
LR_score 0.93
LR_feature _importance: [[-1.70591909  1.90908213 11.87340309]]
SVM_SCore 0.925
SVM_feature_importance [[-1.62599943 -1.82951121 15.16606146]]
```

OBSERVATION

1.Here as the data features are standardized , we can see the F3 has high importance and its easy to interpret also.

2.score of the model is also approx 0.9, this shows model fits data better if features are standarised.

3.here feature importance value are not very large in +ve or -Ve direction,so its easy to interpret the best feature .

In [ ]:

```

```

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