Machine Learning Phase 1

Adharsh S Mathew AM.EN.U4CSE19302 CSE - D

In [1]:
```
import pandas as pd
import numpy as np
```

In [2]:
```
raw = pd.read_table("C:\\Users\\DELL\\Jupyter Notebooks\\ML\\Project\Data\\Unprocessed\\balance_sheet_filtered_data.txt",sep=',')
raw
```

Out[2]:

	Symbol	Period Ending	Next Period Start Opening Price (Period Ending + 1 Day)	Next Period End Closing Price (Period Ending + 1 year)	Price Percentage Change	Cash and Cash Equivalents	Short-Term Investments	Net Receivables	Inventory	Other Current Assets	...	Misc. Stocks	Minority Interest	Total Liabilities	Common Stocks	Capital Surplus	Retained Earnings	Treasury Stock	Other Equity	Total Equity	Total Liabilities & Equity
0	AROW	12/31/2019	36.6893	29.9100	-18.477594	674683.0	0.0	0.0	0.0	0.0	...	0.0	0.0	2882547.0	19606.0	33218.0	-80094.0	335355.0	-6357	301728.0	3184275
1	AROW	12/31/2018	29.7672	36.6990	23.286705	687024.0	0.0	0.0	0.0	0.0	...	0.0	0.0	2718750.0	19035.0	29257.0	-79331.0	314533.0	-13910	269584.0	2988334
2	AROW	12/31/2017	31.2978	30.1819	-3.565426	708945.0	0.0	0.0	0.0	0.0	...	0.0	0.0	2510862.0	18481.0	28818.0	-79201.0	290219.0	-8714	249603.0	2760465
3	AROW	12/31/2016	36.3835	31.1148	-14.481015	749778.0	0.0	0.0	0.0	0.0	...	0.0	0.0	2372390.0	17943.0	28644.0	-77381.0	270880.0	-7234	232852.0	2605242
4	NSTG	6/30/2019	30.3900	29.3500	-3.422178	8028.0	0.0	25970.0	29576.0	1881.0	...	0.0	0.0	14736.0	212.0	70924.0	-17067.0	17103.0	0	71172.0	85908
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
7229	LIV	12/31/2016	25.2818	25.2818	0.000000	1209.0	0.0	2061.0	5175.0	62.0	...	0.0	0.0	5143.0	37.0	-24011.0	0.0	36084.0	-5840	6270.0	11412
7230	FBIO	12/31/2019	2.5900	3.1700	22.393822	136858.0	0.0	14404.0	857.0	4133.0	...	46317.0	0.0	200207.0	574.0	-436234.0	0.0	461874.0	0	26215.0	226422
7231	FBIO	12/31/2018	0.9200	2.5700	179.347826	65508.0	17604.0	7593.0	678.0	19824.0	...	17891.0	0.0	139141.0	717.0	-396274.0	0.0	397408.0	0	1852.0	140993
7232	FBIO	12/31/2017	3.9800	0.8600	-78.391960	94952.0	36002.0	8376.0	171.0	43680.0	...	67929.0	0.0	193377.0	551.0	-312127.0	0.0	364148.0	0	52573.0	245950
7233	FBIO	12/31/2016	2.7500	4.0700	48.000000	88294.0	2357.0	8689.0	203.0	10091.0	...	44473.0	0.0	132236.0	49.0	-245251.0	0.0	283697.0	0	38495.0	170731

7234 rows × 35 columns

Data Preprocessing

*Normalization
*Standardization
*Imputing Missing values

In [3]:
```
array = np.array(raw.values)
X = np.delete(array,4,1)
y = X[:,3]
X = np.delete(X,3,1)
```

In [4]:
```
print(X,y)
X.shape
y.shape
```

```
[['AROW' '12/31/2019' 36.6893 ... -6357 301728.0 3184275]
 ['AROW' '12/31/2018' 29.7672 ... -13910 269584.0 2988334]
 ['AROW' '12/31/2017' 31.2978 ... -8714 249603.0 2760465]
 ...
 ['FBIO' '12/31/2018' 0.92 ... 0 1852.0 140993]
 ['FBIO' '12/31/2017' 3.98 ... 0 52573.0 245950]
 ['FBIO' '12/31/2016' 2.75 ... 0 38495.0 170731]] [29.91 36.699 30.1819 ... 2.57 0.86 4.07]
```

Out[4]:
```
(7234,)
```

In [5]:
```
from sklearn.impute import SimpleImputer
imputer = SimpleImputer(missing_values = np.nan,strategy = 'mean')
imputer.fit(X[:,2:])
X[:,2:] = imputer.transform(X[:,2:])
```

In [6]:
```
from sklearn.preprocessing import LabelEncoder
le = LabelEncoder()
X[:,0] = le.fit_transform(X[:,0])
print(X)
```

```
[[102 '12/31/2019' 36.6893 ... -6357.0 301728.0 3184275.0]
 [102 '12/31/2018' 29.7672 ... -13910.0 269584.0 2988334.0]
 [102 '12/31/2017' 31.2978 ... -8714.0 249603.0 2760465.0]
 ...
 [704 '12/31/2018' 0.92 ... 0.0 1852.0 140993.0]
 [704 '12/31/2017' 3.98 ... 0.0 52573.0 245950.0]
 [704 '12/31/2016' 2.75 ... 0.0 38495.0 170731.0]]
```

In [7]:
```
from sklearn.model_selection import train_test_split
X_train,X_test,y_train,y_test = train_test_split(X,y,test_size = 0.2,random_state = 1)
```

In [8]:
```
from sklearn.preprocessing import StandardScaler
scalar = StandardScaler()
X_train[:,2:] = scalar.fit_transform(X_train[:,2:])
X_test[:,2:] = scalar.transform(X_test[:,2:])
```

In [9]:
```
X_train
```

Out[9]:
```
array([[408, '10/31/2018', -0.013119859560394532, ...,
        -0.05514636637800353, -0.05692340151781441, -0.11983850911349436],
       [1318, '1/29/2017', -0.013202557678810624, ...,
        0.031337316856191205, 0.07465009630409784, -0.09704755221994561],
       [509, '12/31/2018', -0.013209917221534341, ...,
        0.03442378541627339, -0.19795320842373307, -0.1468505796101211],
       ...,
       [220, '12/31/2019', -0.013211217069304839, ...,
        0.030485201279400947, -0.14483844975728907, -0.1332564664403115],
       [521, '2/1/2019', -0.013198628081215 42, ..., -0.06271655615712403,
        -0.543280033011899, 0.5263629389144375],
       [1756, '12/31/2016', -0.013215946862203844, ...,
        1.2776032705758626, 0.08526432078614586, -0.027651344222637184]],
      dtype=object)
```

In [10]:
```
pd.DataFrame(X_train).to_csv("C:\\Users\\DELL\\Jupyter Notebooks\\ML\\Project\Data\\Processed\\Train\\X_Train.csv")
pd.DataFrame(X_test).to_csv("C:\\Users\\DELL\\Jupyter Notebooks\\ML\\Project\Data\\Processed\\Test\\X_Test.csv")
pd.DataFrame(y_train).to_csv("C:\\Users\\DELL\\Jupyter Notebooks\\ML\\Project\Data\\Processed\\Train\\Y_Train.csv")
pd.DataFrame(y_test).to_csv("C:\\Users\\DELL\\Jupyter Notebooks\\ML\\Project\Data\\Processed\\Test\\Y_Test.csv")
```

Data Summarization

In [ ]:

In [ ]:

Data Visualization

In [ ]:

In [ ]:

Data Interpretation

In [ ]:

In [ ]:

In [ ]: