```
# Machine learning
from sklearn.svm import SVC
from sklearn.metrics import accuracy_score

# For data manipulation
import pandas as pd
import numpy as np

# To plot
import matplotlib.pyplot as plt
plt.style.use('seaborn-darkgrid')

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

```
<ipython-input-1-ecc59adf5d07>:11: MatplotlibDeprecationWarning: The seaborn styles shipped by Matplotlib are deprecated since 3.6, as t
  plt.style.use('seaborn-darkgrid')
```

```
# Read the csv file using read_csv
# method of pandas
df = pd.read_csv('/content/Reliance.csv')
df
```

	Date	Open	High	Low	Close	Adj Close	Volume
0	2015-11-18	463.799988	465.649994	454.975006	456.000000	436.671021	5142766.0
1	2015-11-19	459.450012	469.350006	458.625000	467.375000	447.563873	5569752.0
2	2015-11-20	467.000000	476.399994	462.774994	473.424988	453.357422	5167930.0
3	2015-11-23	475.000000	478.950012	473.100006	476.875000	456.661224	4800026.0
4	2015-11-24	476.500000	485.799988	475.524994	483.850006	463.340515	6768886.0
...	...	...	...	...	...	...	...
1228	2020-11-10	2077.000000	2090.000000	2041.199951	2084.550049	2084.550049	17045147.0
1229	2020-11-11	2089.000000	2095.000000	1978.099976	1997.199951	1997.199951	26178477.0

```
# Changes The Date column as index columns
df.index = pd.to_datetime(df['Date'])
df
```

```
# drop The original date column
df = df.drop(['Date'], axis='columns')
df
```

Date	Open	High	Low	Close	Adj Close	Volume
2015-11-18	463.799988	465.649994	454.975006	456.000000	436.671021	5142766.0
2015-11-19	459.450012	469.350006	458.625000	467.375000	447.563873	5569752.0

```
# Create predictor variables
df['Open-Close'] = df.Open - df.Close
df['High-Low'] = df.High - df.Low

# Store all predictor variables in a variable X
X = df[['Open-Close', 'High-Low']]
X.head()
```

Date	Open-Close	High-Low
2015-11-18	7.799988	10.674988
2015-11-19	-7.924988	10.725006
2015-11-20	-6.424988	13.625000
2015-11-23	-1.875000	5.850006
2015-11-24	-7.350006	10.274994

```
X.isnull().sum()
```

```
Open-Close    1
High-Low      1
dtype: int64
```

```
X['Open-Close'].fillna(X['Open-Close'].mean(), inplace=True)
X['High-Low'].fillna(X['High-Low'].mean(), inplace=True)
```

```
X.isnull().sum()
```

```
Open-Close    0
High-Low      0
dtype: int64
```

```
# Target variables
y = np.where(df['Close'].shift(-1) > df['Close'], 1, 0)
y
```

```
array([1, 1, 1, ..., 1, 0, 0])
```

```
split_percentage = 0.8
split = int(split_percentage*len(df))

# Train data set
X_train = X[:split]
y_train = y[:split]

# Test data set
X_test = X[split:]
y_test = y[split:]
```

```
# Support vector classifier
cls = SVC().fit(X_train, y_train)
```

```
df['Predicted_Signal'] = cls.predict(X)


# Calculate daily returns
df['Return'] = df.Close.pct_change()


# Calculate strategy returns
df['Strategy_Return'] = df.Return *df.Predicted_Signal.shift(1)


# Calculate Cumulutive returns
df['Cum_Ret'] = df['Return'].cumsum()
df
```

Date	Open	High	Low	Close	Adj Close	Volume	Op Cl
2015-11-18	463.799988	465.649994	454.975006	456.000000	436.671021	5142766.0	7.799
2015-11-19	459.450012	469.350006	458.625000	467.375000	447.563873	5569752.0	-7.924
2015-11-20	467.000000	476.399994	462.774994	473.424988	453.357422	5167930.0	-6.424
2015-11-23	475.000000	478.950012	473.100006	476.875000	456.661224	4800026.0	-1.875
2015-11-24	476.500000	485.799988	475.524994	483.850006	463.340515	6768886.0	-7.350
...	...	...	...	...	...	...	
2020-11-10	2077.000000	2090.000000	2041.199951	2084.550049	2084.550049	17045147.0	-7.550
2020-11-11	2089.000000	2095.000000	1978.099976	1997.199951	1997.199951	26178477.0	91.800
2020-11-12	1981.000000	2008.449951	1965.000000	1980.000000	1980.000000	18481466.0	1.000
2020-11-13	1982.000000	2036.650024	1981.750000	1996.400024	1996.400024	20946864.0	-14.400
2020-11-17	2085.000000	2085.000000	1985.000000	1993.250000	1993.250000	21479385.0	91.750

1233 rows × 12 columns

```
# Plot Strategy Cumulative returns
df['Cum_Strategy'] = df['Strategy_Return'].cumsum()
df
```

Date	Open	High	Low	Close	Adj Close	Volume	Op Cl
2015-11-18	463.799988	465.649994	454.975006	456.000000	436.671021	5142766.0	7.799
2015-11-19	459.450012	469.350006	458.625000	467.375000	447.563873	5569752.0	-7.924
2015-11-20	467.000000	476.399994	462.774994	473.424988	453.357422	5167930.0	-6.424
2015-11-23	475.000000	478.950012	473.100006	476.875000	456.661224	4800026.0	-1.875
2015-11-24	476.500000	485.799988	475.524994	483.850006	463.340515	6768886.0	-7.350

```
import matplotlib.pyplot as plt
%matplotlib inline

plt.plot(df['Cum_Ret'],color='red')
plt.plot(df['Cum_Strategy'],color='blue')
```

```
[<matplotlib.lines.Line2D at 0x7e6033c0fe20>]
```