```
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)
         import warnings
         warnings.filterwarnings("ignore")
```

```
In [12]: data = pd.read_csv('task_b.csv')
         data=data.iloc[:,1:]
```

```
In [13]: data.head()
```

Out[13]:

	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 [14]: data.corr()['y']

Out[14]: f1    0.067172
         f2   -0.017944
         f3    0.839060
         y     1.000000
         Name: y, dtype: float64
```

```
In [15]: data.std()

Out[15]: f1      488.195035
         f2    10403.417325
         f3        2.926662
         y         0.501255
         dtype: float64
```

```
In [16]: 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 f
    eature 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 standardi
    zation
           i.e standardization(data, column wise): (column-mean(column))/std(column) and c
    heck 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 c
    heck the feature importance
```

Task 1:

Applying Logistic Regression and checking feature importance:

```
In [21]: from sklearn.linear_model import LogisticRegression
         from sklearn.linear_model import SGDClassifier

         lr_sgd = SGDClassifier(loss='log',random_state=0)
         lr_sgd.fit(X,Y)
         imp_feat=lr_sgd.coef_[0]
         imp_feat
         print('After applying Logistic Regression ')
         for i,feat in enumerate(data.columns[:-1]):
             print('feature:{} and feat_importance:{}'.format(feat,imp_feat[i]))
         acc = lr_sgd.score(X,Y)
         print('Accuracy score is : ',acc)

         After applying Logistic Regression
         feature:f1 and feat_importance:-1481.8259519609035
         feature:f2 and feat_importance:14346.683837052571
         feature:f3 and feat_importance:10505.385694069439
         Accuracy score is :  0.475
```

Applying SVM and checking feature importance:

```
In [22]: svc = SGDClassifier(loss='hinge',random_state=0)
         svc.fit(X,Y)
         imp_feat=svc.coef_[0]
         imp_feat
         print('After applying SVC ')
         for i,feat in enumerate(data.columns[:-1]):
             print('feature:{} and feat_importance:{}'.format(feat,imp_feat[i]))
         acc1 = svc.score(X,Y)
         print('Accuracy score is : ',acc1)

         After applying SVC
         feature:f1 and feat_importance:10127.953228543702
         feature:f2 and feat_importance:14938.464404617524
         feature:f3 and feat_importance:10232.765491481385
         Accuracy score is :  0.47
```

Observations from task-1:

Before Standardization

- We can observe that the variance of the features are very high with feature f2 having the highest variance and low correlation to the dependent variable.
- After applying LR and SVM, the feature importance is in the order of f2>f3>f1 and accuracy is found to be around 47% for both the models. Hence the high variance of the dataset is clearly affecting the behaviour of the models and

Applying Logistic Regression and checking feature importance after standardization:

```
In [27]: scaler = StandardScaler()
         df = scaler.fit_transform(data)
         X_s = df[:,0:3]
         Y_s = df[:,3]
         lr_sgd1 = SGDClassifier(loss='log',random_state=0)
         lr_sgd1.fit(X_s,Y_s)
         imp_feat1=lr_sgd1.coef_[0]

         print('Applying Logistic Regression after standardization:')
         for i,feat in enumerate(data.columns[:-1]):
             print('feature:{} and feat_importance:{}'.format(feat,imp_feat1[i]))

         acc2 = lr_sgd1.score(X_s,Y_s)
         print('Accuracy score is : ',acc2)

         Applying Logistic Regression after standardization:
         feature:f1 and feat_importance:1.679926712470059
         feature:f2 and feat_importance:0.4523576138138878
         feature:f3 and feat_importance:9.618068818831835
         Accuracy score is :  0.91
```

Applying SVM and checking feature importance after standardization:

```
In [29]: svc1 = SGDClassifier(loss='hinge',random_state=0)
         svc1.fit(X_s,Y_s)
         imp_feat2=svc1.coef_[0]

         print('Applying SVC after standardization:')
         for i,feat in enumerate(data.columns[:-1]):
             print('feature:{} and feat_importance:{}'.format(feat,imp_feat2[i]))
         acc3 = svc1.score(X_s,Y_s)
         print('Accuracy score is : ',acc3)

         Applying SVC after standardization:
         feature:f1 and feat_importance:0.0872391577433691
         feature:f2 and feat_importance:0.4659565548762578
         feature:f3 and feat_importance:9.980699843870545
         Accuracy score is :  0.91
```

Observations from task-2:

After Standardization:

- We can observe that for both LR and SVM is feature importance has changed from f2>f3>f1 to f3>f1>f2.
- Also the accuracy of both the models has improved massively to 91%.

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

1) From the given dataset,we can see that the feature f3 is the most correlated feature to the dependent variable and the major parameter to look at is the variance of the feature.Clearly feature f2 has more variance than other two features.

2) After applying Logistic regression and SVM to the dataset initially, we can observe that the weights of the features are leaning towards the features which has more variance and more importance is given to them. Hence we have to standardize the dataset before applying the model.

3) After standardizing the dataset, we can observe that f3 feature has more weight than the other features which is justifying