PRISM ALGORITHM

INTRODUCTION

The Prism Algorithm is a rule-based classification algorithm that generates if-then rules to classify data. It follows a separate-and-conquer approach- focusing on producing rules that cover all classes.

The modus operand of the PRISM algorithm maybe described as follows:
For i=1 to n
repeat until all instances of class i have been removed

(1) Calculate the probability of occurrence of class i for each attribute-value pair.
(2) Select the attribute-value pair with maximum probability and create a subset of the training set comprising all instances with the selected combination.
(3) Repeat (1) and (2) for this subset until it contains only instances of class i. The induced rule is then the conjunction of all the attribute-value pairs selected in creating this subset.
(4) Remove all instances covered by this rule from training set. [1]

PROBLEM OVERVIEW

To build a set of rules that determine whether the opening price of the stock on the next day will be greater than it's closing price on the current day. The training data spans over 200 market days, and the validation data spans over the next 65 days.
The prediction problem requires classifying whether the next day's opening price will be greater than the current day's closing price (YES or NO).

WHY PRISM ALGORITHM?

The PRISM algorithm is a natural choice for this problem due to several key factors:

- Rule Accuracy and Interpretability: Since stock movements can be volatile, a precise rule-based model helps identify clear patterns in price movement, leading to more accurate predictions.
- Class-Based Rule Generation: PRISM allows focusing on one class at a time (YES or NO), making it easier to isolate specific patterns that predict upward or downward movements.
- Efficient Handling of Features: The dataset contains a mix of numerical prices and indicators (EMAs, RSIs), which can be effectively handled by PRISM's rule-based approach to maximize accuracy.
- Flexibility: The stock market is dynamic, and new patterns may emerge. PRISM allows for easy modification or addition of rules without overhauling the entire model.

DATA PREPROCESSING

Initially, when the data was visualized- it could be observed that the “Stock Splits” column contained only 1 unique value. Thus, the column could be dropped without negatively affecting the prediction of the problem.

```
Unnamed: 0: 200 unique values
Open: 200 unique values
High: 200 unique values
Low: 200 unique values
Close: 200 unique values
Volume: 200 unique values
Dividends: 2 unique values
Stock Splits: 1 unique values
EMA_08: 200 unique values
EMA_12: 200 unique values
EMA_20: 200 unique values
RSI_05: 200 unique values
RSI_09: 200 unique values
RSI_14: 200 unique values
```

The dataset does not explicitly provide us with the Target Variable- this must therefore be calculated for each individual row in the dataset. This was done using the code in the function createClassLabel():

```
def create_class_label(df, is_training=True):
    df = df1.copy()
    df['Next_Day_Open'] = df['Open'].shift(-1)
    if is_training:
        # For the last day in training data, as per the user's note
        df.loc[df.index[-1], 'Next_Day_Open'] = df.loc[df.index[-1], 'Close'] + 1  # Ensuring it's greater
    else:
        df = df[:-1]  # Remove the last row since we don't have the next day's opening price
    df['Class'] = np.where(df['Next_Day_Open'] > df['Close'], 'YES', 'NO')
    df = df.drop(columns=['Next_Day_Open', 'Unnamed: 0', "Stock Splits"])
    return df
```

The dataset in itself has 128 YES and 72 Nos- meaning that the dataset is imbalanced.

RULES

.....
Target: NO
.....
High = Bin_3 AND RSI_14 = Bin_7
Close = Bin_2 AND RSI_05 = Bin_9
Volume = Bin_6 AND RSI_14 = Bin_4

Open = Bin_9 AND Volume = Bin_9
RSI_05 = Bin_1 AND Open = Bin_1
RSI_14 = Bin_2 AND Close = Bin_9
RSI_09 = Bin_6 AND Volume = Bin_0
RSI_14 = Bin_0 AND Open = Bin_1
Volume = Bin_3 AND RSI_14 = Bin_5
Close = Bin_2 AND RSI_05 = Bin_7
RSI_05 = Bin_4 AND High = Bin_5
RSI_05 = Bin_1 AND High = Bin_8
RSI_14 = Bin_2 AND RSI_05 = Bin_4
Close = Bin_2 AND Volume = Bin_3
Volume = Bin_6 AND RSI_09 = Bin_1
EMA_12 = Bin_2 AND Open = Bin_3
RSI_14 = Bin_0 AND Volume = Bin_2
EMA_08 = Bin_4 AND High = Bin_3
RSI_05 = Bin_1 AND EMA_20 = Bin_1
RSI_14 = Bin_2 AND Open = Bin_7
RSI_05 = Bin_6 AND Open = Bin_1
Close = Bin_2 AND RSI_05 = Bin_2
Open = Bin_9 AND RSI_05 = Bin_8
EMA_08 = Bin_4 AND Low = Bin_3
EMA_12 = Bin_4 AND Volume = Bin_9
EMA_08 = Bin_6 AND EMA_12 = Bin_5
RSI_09 = Bin_1 AND Close = Bin_6
Volume = Bin_4 AND EMA_08 = Bin_7
Close = Bin_2 AND RSI_05 = Bin_1
EMA_08 = Bin_6 AND RSI_05 = Bin_6 AND Open = Bin_5
EMA_08 = Bin_4 AND RSI_05 = Bin_6
EMA_20 = Bin_9 AND EMA_12 = Bin_8
RSI_09 = Bin_1 AND Volume = Bin_0 AND EMA_08 = Bin_9
EMA_08 = Bin_6 AND Open = Bin_7 AND High = Bin_6
Close = Bin_2 AND Volume = Bin_4 AND RSI_05 = Bin_5
EMA_08 = Bin_4 AND Close = Bin_5 AND High = Bin_4
High = Bin_0 AND Volume = Bin_8
Volume = Bin_7 AND RSI_05 = Bin_4
EMA_08 = Bin_6 AND Volume = Bin_4 AND RSI_05 = Bin_6
RSI_05 = Bin_0 AND EMA_08 = Bin_7
Close = Bin_2 AND RSI_09 = Bin_3 AND Volume = Bin_4
High = Bin_0 AND Volume = Bin_9 AND EMA_08 = Bin_0 AND RSI_05 = Bin_0
Volume = Bin_3 AND High = Bin_4
Volume = Bin_0 AND RSI_09 = Bin_9

.....
Target: YES
.....
EMA_08 = Bin_5 AND Open = Bin_5

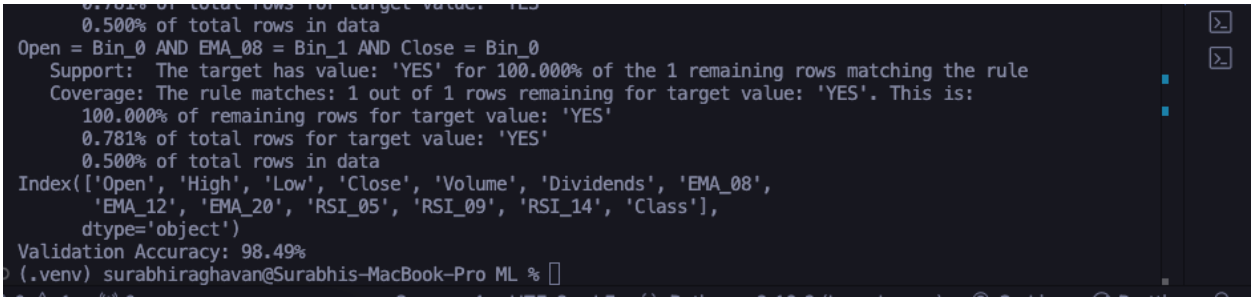
EMA_20 = Bin_5 AND High = Bin_6
RSI_09 = Bin_8 AND Open = Bin_6
RSI_05 = Bin_5 AND High = Bin_1
EMA_20 = Bin_7 AND Close = Bin_8
Close = Bin_7 AND Volume = Bin_1
Volume = Bin_2 AND Open = Bin_2
RSI_09 = Bin_8 AND EMA_12 = Bin_3
Close = Bin_7 AND Volume = Bin_8
RSI_05 = Bin_5 AND Close = Bin_3
Volume = Bin_4 AND RSI_05 = Bin_7
Volume = Bin_2 AND High = Bin_9
Dividends = 0.004 AND Open = Bin_3
Close = Bin_7 AND Open = Bin_6
Open = Bin_0 AND High = Bin_1
Close = Bin_0 AND Volume = Bin_0
RSI_05 = Bin_0 AND High = Bin_8
High = Bin_4 AND EMA_08 = Bin_3
EMA_20 = Bin_7 AND RSI_05 = Bin_1
Open = Bin_2 AND Low = Bin_1
RSI_14 = Bin_3 AND RSI_05 = Bin_2
EMA_08 = Bin_8 AND RSI_05 = Bin_6
RSI_05 = Bin_8 AND EMA_12 = Bin_1
RSI_09 = Bin_5 AND Volume = Bin_0
Close = Bin_0 AND Volume = Bin_7
High = Bin_6 AND Volume = Bin_2
EMA_20 = Bin_7 AND Open = Bin_6
Dividends = 0.004 AND Open = Bin_8
Close = Bin_7 AND High = Bin_6
Volume = Bin_4 AND High = Bin_3
RSI_09 = Bin_5 AND Open = Bin_1
Volume = Bin_5 AND Open = Bin_4
RSI_05 = Bin_0 AND Open = Bin_8
RSI_09 = Bin_3 AND EMA_08 = Bin_3
RSI_05 = Bin_8 AND Open = Bin_7
Open = Bin_0 AND EMA_20 = Bin_2
Low = Bin_8 AND RSI_05 = Bin_3
High = Bin_2 AND Close = Bin_3
Volume = Bin_9 AND RSI_14 = Bin_2
RSI_14 = Bin_6 AND Open = Bin_8
RSI_09 = Bin_5 AND Open = Bin_3
Volume = Bin_8 AND Open = Bin_2
Low = Bin_5 AND Open = Bin_4
Volume = Bin_2 AND Open = Bin_3
RSI_05 = Bin_5 AND Open = Bin_4
Open = Bin_0 AND Volume = Bin_8
EMA_12 = Bin_1 AND RSI_09 = Bin_4

RSI_05 = Bin_0 AND Volume = Bin_1
RSI_14 = Bin_4 AND Open = Bin_9
High = Bin_6 AND Volume = Bin_3
EMA_20 = Bin_7 AND Volume = Bin_9 AND Open = Bin_7
Volume = Bin_4 AND RSI_05 = Bin_4
Open = Bin_0 AND EMA_08 = Bin_1 AND Close = Bin_0

BIN VALUES

The data in the dataset is not discrete- and therefore must be discretized. Thus I chose to discretise my data using quantile binning. Quantile binning ensures that in each range there are equal amount of datapoints.

ACCURACY



ACCURACY: 98.49%

REFERENCES

[1] Yao, J, and Y Yao. *A GRANULAR COMPUTING APPROACH to MACHINE LEARNING*.