Data Preprocessing

```
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import r2_score, accuracy_score, precision_score,
recall_score, classification_report from sklearn.svm import SVR
from sklearn.neural_network import MLPRegressor
from sklearn.preprocessing import StandardScaler

# Download from https://www.nseindia.com/reports-indices-
historicalindex-data

nifty = pd.read_csv("NIFTY.csv")
nifty
```

```
{"summary":"{\n  \"name\": \"nifty\",\n  \"rows\": 8341,\n
\"fields\": [\n    {\n      \"column\": \"Index Name\",\n
\"properties\": {\n        \"dtype\": \"category\",\n
\"num_unique_values\": 1,\n        \"samples\": [\n          \"NIFTY
50\"\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"Date\",\n      \"properties\": {\n        \"dtype\": \"object\",\n
\"num_unique_values\": 8341,\n        \"samples\": [\n          \"09
Aug 1996\"\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"Open\",\n      \"properties\": {\n        \"dtype\": \"string\",\n
\"num_unique_values\": 7021,\n        \"samples\": [\n
\"2786.65\"\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"High\",\n      \"properties\": {\n        \"dtype\": \"string\",\n
\"num_unique_values\": 7040,\n        \"samples\": [\n
\"10242.95\"\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"Low\",\n      \"properties\": {\n        \"dtype\": \"string\",\n
\"num_unique_values\": 7014,\n        \"samples\": [\n
\"5282.70\"\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"Close\",\n      \"properties\": {\n        \"dtype\": \"number\",\n
\"std\": 5790.477107889755,\n        \"min\": 279.02,\n
\"max\": 26216.05,\n        \"num_unique_values\": 8088,\n
\"samples\": [\n          668.67\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    }\n  ]\n}","type":"dataframe","variable_name":"nifty"}
```

```
nifty['Date'] = pd.to_datetime(nifty['Date'], format='%d %b %Y')
nifty.set_index('Date', inplace=True)
```

```
nifty['Open'] = pd.to_numeric(nifty['Open'], errors='coerce')
nifty['High'] = pd.to_numeric(nifty['High'], errors='coerce')
nifty['Low'] = pd.to_numeric(nifty['Low'], errors='coerce')
nifty = nifty.sort_index(ascending=True)
nifty.dropna(inplace=True) nifty.head()
```

```
{"summary":"{\n  \"name\": \"nifty\",\n  \"rows\": 7216,\n
\"fields\": [\n    {\n      \"column\": \"Date\",\n
\"properties\": {\n        \"dtype\": \"date\",\n        \"min\":
\"1995-11-03 00:00:00\",\n        \"max\": \"2024-11-04 00:00:00\",\n
\"num_unique_values\": 7216,\n        \"samples\": [\n
\"1997-01-29 00:00:00\",\n          \"1997-05-20 00:00:00\",\n
\"2015-08-18 00:00:00\"\n        ],\n        \"semantic_type\": \"\",\
n        \"description\": \"\"\n      }\n    },\n    {\n
\"column\": \"Index Name\",\n      \"properties\": {\n
\"dtype\": \"category\",\n        \"num_unique_values\": 1,\n
\"samples\":  [\n                   \"NIFTY  50\"\n                ],\n
\"semantic_type\": \"\",\n          \"description\": \"\"\n        }\ n
},\n    {\n      \"column\": \"Open\",\n      \"properties\": {\n
\"dtype\": \"number\",\n        \"std\": 5864.198759362866,\n
\"min\": 786.37,\n        \"max\": 26248.25,\n
\"num_unique_values\": 7020,\n        \"samples\": [\n
5360.05\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"High\",\n      \"properties\": {\n        \"dtype\": \"number\",\n
\"std\": 5884.937044002429,\n        \"min\": 805.82,\n
\"max\": 26277.35,\n        \"num_unique_values\": 7039,\n
\"samples\": [\n          1003.8\n        ],\n
\"semantic_type\": \"\",\n          \"description\": \"\"\n       }\ n
},\n    {\n      \"column\": \"Low\",\n      \"properties\": {\n
\"dtype\": \"number\",\n        \"std\": 5833.34489986645,\n
\"min\": 775.43,\n        \"max\": 26151.4,\n
\"num_unique_values\": 7013,\n        \"samples\": [\n
9726.35\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"Close\",\n      \"properties\": {\n        \"dtype\": \"number\",\n
\"std\": 5860.07750996219,\n        \"min\": 788.15,\n        \"max\":
26216.05,\n        \"num_unique_values\": 7018,\n        \"samples\":
[\n          5281.2\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    }\n  ]\
n}","type":"dataframe","variable_name":"nifty"}
```

Feature Engineering

```
return_periods = [1, 2, 3, 4, 7, 10, 14, 15, 16, 30, 90, 180, 365]

# Create new DataFrame to store returns
returns_df = nifty.copy()
```

```
for period in return_periods:
returns_df[f"{period}D_return"] =
nifty["Close"].pct_change(periods=period) * 100
returns_df.head()
```

```
{"summary":"{\n  \"name\": \"returns_df\",\n  \"rows\": 7216,\n
\"fields\": [\n    {\n      \"column\": \"Date\",\n
\"properties\": {\n        \"dtype\": \"date\",\n        \"min\":
\"1995-11-03 00:00:00\",\n        \"max\": \"2024-11-04 00:00:00\",\n
\"num_unique_values\": 7216,\n        \"samples\": [\n
\"1997-01-29 00:00:00\",\n          \"1997-05-20 00:00:00\",\n
\"2015-08-18 00:00:00\"\n        ],\n        \"semantic_type\": \"\",\
n        \"description\": \"\"\n      }\n    },\n    {\n
\"column\": \"Index Name\",\n      \"properties\": {\n
\"dtype\": \"category\",\n        \"num_unique_values\": 1,\n
\"samples\":  [\n                  \"NIFTY  50\"\n              ],\n
\"semantic_type\": \"\",\n          \"description\": \"\"\n       }\ n
},\n    {\n      \"column\": \"Open\",\n      \"properties\": {\n
\"dtype\": \"number\",\n        \"std\": 5864.198759362866,\n
\"min\": 786.37,\n        \"max\": 26248.25,\n
\"num_unique_values\": 7020,\n        \"samples\": [\n
5360.05\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"High\",\n      \"properties\": {\n        \"dtype\": \"number\",\n
\"std\": 5884.937044002429,\n        \"min\": 805.82,\n
\"max\": 26277.35,\n        \"num_unique_values\": 7039,\n
\"samples\": [\n          1003.8\n        ],\n
\"semantic_type\": \"\",\n          \"description\": \"\"\n       }\ n
},\n    {\n      \"column\": \"Low\",\n      \"properties\": {\n
\"dtype\": \"number\",\n        \"std\": 5833.34489986645,\n
\"min\": 775.43,\n        \"max\": 26151.4,\n
\"num_unique_values\": 7013,\n        \"samples\": [\n
9726.35\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"Close\",\n      \"properties\": {\n        \"dtype\": \"number\",\n
\"std\": 5860.07750996219,\n        \"min\": 788.15,\n        \"max\":
26216.05,\n        \"num_unique_values\": 7018,\n        \"samples\":
[\n          5281.2\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"1D_return\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": 1.441745590698286,\n        \"min\":
12.980464127060365,\n        \"max\": 17.74406601936458,\n
\"num_unique_values\": 7206,\n        \"samples\": [\n
2.642962193961406 7\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"2D_return\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": 2.0850178969774884,\n        \"min\":
19.141193595342067,\n        \"max\": 20.30639079436196,\n
```

```
\"num_unique_values\": 7208,\n        \"samples\": [\n          -
```

```
4.379669681478571\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"3D_return\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": 2.550296171035094,\n        \"min\": -
20.121178398095772,\n        \"max\": 20.175597267250133,\n
\"num_unique_values\": 7212,\n        \"samples\": [\n          -
6.783326779394416\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"4D_return\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": 2.9461074446126885,\n        \"min\":
21.969767226189386,\n        \"max\": 20.586720545123207,\n
\"num_unique_values\": 7211,\n        \"samples\": [\n          -
5.7904050334250945\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"7D_return\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": 3.8914736429948213,\n        \"min\":
24.22795722391968,\n        \"max\": 21.758091511279787,\n
\"num_unique_values\": 7209,\n        \"samples\": [\n          -
3.77506881635864\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"10D_return\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": 4.653093012568373,\n        \"min\":
30.74949155781227,\n        \"max\": 26.763363274922703,\n
\"num_unique_values\": 7206,\n        \"samples\": [\n
2.2193813395925854\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"11D_return\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": 4.906349585507159,\n        \"min\":
32.46738841068417,\n        \"max\": 26.179628430785762,\n
\"num_unique_values\": 7201,\n        \"samples\": [\n
0.3929775236026112\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"14D_return\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": 5.5675348654226875,\n        \"min\":
32.32590419820339,\n        \"max\": 28.496744799618902,\n
\"num_unique_values\": 7202,\n        \"samples\": [\n          -
1.9495349984469668\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"15D_return\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": 5.779657047077776,\n        \"min\":
34.594697209390624,\n        \"max\": 27.794403253978906,\n
\"num_unique_values\": 7200,\n        \"samples\": [\n
4.456760567999218\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"16D_return\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": 5.985041895524804,\n        \"min\":
36.10833386065937,\n        \"max\": 29.81262949283847,\n
\"num_unique_values\": 7200,\n        \"samples\": [\n
```

```
5.4003970880211805\n        ],\n        \"semantic_type\": \"\",\n
```

```
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"30D_return\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": 8.23347138693337,\n        \"min\": -
41.27606386000795,\n        \"max\": 45.00444907073182,\n
\"num_unique_values\": 7186,\n        \"samples\": [\n          -
17.37212776552608\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"60D_return\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": 11.967644041436598,\n        \"min\":
41.02020426287745,\n        \"max\": 80.2343431203,\n
\"num_unique_values\": 7156,\n        \"samples\": [\n          -
22.187660711027057\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"90D_return\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": 15.044416696960594,\n        \"min\":
45.7511283043198,\n        \"max\": 75.80591881545963,\n
\"num_unique_values\": 7126,\n        \"samples\": [\n
11.399607011232437\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"180D_return\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": 22.24255197960949,\n        \"min\":
52.52809252903287,\n        \"max\": 111.78590640129103,\n
\"num_unique_values\": 7036,\n        \"samples\": [\n
39.020482892779505\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"365D_return\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": 31.482684997702904,\n        \"min\":
47.042777433354,\n        \"max\": 128.2349462895437,\n
\"num_unique_values\": 6851,\n        \"samples\": [\n          -
17.121622560255613\n        ],\n        \"semantic_type\": \"\",\n
\"description\":  \"\"\n                }\n            }\n     ]\
n}","type":"dataframe","variable_name":"returns_df"}
```

```
returns_df.dropna().head()
```

```
{"summary":"{\n  \"name\": \"returns_df\",\n  \"rows\": 5,\n
\"fields\": [\n    {\n      \"column\": \"Date\",\n
\"properties\": {\n        \"dtype\": \"date\",\n        \"min\":
\"1997-04-25 00:00:00\",\n        \"max\": \"1997-05-02 00:00:00\",\n
\"num_unique_values\": 5,\n        \"samples\": [\n          \"1997-
04-28 00:00:00\",\n          \"1997-05-02 00:00:00\",\n
\"1997-04-29 00:00:00\"\n        ],\n        \"semantic_type\": \"\",\
n        \"description\": \"\"\n      }\n    },\n    {\n
\"column\": \"Index Name\",\n      \"properties\": {\n
\"dtype\": \"category\",\n        \"num_unique_values\": 1,\n
\"samples\":  [\n                    \"NIFTY  50\"\n                ],\n
\"semantic_type\": \"\",\n          \"description\": \"\"\n       }\  n
},\n    {\n      \"column\": \"Open\",\n      \"properties\": {\n
\"dtype\": \"number\",\n        \"std\": 7.194928769626548,\n
```

```
\"min\": 1063.6,\n        \"max\": 1081.1,\n
```

```
\"num_unique_values\": 5,\n        \"samples\": [\n          1068.6\n
],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n
}\n    },\n    {\n      \"column\": \"High\",\n      \"properties\":
{\n        \"dtype\": \"number\",\n        \"std\":
8.680293773830485,\n        \"min\": 1065.4,\n        \"max\":
1088.4,\n        \"num_unique_values\": 5,\n        \"samples\": [\n
1070.7\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"Low\",\n      \"properties\": {\n        \"dtype\": \"number\",\n
\"std\": 8.291607202466869,\n        \"min\": 1052.8,\n
\"max\": 1075.95,\n        \"num_unique_values\": 5,\n
\"samples\": [\n          1061.2\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\ n
},\n    {\n      \"column\": \"Close\",\n      \"properties\": {\ n
\"dtype\": \"number\",\n        \"std\": 8.080887946259319,\n
\"min\": 1057.45,\n        \"max\": 1079.85,\n
\"num_unique_values\": 5,\n        \"samples\": [\n          1065.4\n
],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n
}\n    },\n    {\n      \"column\": \"1D_return\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
1.247801197229764,\n        \"min\": -1.004769180904752,\n
\"max\": 2.1183034658849076,\n        \"num_unique_values\": 5,\n
\"samples\": [\n          -0.37404151860856993\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"2D_return\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
1.0161277732743277,\n        \"min\": -1.117449036843099,\n
\"max\": 1.3562981039984878,\n        \"num_unique_values\": 5,\n
\"samples\": [\n          -0.16866566716641218\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"3D_return\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
0.848665550687769,\n        \"min\": -0.9136056971514206,\n
\"max\": 0.9771834673648616,\n        \"num_unique_values\": 5,\n
\"samples\": [\n          -0.5275197236356721\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"4D_return\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
0.9592643444798763,\n        \"min\": -1.2697819896363316,\n
\"max\": 1.1853448275861878,\n        \"num_unique_values\": 5,\n
\"samples\": [\n          0.5284015852047741\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"7D_return\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
2.402997555598542,\n        \"min\": 0.8680883185506749,\n
\"max\": 6.593570894592582,\n        \"num_unique_values\": 5,\n
```

```
\"samples\": [\n          4.975859690609918\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"10D_return\",\n
```

```
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
1.4783109211037806,\n        \"min\": 3.7950111617975457,\n
\"max\": 7.635185646648379,\n        \"num_unique_values\": 5,\n
\"samples\": [\n          5.657757723012846\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"11D_return\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
1.747794272165197,\n        \"min\": 3.406774725808037,\n
\"max\": 7.92209102835808,\n        \"num_unique_values\": 5,\n
\"samples\": [\n          3.406774725808037\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"14D_return\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
1.9244070767620272,\n        \"min\": 3.756187518198595,\n
\"max\": 8.976687859521636,\n        \"num_unique_values\": 5,\n
\"samples\": [\n          5.417305694355123\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"15D_return\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
2.0817851020874163,\n        \"min\": 4.630683223667953,\n
\"max\": 10.230376745864046,\n        \"num_unique_values\": 5,\n
\"samples\": [\n          6.3061265216523665\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"16D_return\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
2.1310818408584744,\n        \"min\": 5.512871682298948,\n
\"max\": 10.440979035422915,\n        \"num_unique_values\": 5,\n
\"samples\": [\n          9.818069370715875\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"30D_return\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
2.4126090912318383,\n        \"min\": -5.131958305610995,\n
\"max\": 0.5259728169800582,\n        \"num_unique_values\": 5,\n
\"samples\": [\n          -1.2329656067488592\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"60D_return\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
1.8269010166780226,\n        \"min\": 3.4889410843609303,\n
\"max\": 8.469418805152662,\n        \"num_unique_values\": 5,\n
\"samples\": [\n          5.573997919040785\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"90D_return\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
2.5262943124305344,\n        \"min\": 26.327547278005415,\n
\"max\": 32.64442231075697,\n        \"num_unique_values\": 5,\n
\"samples\": [\n          32.64442231075697\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"180D_return\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
```

1.4646072254527112,\n   \"min\": 2.008431166375657,\n
\"max\": 5.3689489708447224,\n   \"num_unique_values\": 5,\n
\"samples\": [\n   2.1662623104880208\n   ],\n
\"semantic_type\": \"\",\n   \"description\": \"\"\n   }\
n   },\n   {\n   \"column\": \"365D_return\",\n
\"properties\": {\n   \"dtype\": \"number\",\n   \"std\":
2.500204926358817,\n   \"min\": 6.940000000000013,\n
\"max\": 12.163849454919351,\n   \"num_unique_values\": 5,\n
\"samples\": [\n   7.733689277191291\n   ],\n
\"semantic_type\": \"\",\n   \"description\": \"\"\n   }\
n   }\n]\n}","type":"dataframe"}

```python
# Create target variable (return after 1 month in future)
returns_df["Target_30D_return"] = nifty["Close"].pct_change(periods=-
15) * 100

returns_df.tail(30)
```

```
{"type":"dataframe"}
```

```python
# Drop NaN values
data_clean = returns_df.dropna()
data_clean.head()
```

```
{"type":"dataframe","variable_name":"data_clean"}
```

```python
# Define features and target
X = data_clean[[f"{period}D_return" for period in return_periods]]
y = data_clean["Target_30D_return"]

X.head()
```

{"summary":"{\n \"name\": \"X\",\n \"rows\": 6836,\n \"fields\": [\
n   {\n   \"column\": \"Date\",\n   \"properties\": {\n
\"dtype\": \"date\",\n   \"min\": \"1997-04-25 00:00:00\",\n
\"max\": \"2024-10-14 00:00:00\",\n   \"num_unique_values\":
6836,\n   \"samples\": [\n   \"2010-04-16 00:00:00\",\n
\"2012-02-24 00:00:00\",\n   \"1999-01-28 00:00:00\"\
n   ],\n   \"semantic_type\": \"\",\n
\"description\": \"\"\n   }\n   },\n   {\n   \"column\":
\"1D_return\",\n   \"properties\": {\n   \"dtype\":
\"number\",\n   \"std\": 1.4275621613449585,\n   \"min\":
12.980464127060365,\n   \"max\": 17.74406601936458,\n
\"num_unique_values\": 6828,\n   \"samples\": [\n
1.068144327078202,\n   1.8090488407221716,\n
1.0645272063184885\n   ],\n   \"semantic_type\": \"\",\n
\"description\": \"\"\n   }\n   },\n   {\n   \"column\":
\"2D_return\",\n   \"properties\": {\n   \"dtype\":
\"number\",\n   \"std\": 2.0597168161804493,\n   \"min\":
19.141193595342067,\n   \"max\": 20.30639079436196,\n
\"num_unique_values\": 6831,\n   \"samples\": [\n

```
0.2636203866432485,\n          1.300068098805185,\n          -
0.3917149015788879\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"3D_return\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": 2.515040979596444,\n        \"min\":
20.121178398095772,\n        \"max\": 20.175597267250133,\n
\"num_unique_values\": 6835,\n        \"samples\": [\n
0.18515386057229577,\n          3.664084171893811,\n          -
0.2037511101823264\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"4D_return\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": 2.900479828045757,\n        \"min\": -
21.969767226189386,\n        \"max\": 20.58672054512307,\n
\"num_unique_values\":   6835,\n                         \"samples\":   [\n
0.5608545317590585,\n          1.0621044799826063,\n          -
1.2510339123242398\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"7D_return\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": 3.8154227117114132,\n        \"min\":
24.22795722391968,\n        \"max\": 20.134473639790706,\n
\"num_unique_values\":   6836,\n                         \"samples\":   [\n
1.9269474468878056,\n          0.24464323630690732,\n
2.2262656534303815\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"10D_return\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": 4.556902797389828,\n        \"min\":
30.74949155781227,\n        \"max\": 21.836300309597533,\n
\"num_unique_values\": 6836,\n        \"samples\": [\n
0.257186946333654,\n          0.31317265143606665,\n          -
0.9694644616102477\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"11D_return\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": 4.804229183151426,\n        \"min\":
32.46738841068417,\n        \"max\": 24.444796269376347,\n
\"num_unique_values\": 6832,\n        \"samples\": [\n
2.4254629832319186,\n          3.520615529733795,\n          -
3.094561688311692\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"14D_return\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": 5.45724535623917,\n        \"min\": -
32.32590419820339,\n        \"max\": 27.00343509747647,\n
\"num_unique_values\": 6836,\n        \"samples\": [\n
0.04182191468331542,\n          1.9424129481679042,\n
2.8925397252895335\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"15D_return\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": 5.66921937678019,\n        \"min\": -
34.59469720939062 4,\n        \"max\": 27.43768637821413,\n
```

```
\"num_unique_values\": 6835,\n        \"samples\": [\n
3.4752717416966394,\n          -2.1557201246408475,\n
5.279982363315683\n        ],\n         \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"16D_return\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": 5.873366643395172,\n        \"min\":
36.10833386065937,\n        \"max\": 29.81262949283847,\n
\"num_unique_values\": 6836,\n        \"samples\": [\n
1.1027434104357248,\n          3.697690853181057,\n
6.382267765649363\n        ],\n         \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"30D_return\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": 8.17229352700231,\n        \"min\": -
41.27606386000795,\n        \"max\": 45.00444907073182,\n
\"num_unique_values\": 6836,\n        \"samples\": [\n
4.89535579031295,\n          11.692158940124875,\n
10.595182955071802\n        ],\n         \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"60D_return\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": 11.832884626381052,\n        \"min\":
41.02020426287745,\n        \"max\": 80.2343431203,\n
\"num_unique_values\": 6836,\n        \"samples\": [\n
0.19801226152851292,\n          12.360178392193788,\n
16.02283770651116\n        ],\n         \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"90D_return\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": 14.964777334305989,\n        \"min\":
45.7511283043198,\n        \"max\": 75.80591881545963,\n
\"num_unique_values\": 6836,\n        \"samples\": [\n
2.7450214759859426,\n          5.786879176977178,\n
4.944511592132739\n        ],\n         \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"180D_return\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": 22.377074769687997,\n        \"min\":
52.52809252903287,\n        \"max\": 111.78590640129103,\n
\"num_unique_values\": 6836,\n        \"samples\": [\n
17.75525273545009,\n          -1.857358483744472,\n          -
12.704505986655569\n        ],\n         \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"365D_return\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": 31.51134083806962,\n        \"min\":
47.042777433354,\n        \"max\": 128.2349462895437,\n
\"num_unique_values\": 6836,\n        \"samples\": [\n
50.76059243131752,\n          -3.183929670016139,\n          -
24.981345481679305\n        ],\n         \"semantic_type\": \"\",\n
\"description\":   \"\"\n                  }\n            }\n      ]\
n}","type":"dataframe","variable_name":"X"} y.head()
```

```
Date
                                                1997-04-25    -1.770080
                                                1997-04-28    -1.384593
                                                1997-04-29    -1.482726
                                                1997-04-30    -4.112032

1997-05-02   -3.414917
Name: Target_30D_return, dtype: float64
```

Train-Test Split

```
# Split last 1 year as test data
split_date = X.index.max() - pd.DateOffset(years=1)
X_train = X[X.index < split_date]
X_test = X[X.index >= split_date]
y_train = y[y.index < split_date]
y_test = y[y.index >= split_date]

model = SVR(kernel='rbf')
model.fit(X_train, y_train)


# Predict
y_pred = model.predict(X_test)
```

Convert it to a Binary Classification Dataset

```
# Convert to binary classification (Positive: 1, Negative: 0)
y_test_binary = (y_test > 0).astype(int) y_pred_binary =
(y_pred > 0).astype(int) y_test_binary
Date

2023-10-16    0
2023-10-17    0
2023-10-18    0
2023-10-19    0
2023-10-20    0
             ..
2024-10-08    0
2024-10-09    0
2024-10-10    0
2024-10-11    0
2024-10-14    0
Name: Target_30D_return, Length: 248, dtype: int64
# Create DataFrame for actual and predicted values
y_results_df = pd.DataFrame({
    "Actual": y_test,
    "Predicted": y_pred,
    "Actual_Class": y_test_binary,
    "Predicted_Class": y_pred_binary
}, index=y_test.index)
y_results_df.tail(25)
```

```
{"summary":"{\n  \"name\": \"y_results_df\",\n  \"rows\": 25,\n
\"fields\": [\n    {\n      \"column\": \"Date\",\n
\"properties\": {\n        \"dtype\": \"date\",\n        \"min\":
\"2024-09-09 00:00:00\",\n        \"max\": \"2024-10-14 00:00:00\",\n
\"num_unique_values\": 25,\n        \"samples\": [\n          \"2024-
09-19 00:00:00\",\n          \"2024-10-01 00:00:00\",\n
\"2024-09-09 00:00:00\"\n        ],\n        \"semantic_type\": \"\",\
n        \"description\": \"\"\n      }\n    },\n    {\n
\"column\": \"Actual\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": 2.3906017705905414,\n        \"min\":
5.64079076392896,\n        \"max\": 3.3879163220118524,\n
\"num_unique_values\":   25,\n                        \"samples\":   [\n
1.8087865647876455,\n          -5.571402263100822,\n
3.3879163220118524\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"Predicted\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": 0.3164604716004378,\n        \"min\":
0.4021402670969816,\n        \"max\": 0.6379131087343479,\n
\"num_unique_values\":   25,\n                        \"samples\":   [\n
0.02796444738699,\n          -0.05059304028900613,\n
0.560615077414414\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n        }\n      },\n      {\n        \"column\":
\"Actual_Class\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": 0,\n        \"min\": 0,\n
\"max\": 1,\n        \"num_unique_values\": 2,\n        \"samples\": [\n
0,\n          1\n        ],\n        \"semantic_type\":
\"\",\n        \"description\": \"\"\n      }\n    },\n    {\n
\"column\": \"Predicted_Class\",\n      \"properties\": {\n
\"dtype\": \"number\",\n        \"std\": 0,\n        \"min\": 0,\n
\"max\": 1,\n        \"num_unique_values\": 2,\n        \"samples\": [\n
0,\n          1\n        ],\n        \"semantic_type\":
\"\",\n        \"description\": \"\"\n      }\n    }\n  ]\
n}","type":"dataframe"}
```

Evaluation Metric

```
accuracy = accuracy_score(y_test_binary, y_pred_binary)
print(f"Accuracy: {accuracy:.2f}")
Accuracy: 0.73
# Compute average actual return grouped by predicted class
average_actual_by_class = y_results_df.groupby("Predicted_Class")
["Actual"].mean()
```

```
print("Average Actual Return by Predicted Class:")
print(average_actual_by_class)
```

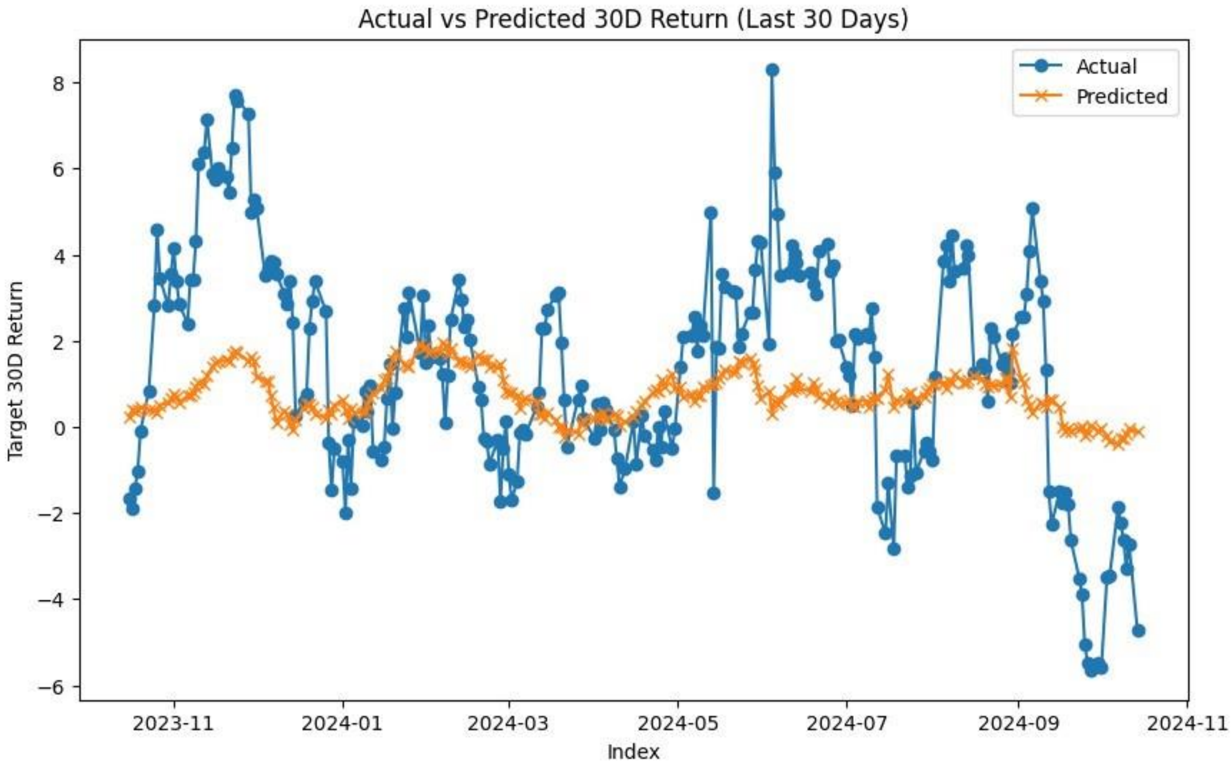
```
Average Actual Return by Predicted Class:
Predicted_Class
0    -2.144021
1    1.741150
Name: Actual, dtype: float64
```

```
(1.741150/3)*52
```

```
30.179933333333334
```

```
# Select last 30 days for plotting
last_30_days = y_test.index[-350:]
y_test_last_30 = y_test.loc[last_30_days]
y_pred_last_30 = pd.Series(y_pred,
index=y_test.index).loc[last_30_days]

# Plot Index vs Target for both Actual and Prediction (Last 30 Days)
plt.figure(figsize=(10,6))
plt.plot(y_test_last_30.index, y_test_last_30, label="Actual",
marker='o')
plt.plot(y_pred_last_30.index, y_pred_last_30, label="Predicted",
marker='x') plt.xlabel("Index")
plt.ylabel("Target 30D Return")
plt.title("Actual vs Predicted 30D Return (Last 30 Days)")
plt.legend() plt.show()
```


Actual vs Predicted 30D Return (Last 30 Days)
Actual
Predicted
Target 30D Return
8
6
4
2
0
−2
−4
−6
2023-11
2024-01
2024-03
2024-05
2024-07
2024-09
2024-11
Index