MARKET MANIPULATION DETECTION USING SUPERVISED LEARNING

Project Supervisor Dr. Liu Qi, Assistant Professor of Computer Science
Second Examiner Dr. Zou Difan, Assistant Professor of Computer Science

Group Members
Ho Megan Qian Hua (3035832749)
Li Wo Him (3035783986)
Tsang Hoi Wei (3035785879)

18 April 2024

Table of Contents

01 BACKGROUND & MOTIVATION

Spectrum of Market Manipulation is Diversified

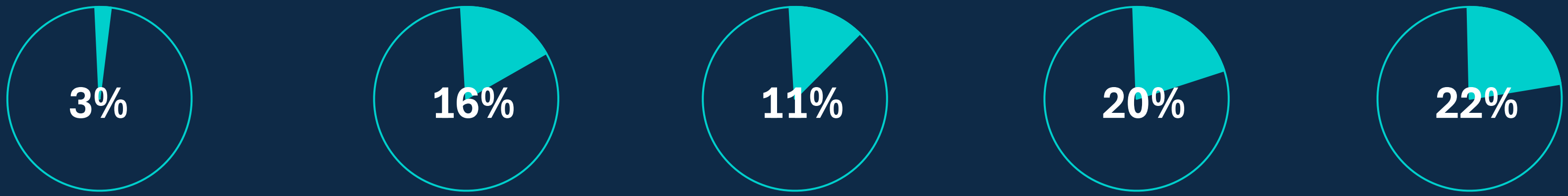
Marking the Close

Spoofing / Layering

Insider Dealing

Misleading Market Information

Other Surveillance



% of number of **Global Incidents**

There is still a Detection Gap in Financial Fraud

≈ ⅔ of fraud is **NOT** identified

≈ ⅓ of fraud is **identified**

Impact in 2021:

- **≈10%** of large publicly traded companies engaged in fraud
- **≈$830 billion** in losses

Literature Review

The early researchers (Allen and Gale, 1992) conducted pioneering studies on stock-price manipulation.

Action-based

Information-based

Trade-based

Literature Review

Just a theoretical framework.

The early researchers (Allen and Gale, 1992) conducted pioneering studies on stock-price manipulation.

Action-based

Information-based

Trade-based

Literature Review

Action-based | Information-based | **Trade-based**

In 2017, a group of researchers have made further investigations on the problem statement and transformed theoretical perspectives into practices.

“Daily and tick real time trading stock data in evaluate those supervised machine”

(Aihua, Jiede, Zhidong, 2017)

Literature **Review**

Action-based

Information-based

Trade-based

In 2017, a group of researchers have made further investigations on the problem statement and transformed theoretical perspectives into practices.

Literature Review

In 2017, a group of researchers have made further investigations on the problem statement and transformed theoretical perspectives into practices.

Weak EMH form claims that past price and volume information cannot be used to predict future movements.

Literature Review

Best solution: Looked into more financial indicators rather than price ticks

Researchers mentioned the consideration of factors such as:

Literature **Review**

Best solution: Looked into more financial indicators rather than price ticks

Researchers mentioned the consideration of factors such as:

"Size of company, ratios, liquidity of stock, status of information clarity, and structure of shareholders "

(Fallh and Kordlouie, 2011)

02 METHODOLOGY

Methodology Overview

Model Building

- Support Vector Machines
- Decision Trees
- Naïve Bayes
- Logistic Regression

Historical Data Analysis

- Analyse **companies** involved in **stock manipulation**

Threshold Detection

- Flagging **suspicious activities**

Webpage Development

- Display **empirical results**

There are **6 STEPS** in this part.

How do we collect data ?

Step 1/6: The **labelled data** (with or without market manipulation) is obtained from https://global-csmar-com.eproxy.lib.hku.hk/ CSRC's Enforcement Actions.

Timeframe

The latest amendment of security law for Shanghai Stock Exchange and Shenzhen Stock Exchange happened in 2019.

Raw Data

Total of 2781 Samples
1508 Negative Samples & 1273 Positive Samples

Eyeball Observation

It **seems like a balanced dataset** upon initial data collection.

How do we collect data ?

Step 1/6: The **labelled data** (with or without market manipulation) is obtained from https://global-csmar-com.eproxy.lib.hku.hk/ CSRC's Enforcement Actions.

Stock	Violation Date	Violation Type	[Other Irrelevant Fields]	Market Manipulation?
0001	03 May 2020, 12 July 2020	A	…	YES
0002	20 October 2021	A	…	NO
0003	19 January 2020	B	…	YES
0004	02 May 2022, 9 May 2022	B	…	YES
…	…	…	…	…

How do we process data ?

Violation **may or may not** be market manipulation.

Stock	Violation Date	Violation Type	[Other Irrelevant Fields]	Market Manipulation?
0001	03 May 2020, 12 July 2020	A	…	YES
0002	20 October 2021	A	…	NO
0003	19 January 2020	B	…	YES
0004	02 May 2022, 9 May 2022	B	…	YES
…	…	…	…	…

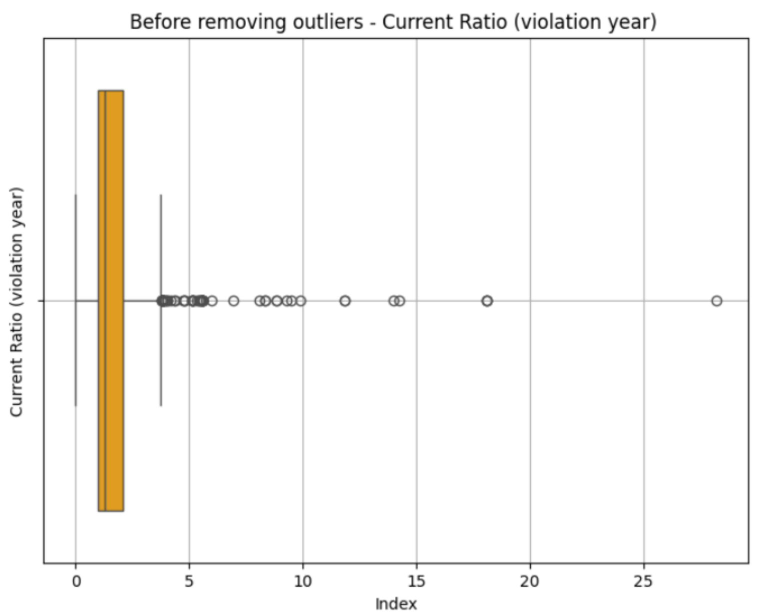
Exploratory Data Analysis (EDA)

Step 2/6: Examining & summarizing data to gain, identify patterns, detect anomalies.

Basic Summary Statistics

Population
Mean
S.D.
Range
Skewness
Kurtosis

Count	651
Mean	1.898587
Std	2.213939
Min	0.0
25%	0.955
50%	1.3
75%	2.07
Max	28.2



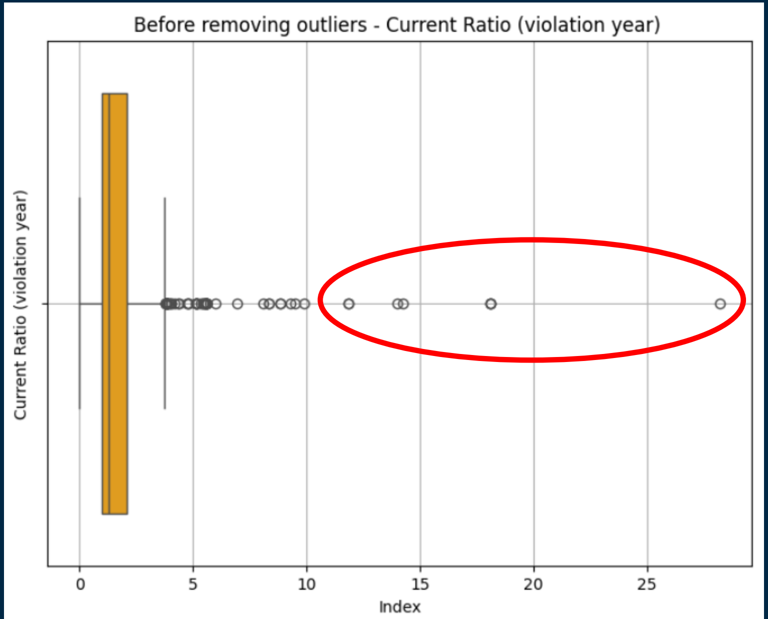
Exploratory Data Analysis (EDA)

Step 2/6: Examining & summarizing data to gain, identify patterns, detect anomalies.

Basic Summary Statistics

Population
Mean
S.D.
Range
Skewness
Kurtosis

Count	651
Mean	1.898587
Std	2.213939
Min	0.0
25%	0.955
50%	1.3
75%	2.07
Max	28.2



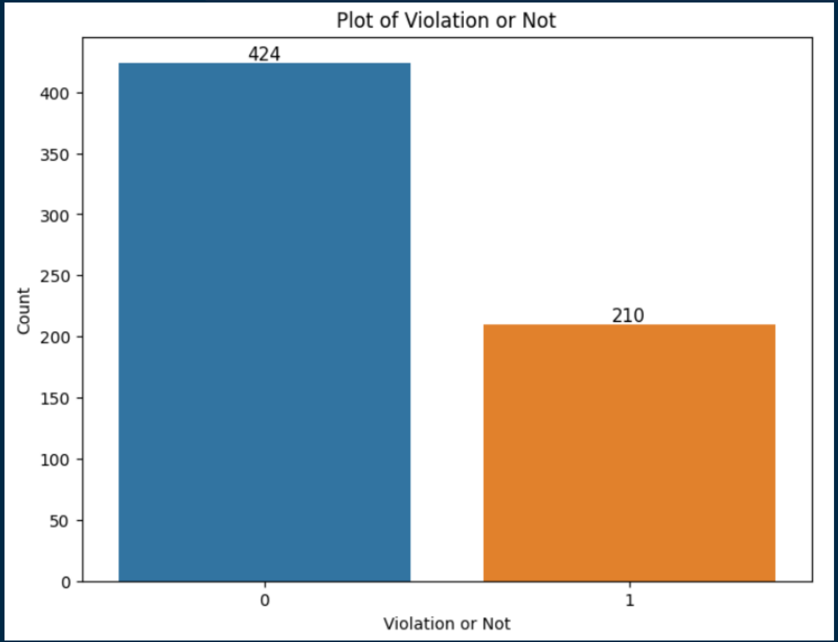
Exploratory Data Analysis (EDA)

Step 2/6: Examining & summarizing identify NaN values, detect anomalies.

The bar chart shows number of positive and negative samples.

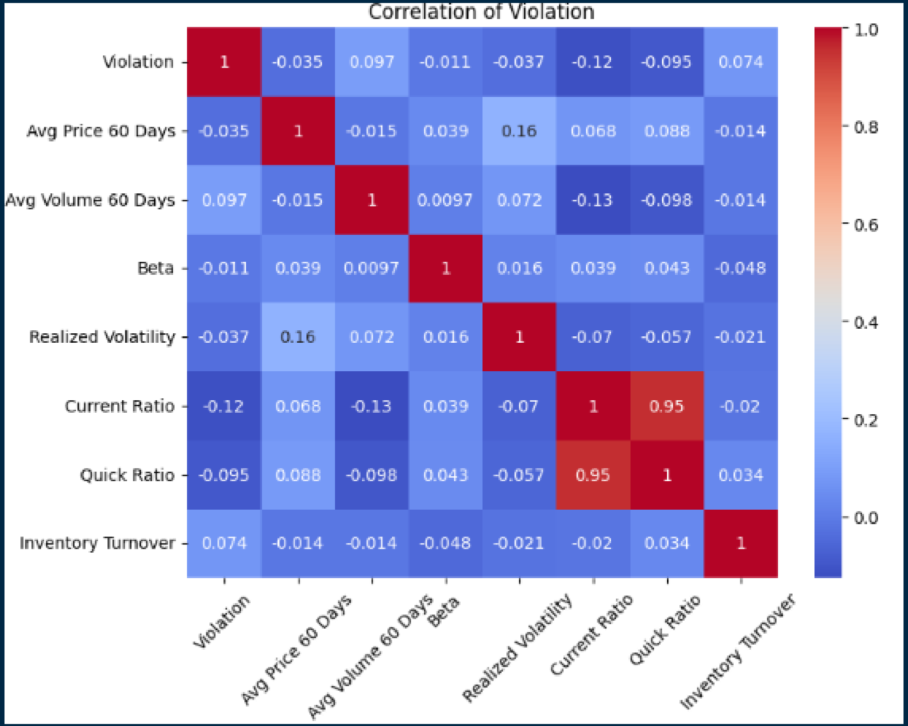
Check whether additional steps is necessary for **imbalanced dataset**.

Before removing outliers – 651 samples
** **After removing outliers** – 634 samples



Exploratory Data Analysis (EDA)

Step 2/6: Examining & summarizing data to gain, identify patterns, detect anomalies.



According to the heatmap, there are **weak correlations** among all features.

It may favour the model training of Naïve Bayes Model.

How do we process data ?

Step 3/6: **Removing and adjusting data** according to the condition below

Removing

Delisted Firms

60 days back-and-fourth no trade

Some of the case categories are not specific

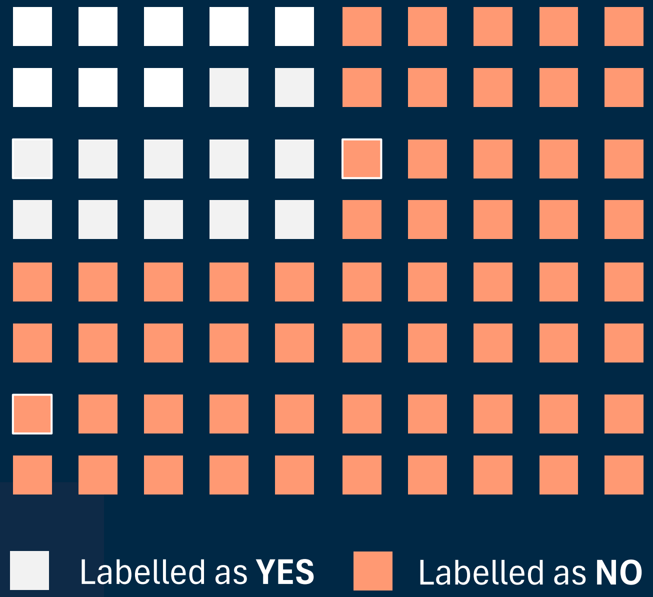
Adjusting

If the incidents are logged during weekend/public holiday → take the closest trading day

If there is no data in a concerned period → take the average of the start and the end date

How do we **process data** ?

2020 – 2022 Data



Stock	Violation Date	Violation Type	Market Manipulation?
…	…	…	YES / NO

Step 4/6: Choose cases from 2020, 2021, and 2022 (focused on case P2512 – Illegal Stock Trading), **651 cases** in total

Step 5/6: Collect **relevant data for each record.**

2020 – 2022 Data

2020

Stock	Violation Date	Violation Type	Manipulation
0001	03 May 2020	A	YES

1. **Average price change 60 days before and after the manipulation date**
2. **Average volume change 60 days before and after the manipulation date**

VWAP (Volume-Weighted Average Price) Analysis

Step 5/6: Collect **relevant data for each record.**

2020 – 2022 Data

2020

Stock	Violation Date	Violation Type	Manipulation
0001	03 May 2020	A	YES

1. Average price change 60 days before and after the manipulation date

2. Average volume change 60 days before and after the manipulation date

3. Inventory turnover

$$\textit{Inventory turnover} = \frac{\textit{Cost of Goods Sold}}{\textit{Average Price of Inventory}}$$

Indicates liquidity

- Higher ratio = More efficiently manage inventory
- Better performance -> Affect investment decision

Step 5/6: Collect relevant data for each record.

2020 – 2022 Data

2020

Stock	Violation Date	Violation Type	Manipulation
0001	03 May 2020	A	YES

1. Average price change 60 days before and after the manipulation date

2. Average volume change 60 days before and after the manipulation date

3. Inventory turnover

4. Beta

$$\beta_i = \frac{Covariance(r_i, r_m)}{Variance(r_m)}$$

$\beta_i = market\ beta\ of\ asset\ i$
$r_i = expected\ return\ on\ an\ asset\ i$
$r_m = average\ expected\ rate\ of\ return\ on\ the\ market$

Step 5/6: Collect **relevant data for each record.**

2020 – 2022 Data

2020

Stock	Violation Date	Violation Type	Manipulation
0001	03 May 2020	A	YES

1. **Average price change 60 days before and after the manipulation date**

2. **Average volume change 60 days before and after the manipulation date**

3. Inventory turnover

4. Beta

5. Realized Volatility

- Measure by standard deviation on logarithmic return
- Higher volatility = Higher risk and uncertainty
- More susceptible to market manipulation

Step 5/6: Collect relevant data for each record.

2020 – 2022 Data

2020

Stock	Violation Date	Violation Type	Manipulation
0001	03 May 2020	A	YES

1. Average price change 60 days before and after the manipulation date
2. Average volume change 60 days before and after the manipulation date
3. Inventory turnover
4. Beta
5. Realized Volatility
6. Current Ratio
7. Quick Ratio

$$Current\ Ratio = \frac{Current\ Asset}{Current\ Liability}$$

$$Quick\ Raio = \frac{Cash\ \&\ Equivalents}{Current\ Liability}$$

How do we **train the model** ?

Step 6/6: **Run each record once** and **train learning models in pipeline**

Stock	Violation Date	Violation Type	Manipulation
0001	03 May 2017, 12 July 2017	A	YES
0002	20 October 2019	A	NO
...			
0030	05 March 2012	E	YES

- Decision Trees
- Naïve Bayes
- Support Vector Machine
- Logistic Regression

How do we train the model ?

Step 6/6: Predict whether it involves market manipulation or not

Output

1: Stock Manipulation

0: No Stock Manipulation

- Decision Trees
- Naïve Bayes
- Support Vector Machine
- Logistic Regression

03 RESULTS & DISCUSSION

Building Machine Learning Models

Common Settings

- Current training size : current testing size = 3 : 1
- Use of Standard Scaler
- K-Fold Cross Validation, k = 10

Select 4 Models

1. Decision Trees
2. Naïve Bayes
3. Support Vector Machines
4. Logistic Regression

Evaluation

- By accuracy
- By sensitivity, specificity
- By ROC curve and AUC

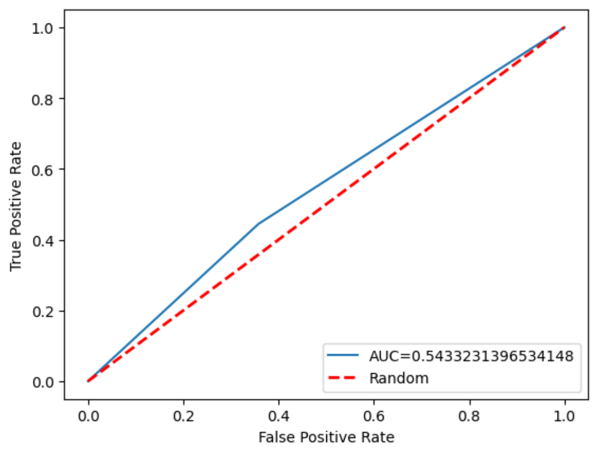
Model 1/4 – Decision Tree

Parameters

- Criterion – determines the quality of a split
- Max_depth – Maximum depth of the tree
- Min_samples_split – Minimum number of samples required to split an internal node
- Min_samples_leaf – Minimum number of samples required to be at a leaf node

Initial Parameters Setting

- Use Default Values
  - → A rough idea for performance
  - → Finding a baseline



Baseline Performance

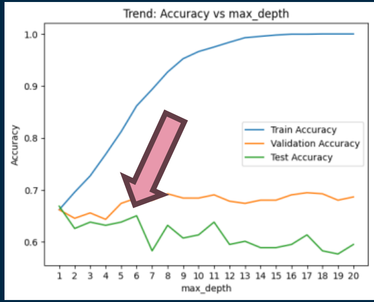
Accuracy		
Train	Val	Test
1.0	0.69	0.58

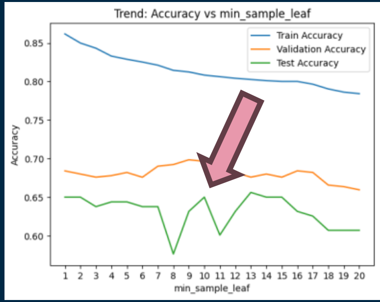
Sensitivity	Specificity	AUC
0.44	0.64	0.54

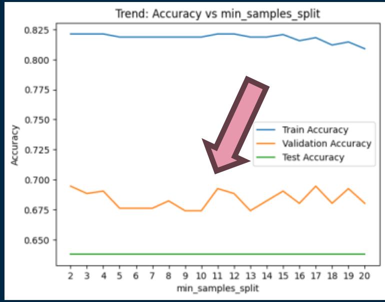
Model 1A/4 – Single Decision Tree

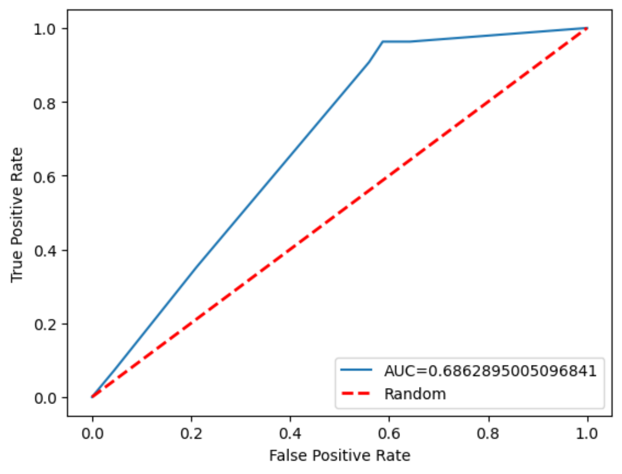
Accuracy
0.64

AUC
0.68







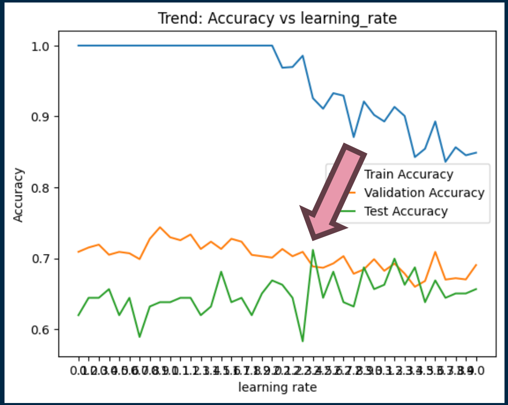


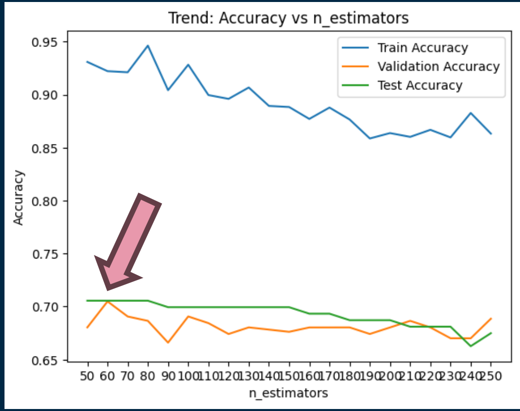
- **Fair performance** in terms of accuracy and AUC.
- Test if performance can be further improved by implementing **boosting** and **random forest classifiers**.

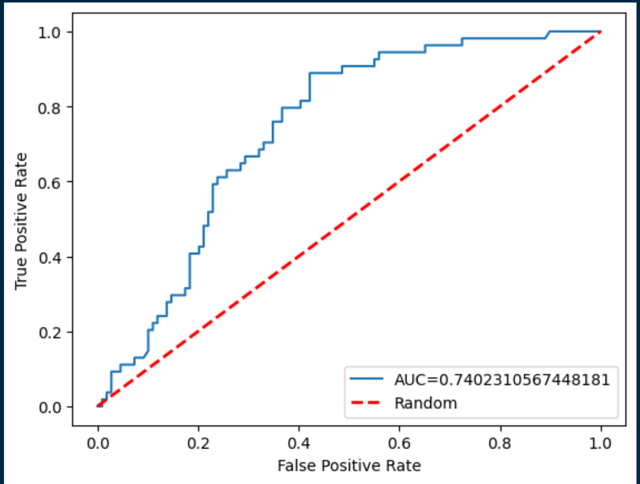
Model 1B/4 – Adaboost Classifier

Accuracy
0.71

AUC
0.74





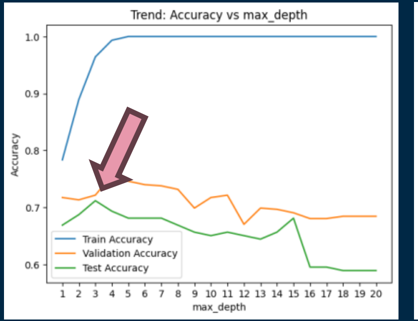


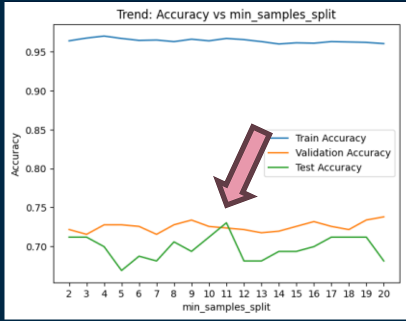
- **High accuracy and AUC,** showing that the model can classify majority of data correctly.
- **0.61 sensitivity and 0.75 specificity**, showing that the model's true positive and negative rate is close and balanced.

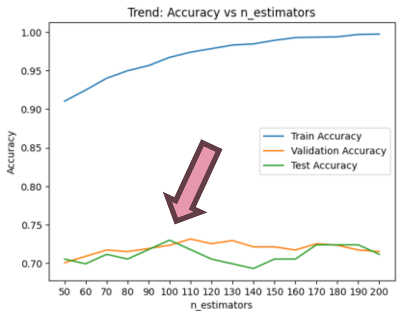
Model 1C/4 – Gradient Boosting Classifier

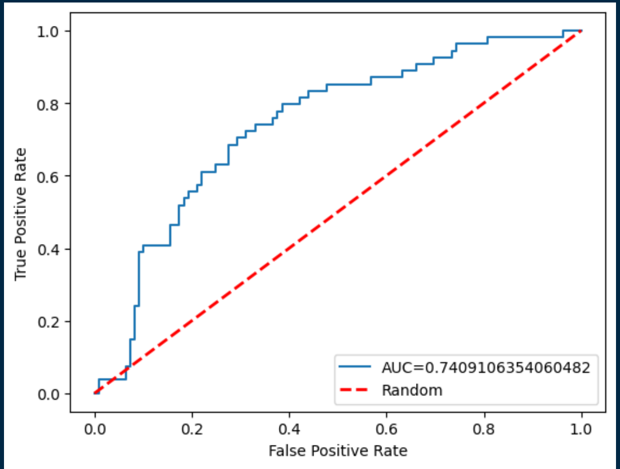
Accuracy
0.71

AUC
0.74







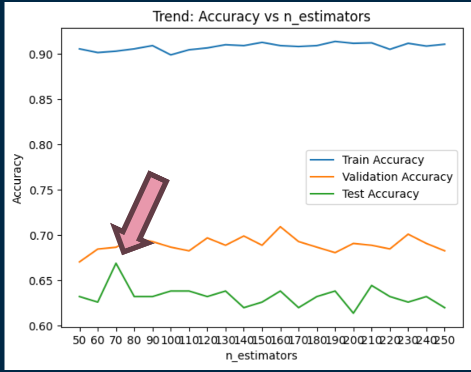


- **High accuracy and AUC,** showing that the model can classify majority of data correctly.
- **0.5 sensitivity and 0.84 specificity**, showing that the model has room of improvement in detecting true positive.

Model 1D/4 – Random Forest Classifier

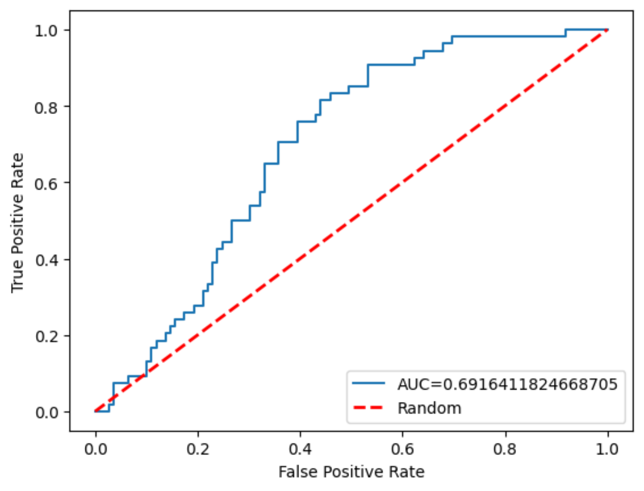
Accuracy
0.63

AUC
0.69



Accuracy			AUC
Train	Val	Test	
0.91	0.68	0.63	0.69

Sensitivity	Specificity	F1 score
0.33	0.79	0.38



- Lack of accuracy might be due to the insufficient data and features.
- Sensitivity is low, more likely to miss identifying the positive samples when it is present.
- Low F1-score shows the model has a high false positive or negative rate.

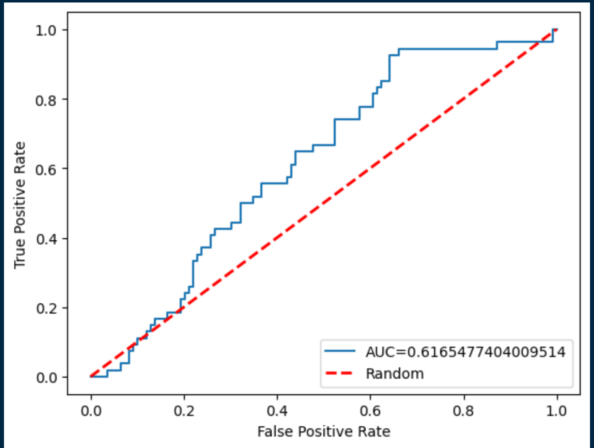
Model 2/4 – Naïve Bayes

Hyperparameters

- Priors – Prior probability assigned to different classes
- Smoothing parameter – handling the issue of zero probability when measuring variance

Initial Parameters Setting

- Use Default Values
  - → A rough idea for performance
  - → Finding a baseline



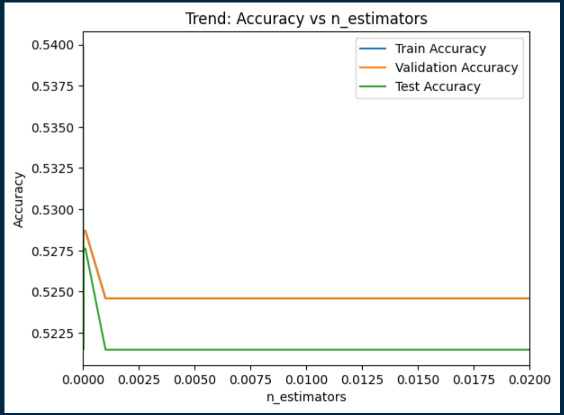
Baseline Performance

Accuracy		
Train	**Val**	**Test**
0.53	0.53	0.54

Sensitivity	Specificity	AUC
0.94	0.34	0.62

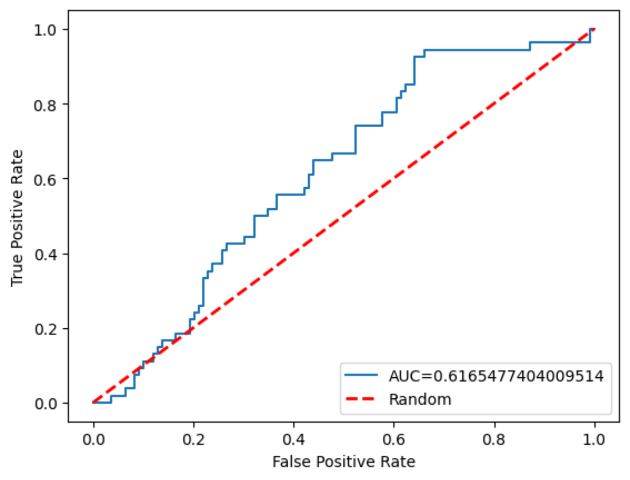
Model 2/4 – Gaussian Naïve Bayes

Accuracy	AUC
0.54	0.62



Accuracy		
Train	Val	Test
0.53	0.53	0.54

Sensitivity	Specificity	AUC
0.94	0.34	0.62



- **Priors** and **smoothing parameter** have no significant effect on the model's performance.
- Given there's only **weak correlation** between features, it is possibly caused by **insufficient data**, or irrelevant features.

Model 3/4 – Support Vector Machine (SVM)

Hyperparameters

- Kernel type – determines the linearity of the relationships
- Regularization (C) – control by maximizing margin & minimizing classification error
- Gamma – determines the influence of individual samples on decision boundary

```python
# Define the number of folds (k)
num_folds = 10

# Create a k-fold cross-validation splitter
kf = KFold(n_splits=num_folds, shuffle=True, random_state=42)

# Create an SVM Randomised
svm = SVC()

parameters = {'kernel': ['rbf', 'sigmoid', 'poly'],
              'C': [0.1, 1.0, 10.0, 100.0, 1000.0, 10000.0, 100000.0],
              'gamma': [0.00001, 0.0001, 0.001, 0.01, 0.1, 1.0],
              }

# Train the SVM model
clf = RandomizedSearchCV(estimator=svm, param_distributions = parameters, cv = kf, random_state = 42)
clf.fit(X_train_SMOTE_scaled, y_train_SMOTE)

#summarize results
print("Best: %f using %s" % (clf.best_score_, clf.best_params_))
```

Initial Parameters Setting

- Use RandomizedSearchCV
  - → Computational power
  - → Better coverage of hyperparameter space

Model 3A/4 – Support Vector Machine (SVM)

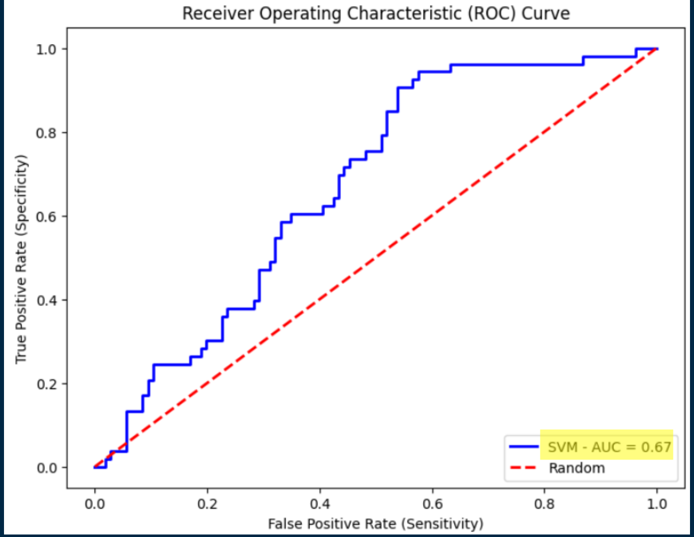
Accuracy
0.62

AUC
0.67

```
Fold 1: Train Accuracy: 0.5667, Validation Accuracy: 0.5417
Fold 2: Train Accuracy: 0.6042, Validation Accuracy: 0.6250
Fold 3: Train Accuracy: 0.6487, Validation Accuracy: 0.6667
Fold 4: Train Accuracy: 0.5878, Validation Accuracy: 0.6875
Fold 5: Train Accuracy: 0.5761, Validation Accuracy: 0.6042
Fold 6: Train Accuracy: 0.5888, Validation Accuracy: 0.6383
Fold 7: Train Accuracy: 0.5794, Validation Accuracy: 0.4681
Fold 8: Train Accuracy: 0.5724, Validation Accuracy: 0.5532
Fold 9: Train Accuracy: 0.6308, Validation Accuracy: 0.7660
Fold 10: Train Accuracy: 0.6519, Validation Accuracy: 0.5957
Average train accuracy: 0.6006971043358358
Average validation accuracy: 0.6146276595744681
Test set accuracy: 0.6163522012578616
```

```
SVC(C=10, gamma=0.1, kernel='sigmoid')
Sensitivity: 0.43
Specificity: 0.71
Accuracy: 0.62
F-score: 0.43
```

	precision	recall	f1-score	support
0	0.71	0.71	0.71	106
1	0.43	0.43	0.43	53
accuracy			0.62	159
macro avg	0.57	0.57	0.57	159
weighted avg	0.62	0.62	0.62	159



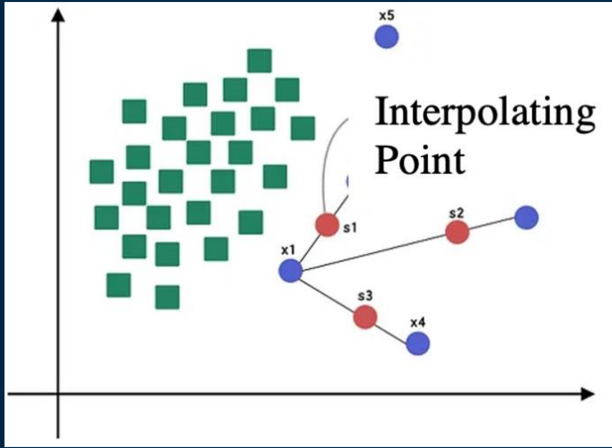
- Lack of accuracy might be due to the imbalanced datasets.
- Sensitivity is low, more likely to miss identifying the positive samples when it is present.
- F1-score shows the model is weak in detecting the class 1 (positive samples).

Over-sampling methods to address class imbalance

Compare TWO over-sampling methods with SVM:

1. **SMOTE**
   - Generate synthetic samples for the minority class by interpolating between existing minority class instances

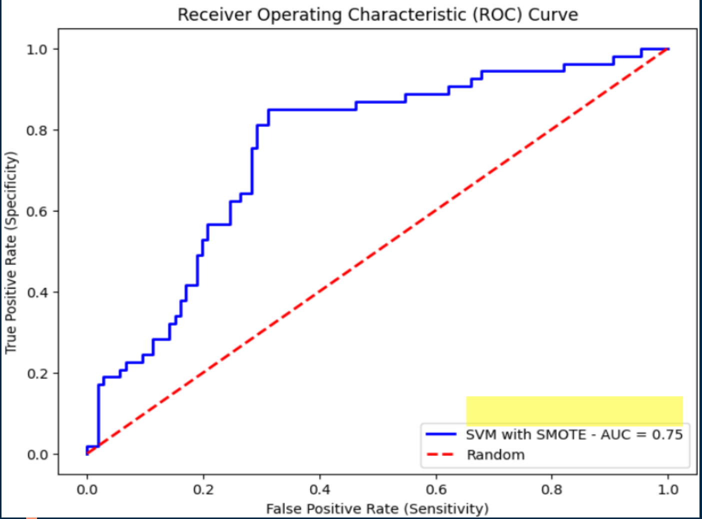
2. **SVM SMOTE**
   - Tailored to SVM
   - Focused on increasing minority points along the decision boundary



Model 3B/4 – SVM with SMOTE

Accuracy
0.72

AUC
0.75




```
Fold 1: Train Accuracy: 0.8566, Validation Accuracy: 0.7656
Fold 2: Train Accuracy: 0.8619, Validation Accuracy: 0.6562
Fold 3: Train Accuracy: 0.8671, Validation Accuracy: 0.7031
Fold 4: Train Accuracy: 0.8636, Validation Accuracy: 0.6250
Fold 5: Train Accuracy: 0.8654, Validation Accuracy: 0.6719
Fold 6: Train Accuracy: 0.8811, Validation Accuracy: 0.5625
Fold 7: Train Accuracy: 0.8621, Validation Accuracy: 0.6825
Fold 8: Train Accuracy: 0.8726, Validation Accuracy: 0.7143
Fold 9: Train Accuracy: 0.8621, Validation Accuracy: 0.7460
Fold 10: Train Accuracy: 0.8604, Validation Accuracy: 0.6508
Average train accuracy: 0.865304677870123
Average validation accuracy: 0.6778025793650795
Test set accuracy: 0.7169811320754716
```

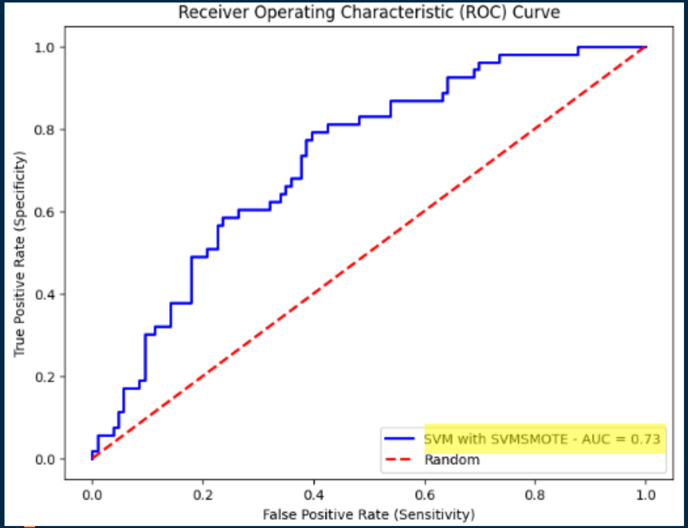
```
SVC(C=0.8, gamma=1.0)
Sensitivity: 0.72
Specificity: 0.72
Accuracy: 0.72
F-score: 0.63
              precision    recall  f1-score   support

           0       0.84      0.72      0.77       106
           1       0.56      0.72      0.63        53

    accuracy                           0.72       159
   macro avg       0.70      0.72      0.70       159
weighted avg       0.74      0.72      0.72       159
```

- Demonstrate a slight overfitting with higher testing accuracy.
- Higher AUC, the model has better performance in classifying positive and negative classes.
- Consistent sensitivity and specificity.

Model 3C/4 – SVM with SVMSMOTE

Accuracy
0.70

AUC
0.73




```
Fold 1: Train Accuracy: 0.6731, Validation Accuracy: 0.6406
Fold 2: Train Accuracy: 0.6976, Validation Accuracy: 0.5938
Fold 3: Train Accuracy: 0.7185, Validation Accuracy: 0.6094
Fold 4: Train Accuracy: 0.6626, Validation Accuracy: 0.5938
Fold 5: Train Accuracy: 0.7343, Validation Accuracy: 0.6406
Fold 6: Train Accuracy: 0.7045, Validation Accuracy: 0.5469
Fold 7: Train Accuracy: 0.6911, Validation Accuracy: 0.6349
Fold 8: Train Accuracy: 0.7243, Validation Accuracy: 0.6190
Fold 9: Train Accuracy: 0.7016, Validation Accuracy: 0.7302
Fold 10: Train Accuracy: 0.7173, Validation Accuracy: 0.7460
Average train accuracy: 0.7024765374241813
Average validation accuracy: 0.635515873015873
Test set accuracy: 0.6981132075471698
```

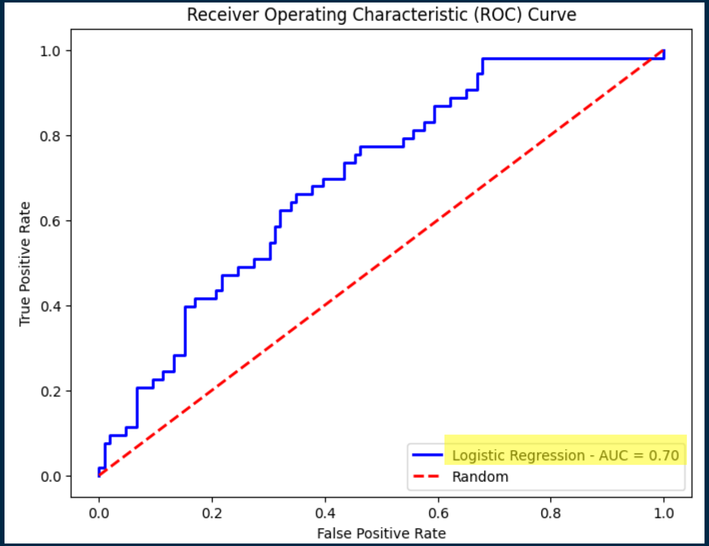
```
SVC(C=0.15, gamma=1)
Sensitivity: 0.51
Specificity: 0.79
Accuracy: 0.70
F-score: 0.53
```

	precision	recall	f1-score	support
0	0.76	0.79	0.78	106
1	0.55	0.51	0.53	53
accuracy			0.70	159
macro avg	0.66	0.65	0.65	159
weighted avg	0.69	0.70	0.69	159

- **Training and Testing accuracy are consistent .**
- **Still lack of performance in identifying the positive samples after performing SVM SMOTE**

Model 4A/4 – Logistic Regression

Accuracy	AUC
0.67	0.70




```
Fold 1: Train Accuracy: 0.6721, Validation Accuracy: 0.6875
Fold 2: Train Accuracy: 0.6674, Validation Accuracy: 0.7292
Fold 3: Train Accuracy: 0.6674, Validation Accuracy: 0.6875
Fold 4: Train Accuracy: 0.6745, Validation Accuracy: 0.6667
Fold 5: Train Accuracy: 0.6745, Validation Accuracy: 0.6458
Fold 6: Train Accuracy: 0.6799, Validation Accuracy: 0.5957
Fold 7: Train Accuracy: 0.6706, Validation Accuracy: 0.6809
Fold 8: Train Accuracy: 0.6799, Validation Accuracy: 0.6170
Fold 9: Train Accuracy: 0.6589, Validation Accuracy: 0.7660
Fold 10: Train Accuracy: 0.6869, Validation Accuracy: 0.5745
Average train accuracy: 0.6732140121254568
Average validation accuracy: 0.6650709219858155
Test set accuracy: 0.6666666666666666
```

```
LogisticRegression(C=0.07, max_iter=250, penalty='l1', solver='liblinear')
Sensitivity: 0.02
Specificity: 0.99
Accuracy: 0.67
F-score: 0.04
```

	precision	recall	f1-score	support
0	0.67	0.99	0.80	106
1	0.50	0.02	0.04	53
accuracy			0.67	159
macro avg	0.58	0.50	0.42	159
weighted avg	0.61	0.67	0.54	159

- **Sensitivity is 0.02 (correctly identify 2% of actual positive samples).**
- **Specificity is 0.99 (correctly identify 99% of the actual negative samples).**
- **Very low F-score, poor in precision and recall.**

Model 4B/4 – Logistic Regression with SMOTE

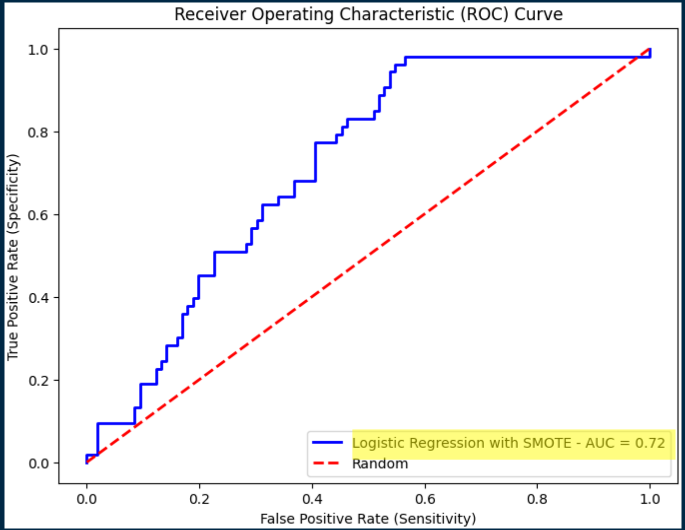
Accuracy
0.53

AUC
0.72

```
Fold 1: Train Accuracy: 0.6311, Validation Accuracy: 0.6719
Fold 2: Train Accuracy: 0.6329, Validation Accuracy: 0.6562
Fold 3: Train Accuracy: 0.6416, Validation Accuracy: 0.5781
Fold 4: Train Accuracy: 0.6294, Validation Accuracy: 0.6875
Fold 5: Train Accuracy: 0.6364, Validation Accuracy: 0.6250
Fold 6: Train Accuracy: 0.6364, Validation Accuracy: 0.6250
Fold 7: Train Accuracy: 0.6353, Validation Accuracy: 0.6349
Fold 8: Train Accuracy: 0.6405, Validation Accuracy: 0.5873
Fold 9: Train Accuracy: 0.6318, Validation Accuracy: 0.6667
Fold 10: Train Accuracy: 0.6370, Validation Accuracy: 0.6190
Average train accuracy: 0.6352194925493355
Average validation accuracy: 0.6351686507936508
Test set accuracy: 0.5345911949685535
```

```
LogisticRegression(C=0.08, max_iter=450, penalty='l1', solver='liblinear')
Sensitivity: 0.98
Specificity: 0.31
Accuracy: 0.53
F-score: 0.58
```

	precision	recall	f1-score	support
0	0.97	0.31	0.47	106
1	0.42	0.98	0.58	53
accuracy			0.53	159
macro avg	0.69	0.65	0.53	159
weighted avg	0.79	0.53	0.51	159



- **Higher AUC but lower accuracy.**
- **Oversampling with SMOTE might introduce noise causing the model to have lower overall performance.**

Summary – Model Performance

Model	Accuracy	Sensitivity	Specificity	F-score	AUC
Support Vector Machine (SVM)	0.62	0.43	0.71	0.43	0.67
SVM with SMOTE	**0.73**	**0.72**	**0.72**	**0.63**	**0.75**
SVM with SVMSMOTE	0.70	0.51	0.79	0.53	0.73
Single Decision Tree	0.64	0.35	0.79	0.40	0.68
Decision Tree (Gradient Boosting)	**0.71**	**0.56**	**0.80**	**0.57**	**0.74**
Decision Tree (Random Forest)	0.63	0.33	0.79	0.38	0.69
Gaussian Naïve Bayes	0.54	0.94	0.34	0.58	0.62
Logistic Regression	0.67	0.02	0.99	0.04	0.70
Logistic Regression with SMOTE	0.53	0.98	0.31	0.58	0.72

SVM with SMOTE is the best model

Model	Accuracy	Sensitivity	Specificity	F-score	AUC
SVM with SMOTE	0.73	0.72	0.72	0.63	0.75

Overall Performance

- The SVM with SMOTE outperforms the rest of the models with the highest accuracy and AUC.
- Balanced sensitivity and specificity in determining the true positives and true negatives.
- The high AUC demonstrated that the model is suitable for binary classification, which aligns with our dataset of detecting violated cases and non-violated cases.

04 CHALLENGES & MITIGATION PLANS

Project Challenges and Mitigation Plans

	Challenges	Description	Mitigation Plans
01	Data Collection	Financial ratio collection for each unique companies (e.g.: Average price change 60 days before and after the manipulation date)	Get data from Yahoo Finance API using Python
02	Data Quality	Historical data inconsistency and incompleteness due to data access limitations	Perform preprocessing steps like data cleaning & standardization
03	Imbalanced Dataset	Class imbalanced can lead to biased models that favor majority classes	Apply SMOTE techniques and continuous tuning for best results
04	Model Generalization	Model might not generalize well to new and unseen data	Apply technique like cross-validation and regularization

05 FUTURE PLANS & CONCLUSION

We have **3 FUTURE PLANS** in this part.

PLAN 1/3 – Sourcing Social Media

Some types of market manipulations are largely contributed by retail investors.

Analyze the news and events with Natural Language Processing (NLP) algorithms.

Add to features for analysis as numerical features

Post 1
Website: Guba
Text: **“It has announced…”**
Replies: **#increase** #technology #chips #AI #shanghai
Read Count: **92K**

Web Crawling

Web Crawling

Post 2
Website: Guba
Text: **“Put in some dry powder!”**
Replies : **#increase** #qualcomm #NASDAQ
Read Count: **1.6K**

Post 3
Website: Guba
Text: **“good outlook ☹…”**
Replies: **#increase** #bearish #FOMO #strke
Read Count: **900K**

PLAN 2/3 - Separate Model Training

Separate the table according to violation type and train each model to each violation type to **reduce the bias brought by the treatment effect.**

Violation Type ID	Violation Type Description
P2501	Fictitious Profit
P2502	Fictitious Assets
P2503	False Recordation (Misleading Statements)
P2504	Delayed Disclosure
P2505	Material Omission
P2506	Other False Information Disclosure
P2507	Fraudulent Listing
P2511	Insider Trading
P2512	Illegal Stock Trading
P2513	Stock Price Manipulation

PLAN 2/3 - Separate Model Training

Stock	Violation Date	Violation Type	Manipulation
0001	03 May 2017, 12 July 2017	P2501	YES
0002	20 October 2019	P2501	NO
…	…	…	…

Learning Model P2501

Stock	Violation Date	Violation Type	Manipulation
0003	19 January 2020	P2502	YES
0004	02 May 2014, 9 May 2014	P2502	YES
…	…	…	…

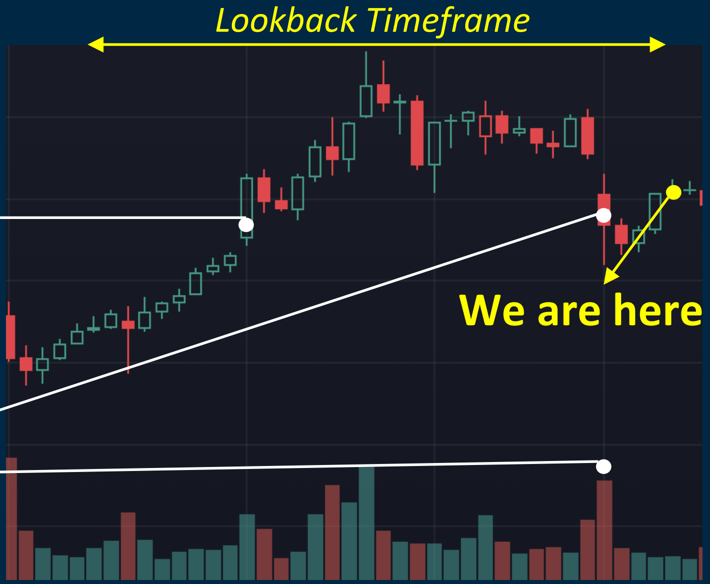
Learning Model P2502

PLAN 3/3 - Anomalies Real-time Detection

By the help from the trained model , we will soon be able to provide real-time analytics. They **detect strange / abnormal deviation of the parameters at a particular time and inform the related parties**.

1. Higher-than-average volatility activates our models

2. Models will also flag the featured trading time slots



3. Dynamically adjusted probabilities

Model	Probabilities
A	78.89%
B	64.30%
C	23.40%
D	2.16%

Motivation **Diversified** Manipulation Source + Detection **Gap** + High **Accuracy** from ML

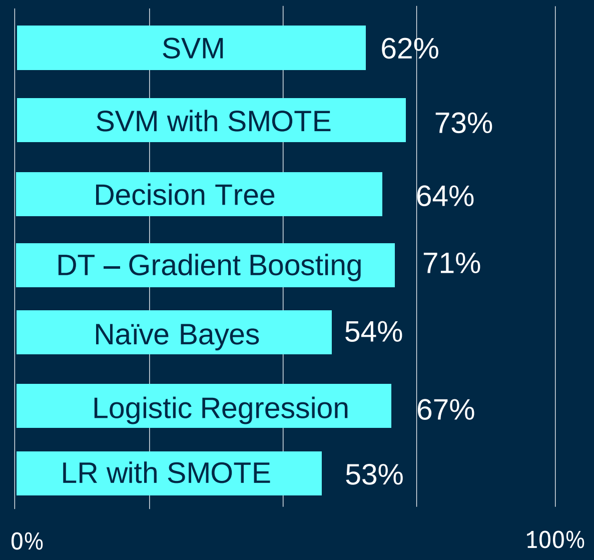
Data are...

Collected
Cleaned
Processed
Balanced
Summarized

Best Model is

Support Vector Machine (SVM) with regular SMOTE

Accuracy



Preferred Model Selection

Model that provides the **best consistent** results across **accuracy, F-score and AUC**

THANK YOU

References

- Everton Gomede, P. (2023, July 30). *Synthetic minority over-sampling technique (SMOTE): Empowering AI through Imbalanced Data handling*. Medium. https://medium.com/@evertongomede/synthetic-minority-over-sampling-technique-smote-empowering-ai-through-imbalanced-data-handling-d86f4de32ea3

- Golmohammadi, K., Zaiane, O. R., & Diaz, D. (2014). Detecting stock market manipulation using supervised learning algorithms. *2014 (DSAA)*. https://doi.org/10.1109/dsaa.2014.7058109

- Liu et al. (2021). *Detecting stock market manipulation via machine learning: Evidence from China Securities Regulatory Commission punishment cases.* doi:https://doi.org/10.1016/j.irfa.2021.101887

- Tsipursky, Dr. G. (2023, April 12). *The hidden epidemic of corporate fraud*. Forbes. https://www.forbes.com/sites/glebtsipursky/2023/04/11/the-hidden-epidemic-of-corporate-fraud/?sh=6c7410a16787

- Tramplin, T. (2023). Market Manipulation. Retrieved from Finance Strategists: https://www.financestrategists.com/financial-advisor/business-ethics/market-manipulation/

Appendix 1

Types and values of a particular company listed in CSRC's Enforcement of Action

Column	Description	Attribute
Date	Trading date	Ordinal
Open	Daily open price	Numeric
High	Daily highest price	Numeric
Low	Daily lowest price	Numeric
Close	Daily close price	Numeric
Adj Close	Daily adjusted closing price	Numeric
Volume	Daily Trading Volume	Numeric

Appendix 2

Descriptions for common types of market manipulations (Tramplin, 2023)

Type	Description
Pump and Dump	Artificially boost the price of a security by disseminating false or deceptive information
Spoofing	Make fake orders in the market without execution to create a false image
Wash Trading	Buying and selling same securities at one time and create an illusion of increased trading volume
Insider Trading	Individuals access to non-disclosure trading information, leaving unfair advantage to other investors
Cornering the Market	Dominant in a security, commodity, or any financial instrument to manipulate and control the price and supply
Front-Running	Exploiting advanced knowledge of impending orders or trades and earning from price fluctuations

Appendix 3

Descriptions for underfitting and overfitting in machine learning

Underfitting	**Characteristics**	**Overfitting**
Model is not complex	**Model**	Model is too complex
Not Accurate	**Training Dataset**	Accurate
Not Accurate	**Testing Dataset**	Not Accurate
Increase number of features Increase training duration Increase model complexity Remove noise from data	**Reduction Techniques**	Reduce number of features Introduce early stopping Reduce model complexity Increase training data

Appendix 4 - SMOTE-Technique for Imbalanced Dataset