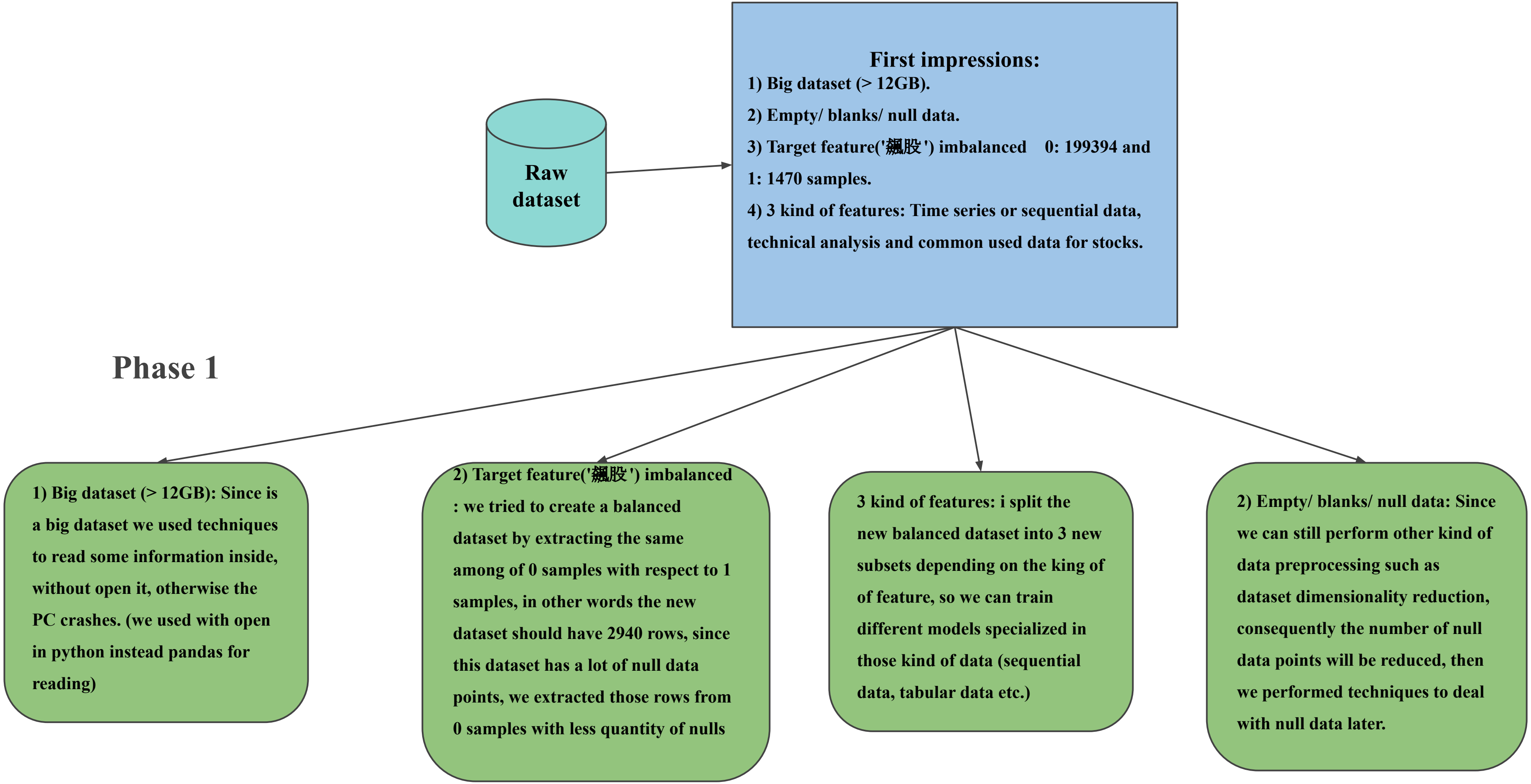
IA Go competition

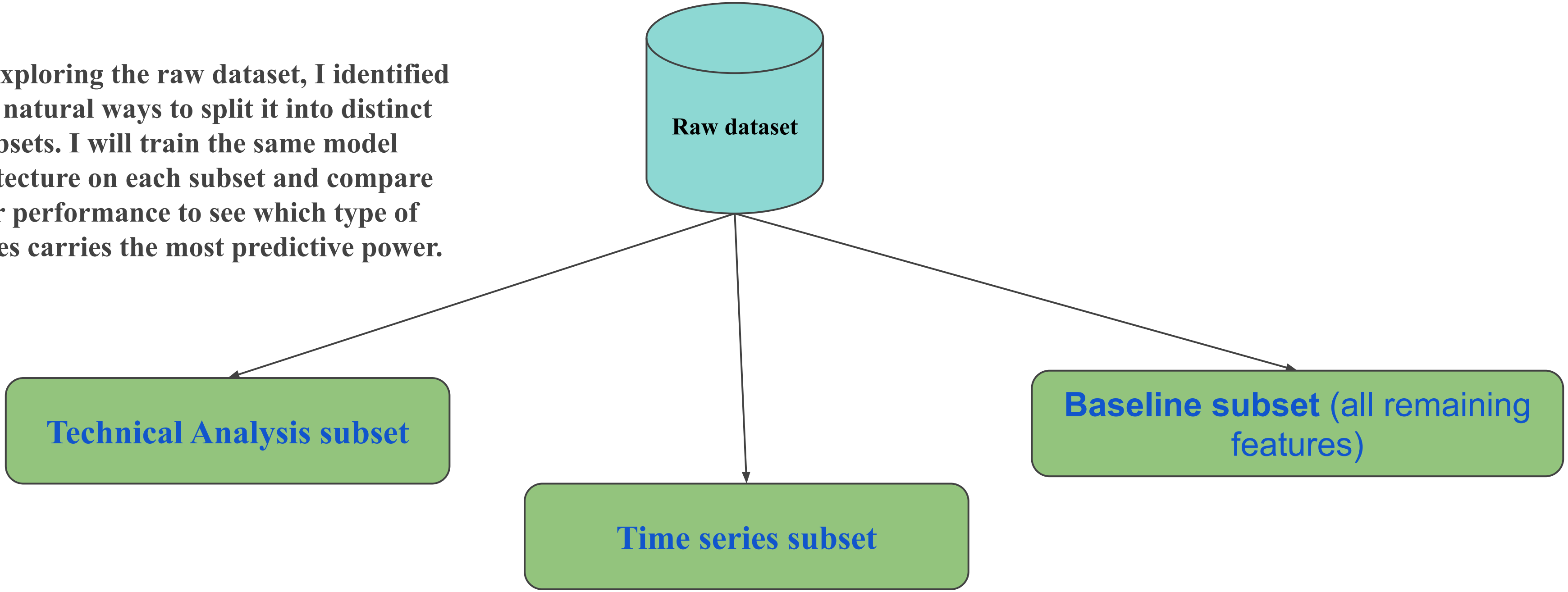
Raw dataset first impressions

Phase 1



Subsets

After exploring the raw dataset, I identified three natural ways to split it into distinct subsets. I will train the same model architecture on each subset and compare their performance to see which type of features carries the most predictive power.



Subsets description

Technical Analysis subset: **this dataset has a total of 2940 rows and 30 columns.**

```
---> The dataset has 2940 rows & 30 columns.

---> Analyzing target column: '飆股'
---> Number of unique classes: 2
---> Samples per class:
     Class 0: 1470 samples
     Class 1: 1470 samples
```

```
---> Dataset features:
Index(['技術指標_乖離率(60日)', '技術指標_乖離率(250日)', '技術指標_乖離率(20日)', '個股1天報酬率',
       '個股5天報酬率', '個股10天報酬率', '個股20天報酬率', '個股5天波動度', '個股10天波動度', '個股20天波動度',
       '個股5天乖離率', '個股10天乖離率', '個股19天乖離率', '個股5天成交量波動度', '個股10天成交量波動度',
       '個股20天成交量波動度', '上市加權指數1天報酬率', '上市加權指數5天報酬率', '上市加權指數10天報酬率',
       '上市加權指數20天報酬率', '上市加權指數5天波動度', '上市加權指數10天波動度', '上市加權指數20天波動度',
       '上市加權指數5天乖離率', '上市加權指數10天乖離率', '上市加權指數19天乖離率', '上市加權指數5天成交量波動度',
       '上市加權指數10天成交量波動度', '上市加權指數20天成交量波動度', '飆股'],
      dtype='object')
```

Since this dataset only has two columns with null values, I could delete them, because there is a column called 技術指標_乖離率(20日) with no null values that represents the same metric over 20 periods, then i could try dimensionality-reduction or redundancy-reduction techniques (*these techniques cannot be applied when a column has missing values)*.

```
=== DATASET DESCRIPTION ===
       技術指標_乖離率(60日)  技術指標_乖離率(250日)  技術指標_乖離率(20日)     個股1天報酬率   ...  上市加權指數5天成交量波動度  上市加權指數10天成交量波動度  上市加權指數20天成交量波動度          飆股
count    2934.000000     2900.000000    2940.000000  2940.000000  ...     2940.000000      2940.000000      2940.000000  2940.000000
mean        2.145741        2.225756       2.107961     1.612497  ...        1.280190         1.238852         1.243960     0.500000
std         1.592988        1.611365       1.560352     1.525553  ...        1.055891         1.032863         0.978809     0.500085
min        -2.291800       -2.203200      -3.581200    -3.426300  ...       -0.515700        -0.379800        -0.154600     0.000000
25%         1.091775        1.098025       1.086675     0.835900  ...        0.585800         0.564000         0.644175     0.000000
50%         1.792800        1.877250       1.735300     1.292900  ...        1.164200         0.972450         0.992100     0.500000
75%         2.951975        3.066425       2.819525     2.088600  ...        1.624100         1.670500         1.646000     1.000000
max        13.985400       17.010600      10.805500     5.818500  ...        8.060900         7.334400         5.862500     1.000000
```

```
Columns with null values:
技術指標_乖離率(60日)      6
技術指標_乖離率(250日)    40
dtype: int64
a total of 2 null columns.
```

```
=== NULL PERCENTAGES ===
技術指標_乖離率(60日)     0.20
技術指標_乖離率(250日)    1.36
dtype: float64
```

Time series subset: this dataset has a total of 2940 rows and 465 columns. (since this is a classification task and the target column has only binary values , we can not apply a time series transformation to train sequential models like CNN, Transformers etc…, however i can use this data to try Traditional ML models or Deep learning non sequential models.

```
---> The dataset has 2940 rows & 465 columns.

---> Analyzing target column: '飆股'
---> Number of unique classes: 2
---> Samples per class:
     Class 0: 1470 samples
     Class 1: 1470 samples
```

```
---> Dataset features:
Index(['外資券商_前1天分點進出', '外資券商_前1天分點買賣力', '外資券商_前1天分點成交力(%)', '外資券商_前1天分點吃貨比(%)',
       '外資券商_前1天分點出貨比(%)', '主力券商_前1天分點進出', '主力券商_前1天分點買賣力', '主力券商_前1天分點成交力(%)',
       '主力券商_前1天分點吃貨比(%)', '主力券商_前1天分點出貨比(%)',
       ...
       '賣超第15名分點前1天賣均張', '賣超第15名分點前1天買均價', '賣超第15名分點前1天賣均價',
       '賣超第15名分點前1天買均值(千)', '賣超第15名分點前1天賣均值(千)', '個股前1天收盤價', '個股前1天成交量',
       '上市加權指數前1天收盤價', '上市加權指數前1天成交量', '飆股'],
      dtype='object', length=465)
```

This subset has 458 columns with null values, which is a lot, so I need to delete some columns or impute missing data before reducing the dataset's dimensionality and redundancy.

```
=== DATASET DESCRIPTION ===
       外資券商_前1天分點進出  外資券商_前1天分點買賣力  外資券商_前1天分點成交力(%)  外資券商_前1天分點吃貨比(%)  ...     個股前1天成交量
上市加權指數前1天收盤價  上市加權指數前1天成交量          飆股
count   2940.000000    2233.000000       2929.000000       2929.000000  ...  2940.000000   2940.000000   2940.000000  2940.000000
mean       1.276847       1.361642          1.225693          1.267450  ...     1.621064      1.345265      1.353743     0.500000
std        1.781628       2.540052          0.735338          0.984561  ...     2.287573      1.098313      1.082883     0.500085
min      -52.954800      -7.073600        -37.842800          0.522800  ...     1.012000     -0.752100     -0.548000     0.000000
25%        1.215600       1.131200          1.211100          0.522800  ...     1.030375      0.279100      0.593600     0.000000
50%        1.225800       1.214700          1.223800          0.911400  ...     1.101850      1.539800      1.111200     0.500000
75%        1.258000       1.345700          1.253100          1.642000  ...     1.436100      2.217000      1.931000     1.000000
max       42.949800     101.303100          4.421000          6.791200  ...    73.647300      4.715500      7.677600     1.000000
```

```
Columns with null values:
外資券商_前1天分點買賣力          707
外資券商_前1天分點成交力(%)        11
外資券商_前1天分點吃貨比(%)        11
外資券商_前1天分點出貨比(%)        11
主力券商_前1天分點買賣力          404
                          ...
賣超第15名分點前1天賣均張          485
賣超第15名分點前1天買均價          485
賣超第15名分點前1天賣均價          485
賣超第15名分點前1天買均值(千)       485
賣超第15名分點前1天賣均值(千)       485
Length: 458, dtype: int64
a total of 458 columns with null values.
```

```
=== NULL PERCENTAGES ===
外資券商_前1天分點買賣力          24.05
外資券商_前1天分點成交力(%)        0.37
外資券商_前1天分點吃貨比(%)        0.37
外資券商_前1天分點出貨比(%)        0.37
主力券商_前1天分點買賣力          13.74
                          ...
賣超第15名分點前1天賣均張          16.50
賣超第15名分點前1天買均價          16.50
賣超第15名分點前1天賣均價          16.50
賣超第15名分點前1天買均值(千)       16.50
賣超第15名分點前1天賣均值(千)       16.50
Length: 458, dtype: float64
```

In theory, columns with more than 30 % missing values could be discarded if they do not contribute valuable information to the model. In our dataset, using a 10 % threshold would remove 452 columns; at 20 %, only 1 column; and at 30 %, none. Therefore, instead of dropping columns, we should impute the missing data.

```
Columns with more than 10.0% of null values:
外資券商_前1天分點買賣力          707
主力券商_前1天分點買賣力          404
買超第1名分點前1天券商代號         399
買超第1名分點前1天張增減          399
買超第1名分點前1天金額增減(千)      399
                         ...
賣超第15名分點前1天賣均張         485
賣超第15名分點前1天買均價         485
賣超第15名分點前1天賣均價         485
賣超第15名分點前1天買均值(千)      485
賣超第15名分點前1天賣均值(千)      485
Length: 452, dtype: int64
A total of 452 discardable columns.
```

```
Columns with more than 20.0% of null values:
外資券商_前1天分點買賣力    707
dtype: int64
A total of 1 discardable columns.
```

```
Columns with more than 30.0% of null values:
Series([], dtype: int64)
A total of 0 discardable columns.
```

Baseline subset (all remaining features)**:** **this dataset has a total of 2940 rows and 905 columns**

```
---> The dataset has 2940 rows & 905 columns.

---> Analyzing target column: '飆股'
---> Number of unique classes: 2
---> Samples per class:
     Class 0: 1470 samples
     Class 1: 1470 samples
```

```
---> Dataset features:
Index(['ID', '外資券商_分點進出', '外資券商_分點買賣力', '外資券商_分點成交力(%)', '外資券商_分點吃貨比(%)',
       '外資券商_分點出貨比(%)', '主力券商_分點進出', '主力券商_分點買賣力', '主力券商_分點成交力(%)',
       '主力券商_分點吃貨比(%)',
       ...
       '賣超第15名分點賣均張', '賣超第15名分點買均價', '賣超第15名分點賣均價', '賣超第15名分點買均值(千)',
       '賣超第15名分點賣均值(千)', '個股收盤價', '個股成交量', '上市加權指數收盤價', '上市加權指數成交量', '飆股'],
      dtype='object', length=905)
```

this seems to be the most problematic dataset to handle with, it has more than 700 columns with null data.

```
=== DATASET DESCRIPTION ===
            ID   外資券商_分點進出  外資券商_分點買賣力  外資券商_分點成交力(%)  外資券商_分點吃貨比(%)  ...      個股收盤價        個股
成交量    上市加權指數收盤價    上市加權指數成交量          飆股
count     2940 2940.000000 2225.000000   2930.000000   2930.000000  ... 2940.000000 2940.000000 2940.000000 2940.000000 2940.000000
unique    2940         NaN         NaN           NaN           NaN  ...         NaN         NaN         NaN         NaN         NaN
top     TR-191         NaN         NaN           NaN           NaN  ...         NaN         NaN         NaN         NaN         NaN
freq         1         NaN         NaN           NaN           NaN  ...         NaN         NaN         NaN         NaN         NaN
mean       NaN    1.282861    1.319383      1.314343      1.277476  ...    1.365666    1.669728    1.345130    1.350145    0.500000
std        NaN    2.080165    0.918218      0.910556      1.021353  ...    1.320179    2.411023    1.100097    1.074946    0.500085
min        NaN  -77.953400   -0.336300     -4.407300      0.531300  ...    0.792700    1.007800   -0.765300   -0.688200    0.000000
25%        NaN    1.213400    0.689400      1.073325      0.531300  ...    0.933800    1.026875    0.287400    0.558300    0.000000
50%        NaN    1.221900    1.368800      1.241300      0.899500  ...    1.055650    1.102750    1.523400    1.151600    0.500000
75%        NaN    1.256800    2.018700      1.581650      1.593300  ...    1.334225    1.482875    2.200400    1.983350    1.000000
max        NaN   35.986200    2.830300      5.813400      7.541600  ...   26.921900   82.051900    4.830500    7.746000    1.000000
```

```
Columns with null values:
外資券商_分點買賣力        715
外資券商_分點成交力(%)      10
外資券商_分點吃貨比(%)      10
外資券商_分點出貨比(%)      10
主力券商_分點買賣力        408
                         ...
賣超第15名分點賣均張       475
賣超第15名分點買均價       475
賣超第15名分點賣均價       475
賣超第15名分點買均值(千)    475
賣超第15名分點賣均值(千)    475
Length: 700, dtype: int64
a total of 700 columns with null values.
```

```
=== NULL PERCENTAGES ===
外資券商_分點買賣力        24.32
外資券商_分點成交力(%)      0.34
外資券商_分點吃貨比(%)      0.34
外資券商_分點出貨比(%)      0.34
主力券商_分點買賣力        13.88
                         ...
賣超第15名分點賣均張       16.16
賣超第15名分點買均價       16.16
賣超第15名分點賣均價       16.16
賣超第15名分點買均值(千)    16.16
賣超第15名分點賣均值(千)    16.16
Length: 700, dtype: float64
```

For this dataset, deleting columns with more than 10% missing values would discard over 54% of the features; at a 20% threshold about 4.09% would be removed; and at 30% only 2.21%. Therefore, we should drop only the columns that exceed 30% missing data and impute the remaining missing values.

```
Columns with more than 10.0% of null values:
外資券商_分點買賣力              715
主力券商_分點買賣力              408
日外資_外資及陸資(不含外資自營商)買張     492
日外資_外資及陸資(不含外資自營商)賣張     492
日外資_外資及陸資(不含外資自營商)買賣超    492
                    ...
賣超第15名分點賣均張             475
賣超第15名分點買均價             475
賣超第15名分點賣均價             475
賣超第15名分點買均值(千)          475
賣超第15名分點賣均值(千)          475
Length: 497, dtype: int64
A total of 497 discardable columns.

If you discard all these columns you might delete 54.92% of the data in the dataset.
```

```
Columns with more than 20.0% of null values:
外資券商_分點買賣力                 715
日外資_外資自營商買張                2940
```

```
A total of 37 discardable columns.

If you discard all these columns you might delete 4.09% of the data in the dataset.
```

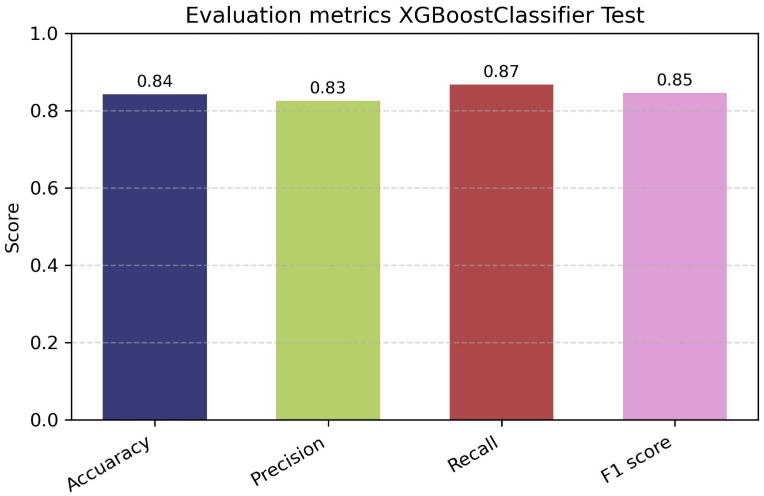
```
Columns with more than 30.0% of null values:
日外資_外資自營商買張                2940
日外資_外資自營商賣張                2940
日外資_外資自營商買賣超               2940
日外資_與前日異動原因                2916
日自營_自營商買均價                1524
日自營_自營商賣均價                1504
日投信_投信買均價                2432
日投信_投信賣均價                2590
日投信_投信持股成本                1140
月營收_預估年營收(千)               2940
月營收_累計營收達成率(%)              2940
月營收_重要子公司本月營業收入淨額(千)         2940
月營收_重要子公司本年累計營收淨額(千)         2940
季IFRS財報_固定資產對長期負債比率(%)     1142
季IFRS財報_權益對長期負債比率(%)       1142
季IFRS財報_長期負債對淨值比率(%)        919
季IFRS財報_營業利益成長率(%)         898
季IFRS財報_稅額扣抵比率(%)         2493
季IFRS財報_預計稅額扣抵比率(%)        2498
季IFRS財報_財務信評             2940
dtype: int64
A total of 20 discardable columns.

If you discard all these columns you might delete 2.21% of the data in the dataset.
```

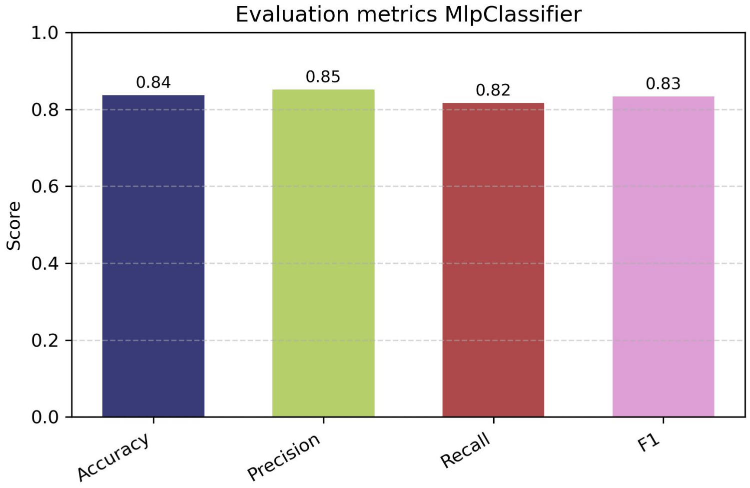
Data Cleanup & Modeling Strategy

Modeling Roadmap

1. **Start with non-linear traditional models**—Decision Tree, Random Forest, XGBoost, etc.—because they:
   - Naturally capture complex, nonlinear interactions
   - Provide built-in measures of feature importance to help me identify the most predictive inputs
2. **Apply the same workflow to the other subsets**, even those with many nulls (after appropriate cleaning), to compare how different feature groups perform.
3. **Refine feature selection**: once I know which variables matter most in each subset, I will:
   - Build a reduced “meta-dataset” that merges only the top features across subsets
   - Retrain and compare models on that distilled dataset
4. **Explore Deep Learning**: finally, with a lean set of highly relevant inputs, I’ll experiment with feed-forward neural networks or other non-sequential deep models to see if they can further improve performance.

Deep learning vs Machine learning





Progress on MLPClassifier Optimization (Technical Analysis Subset)

An optimized **MLPClassifier** was implemented and trained on the *Technical Analysis* subset. The model achieved solid results: **Accuracy = 0.84**, **Precision = 0.85**, **Recall = 0.82**, and **F1-score = 0.83**, as shown in the figure. These scores are comparable to the **XGBoostClassifier**, which reached **Accuracy = 0.84**, **Precision = 0.83**, **Recall = 0.87**, and **F1-score = 0.85**.

While MLP didn't outperform in every metric, it shows competitive performance. The next step is to create an improved dataset using the **top features selected by XGBoost**, then retrain the MLP to evaluate potential improvements. Additionally, **cross-validation** will be applied to further optimize the model and assess its generalization.

Technical Analysis subset