What is Classification?

- Supervised Learning is divided into regression and classification where regression is used when the target variable is continuous and classification is used when the target variable is discrete.

Classification Techniques

- Logistic Regression
- Support Vector Machines
- Principal Component Analysis
- Decision Tree
- Ensemble Techniques
- K Nearest Neighbors
- Naive Bayes

1. Logistic Regression

- Classification Algorithm that classifies based on probability.
- Uses Sigmoid curve as a cost function.
- We use sigmoid function to map predicted values to probabilities between 1 and 0.
- A sigmoid function is represented as below

$$f(x) = \frac{1}{1+e^{-x}}$$

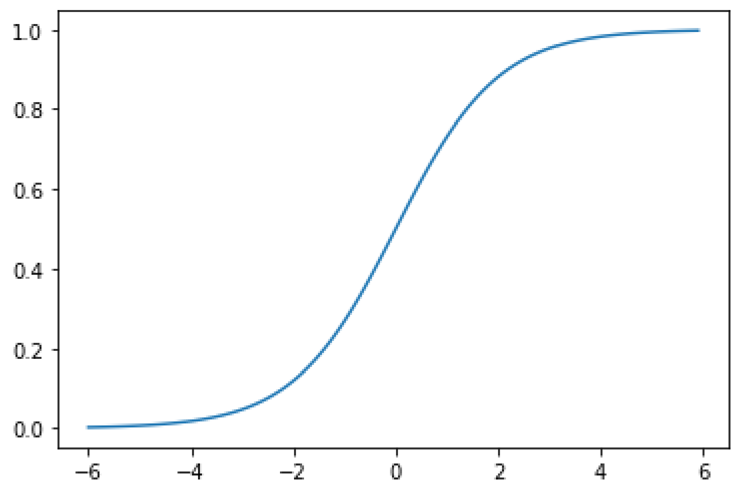
In [1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import math
import os

from sklearn.preprocessing import LabelEncoder
from sklearn.model_selection import train_test_split

from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score, auc, confusion_matrix, roc_auc_score, roc_curve, recall_score
```

In [2]:
```
array = np.arange(-6,6,0.1)
sigmoid = []
for i in array:
    sig = 1/(1+math.exp(-i))
    sigmoid.append(sig)
```

In [3]:
```
plt.plot(array,sigmoid)
plt.show()
```



In [4]:
```python
df = pd.read_csv('./bank.csv',sep = ';')
df.head()
```

Out[4]:

	age	job	marital	education	default	balance	housing	loan	contact	day	month	duration	campaign	pdays	previous	poutcome	
0	30	unemployed	married	primary	no	1787	no	no	cellular	19	oct	79	1	-1	0	unknown	n
1	33	services	married	secondary	no	4789	yes	yes	cellular	11	may	220	1	339	4	failure	n
2	35	management	single	tertiary	no	1350	yes	no	cellular	16	apr	185	1	330	1	failure	n
3	30	management	married	tertiary	no	1476	yes	yes	unknown	3	jun	199	4	-1	0	unknown	n
4	59	blue-collar	married	secondary	no	0	yes	no	unknown	5	may	226	1	-1	0	unknown	n

The classification goal is to predict if the client will subscribe (Yes/No) a term deposit (variable y).

Data

In [5]:
```python
df.shape
```

Out[5]: (4521, 17)

In [6]:
```python
df.dtypes[df.dtypes == 'object'].index
```

Out[6]:
```
Index(['job', 'marital', 'education', 'default', 'housing', 'loan', 'contact',
       'month', 'poutcome', 'y'],
      dtype='object')
```

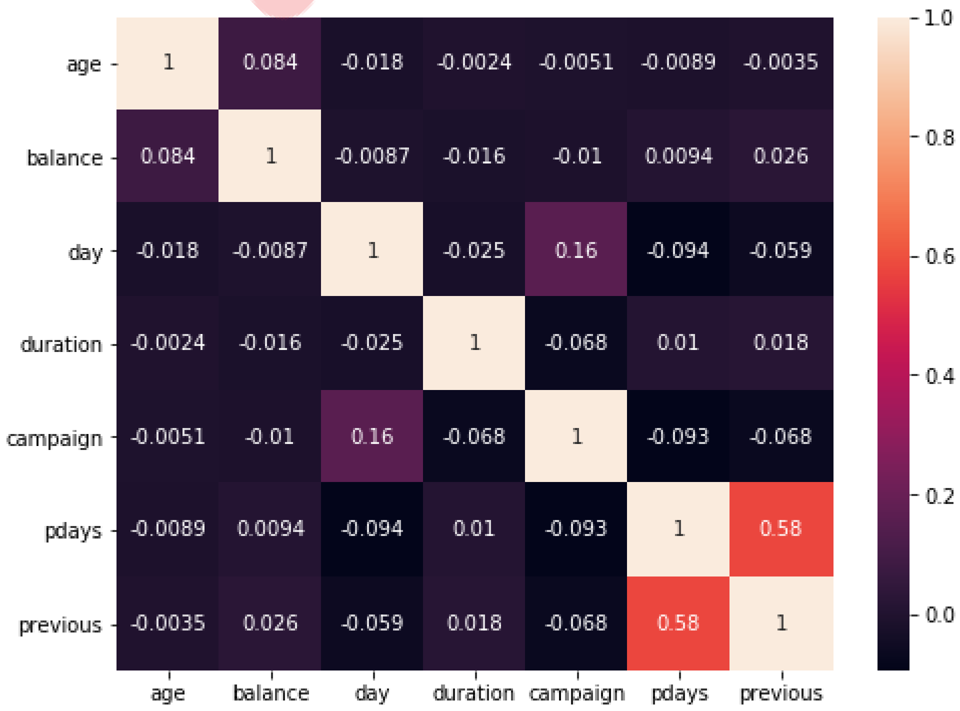
4. Checking Multi-Collineartity

In [7]:
```python
corr = df.corr()
corr
```

Out[7]:

	age	balance	day	duration	campaign	pdays	previous
age	1.000000	0.083820	-0.017853	-0.002367	-0.005148	-0.008894	-0.003511
balance	0.083820	1.000000	-0.008677	-0.015950	-0.009976	0.009437	0.026196
day	-0.017853	-0.008677	1.000000	-0.024629	0.160706	-0.094352	-0.059114
duration	-0.002367	-0.015950	-0.024629	1.000000	-0.068382	0.010380	0.018080
campaign	-0.005148	-0.009976	0.160706	-0.068382	1.000000	-0.093137	-0.067833
pdays	-0.008894	0.009437	-0.094352	0.010380	-0.093137	1.000000	0.577562
previous	-0.003511	0.026196	-0.059114	0.018080	-0.067833	0.577562	1.000000

In [8]:
```python
plt.figure(figsize=(8,6))
sns.heatmap(corr,annot=True)
plt.show()
```



In [9]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4521 entries, 0 to 4520
Data columns (total 17 columns):
age          4521 non-null int64
job          4521 non-null object
marital      4521 non-null object
education    4521 non-null object
default      4521 non-null object
balance      4521 non-null int64
housing      4521 non-null object
loan         4521 non-null object
contact      4521 non-null object
day          4521 non-null int64
month        4521 non-null object
duration     4521 non-null int64
campaign     4521 non-null int64
pdays        4521 non-null int64
previous     4521 non-null int64
poutcome     4521 non-null object
y            4521 non-null object
dtypes: int64(7), object(10)
memory usage: 600.6+ KB
```

In [10]:
```
df.dtypes[df.dtypes == 'int64'].index
```

Out[10]:
```
Index(['age', 'balance', 'day', 'duration', 'campaign', 'pdays', 'previous'], dtype='object')
```

In [11]:
```
# consider numerical data for calculating Variance Inflation Factor
num_col = df.dtypes[df.dtypes == 'int64'].index
```

In [12]:
```
X_num = df[num_col]
X_num.shape
```

Out[12]:
```
(4521, 7)
```

In [13]:
```
from statsmodels.stats.outliers_influence import variance_inflation_factor
```

In [14]:
```
vif = pd.DataFrame()
vif['Features'] = X_num.keys()
vif['Values'] = [variance_inflation_factor(X_num.values,i) for i in range(7)]
vif
```

Out[14]:

	Features	Values
0	age	5.092604
1	balance	1.231819
2	day	4.057535
3	duration	1.928720
4	campaign	1.830360
5	pdays	1.733904
6	previous	1.655651

There is no multi-collinearity effect in the dataset since all VIF values are less than ~5

Create Dummy Features

In [15]:
```
df_dum = pd.get_dummies(df,drop_first = True)
```

In [16]:
```python
df_dum.head()
```

Out[16]:

	age	balance	day	duration	campaign	pdays	previous	job_blue-collar	job_entrepreneur	job_housemaid	...	month_jun	month_mar	month_may	n
0	30	1787	19	79	1	-1	0	0	0	0	...	0	0	0	
1	33	4789	11	220	1	339	4	0	0	0	...	0	0	1	
2	35	1350	16	185	1	330	1	0	0	0	...	0	0	0	
3	30	1476	3	199	4	-1	0	0	0	0	...	1	0	0	
4	59	0	5	226	1	-1	0	1	0	0	...	0	0	1	

5 rows × 43 columns

Feature Engineering

In [17]:
```python
# Splitting data into independent and dependent
X = df_dum.iloc[:,:-1] # independent variables
y = df_dum.iloc[:,-1] # dependent variables
```

In [18]:
```python
y.value_counts()/len(y)
```

Out[18]:
```
0    0.88476
1    0.11524
Name: y_yes, dtype: float64
```

There is clearly unbalance in the dataset

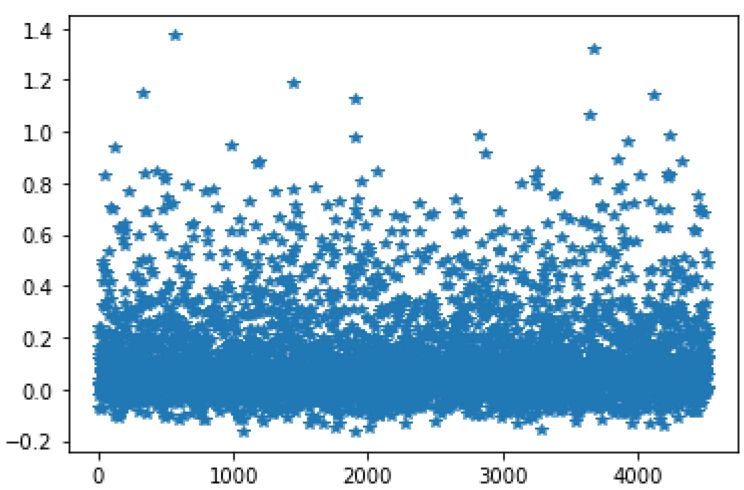
In [19]:
```python
import statsmodels.api as sm
```

In [20]:
```python
model = sm.GLM(y,X).fit()
print(model.summary())
```

```
                 Generalized Linear Model Regression Results
==============================================================================
Dep. Variable:                  y_yes   No. Observations:                 4521
Model:                            GLM   Df Residuals:                     4479
Model Family:                Gaussian   Df Model:                           41
Link Function:               identity   Scale:                        0.073097
Method:                          IRLS   Log-Likelihood:                -480.52
Date:                Sat, 22 Feb 2020   Deviance:                       327.40
Time:                        12:52:00   Pearson chi2:                     327.
No. Iterations:                     3
Covariance Type:            nonrobust
=======================================================================================
                          coef    std err          z      P>|z|      [0.025      0.975]
---------------------------------------------------------------------------------------
age                     0.0005      0.000      1.267      0.205      -0.000       0.001
balance              -7.02e-07   1.38e-06     -0.510      0.610    -3.4e-06       2e-06
day                     0.0017      0.001      3.156      0.002       0.001       0.003
duration                0.0005   1.56e-05     31.097      0.000       0.000       0.001
campaign               -0.0015      0.001     -1.069      0.285      -0.004       0.001
pdays                7.641e-06   7.52e-05      0.102      0.919      -0.000       0.000
previous                0.0005      0.003      0.152      0.879      -0.006       0.007
job_blue-collar        -0.0224      0.015     -1.453      0.146      -0.053       0.008
job_entrepreneur       -0.0151      0.025     -0.608      0.543      -0.064       0.034
job_housemaid          -0.0249      0.029     -0.854      0.393      -0.082       0.032
job_management         -0.0042      0.017     -0.241      0.809      -0.038       0.030
job_retired             0.0554      0.024      2.290      0.022       0.008       0.103
job_self-employed      -0.0065      0.024     -0.271      0.786      -0.054       0.041
job_services           -0.0075      0.018     -0.418      0.676      -0.043       0.028
job_student             0.0578      0.033      1.761      0.078      -0.007       0.122
job_technician         -0.0146      0.016     -0.925      0.355      -0.046       0.016
job_unemployed         -0.0430      0.027     -1.583      0.113      -0.096       0.010
job_unknown             0.0454      0.047      0.971      0.332      -0.046       0.137
marital_married        -0.0342      0.012     -2.738      0.006      -0.059      -0.010
marital_single         -0.0140      0.014     -0.993      0.321      -0.042       0.014
education_secondary     0.0074      0.012      0.598      0.550      -0.017       0.032
education_tertiary      0.0270      0.015      1.756      0.079      -0.003       0.057
education_unknown      -0.0253      0.023     -1.102      0.271      -0.070       0.020
default_yes             0.0495      0.032      1.568      0.117      -0.012       0.111
housing_yes            -0.0153      0.010     -1.613      0.107      -0.034       0.003
loan_yes               -0.0330      0.012     -2.867      0.004      -0.056      -0.010
contact_telephone       0.0039      0.017      0.231      0.817      -0.029       0.037
contact_unknown        -0.0776      0.014     -5.613      0.000      -0.105      -0.051
month_aug              -0.0256      0.020     -1.285      0.199      -0.065       0.013
month_dec               0.0578      0.063      0.917      0.359      -0.066       0.181
month_feb               0.0275      0.024      1.141      0.254      -0.020       0.075
month_jan              -0.0959      0.028     -3.459      0.001      -0.150      -0.042
month_jul              -0.0571      0.019     -3.000      0.003      -0.094      -0.020
month_jun               0.0404      0.023      1.792      0.073      -0.004       0.084
month_mar               0.2289      0.042      5.456      0.000       0.147       0.311
month_may              -0.0267      0.018     -1.447      0.148      -0.063       0.009
month_nov              -0.0620      0.020     -3.036      0.002      -0.102      -0.022
month_oct               0.2339      0.034      6.804      0.000       0.166       0.301
month_sep               0.1071      0.041      2.613      0.009       0.027       0.187
poutcome_other          0.0598      0.023      2.633      0.008       0.015       0.104
poutcome_success        0.4290      0.027     15.640      0.000       0.375       0.483
poutcome_unknown       -0.0020      0.022     -0.091      0.928      -0.046       0.042
=======================================================================================
```

In [21]:
```python
plt.plot(X.index,model.predict(X),'*')
```

Out[21]: [<matplotlib.lines.Line2D at 0x296069142c8>]



In [22]:
```python
features = model.pvalues.sort_values(ascending = True)
features[features < 0.025]
```

Out[22]:
```
duration            2.669761e-212
poutcome_success     3.859958e-55
month_oct            1.015051e-11
contact_unknown      1.990262e-08
month_mar            4.873139e-08
month_jan            5.431500e-04
day                  1.601168e-03
month_nov            2.399081e-03
month_jul            2.700422e-03
loan_yes             4.142675e-03
marital_married      6.180763e-03
poutcome_other       8.452288e-03
month_sep            8.965575e-03
job_retired          2.201513e-02
dtype: float64
```

In [23]:
```python
selected_features = list(features[features<0.025].index) + ['y_yes']
print(selected_features)
```

```
['duration', 'poutcome_success', 'month_oct', 'contact_unknown', 'month_mar', 'month_jan', 'day', 'month_nov', 'month_j
ul', 'loan_yes', 'marital_married', 'poutcome_other', 'month_sep', 'job_retired', 'y_yes']
```

In [24]:
```python
features_data = df_dum[selected_features]
features_data.head()
```

Out[24]:

	duration	poutcome_success	month_oct	contact_unknown	month_mar	month_jan	day	month_nov	month_jul	loan_yes	marital_married	poutc
0	79	0	1	0	0	0	19	0	0	0	1	
1	220	0	0	0	0	0	11	0	0	1	1	
2	185	0	0	0	0	0	16	0	0	0	0	
3	199	0	0	1	0	0	3	0	0	1	1	
4	226	0	0	1	0	0	5	0	0	0	1	

In [25]:
```python
#save to csv
features_data.to_csv('bank_features.csv',index=False)
```

In [ ]: