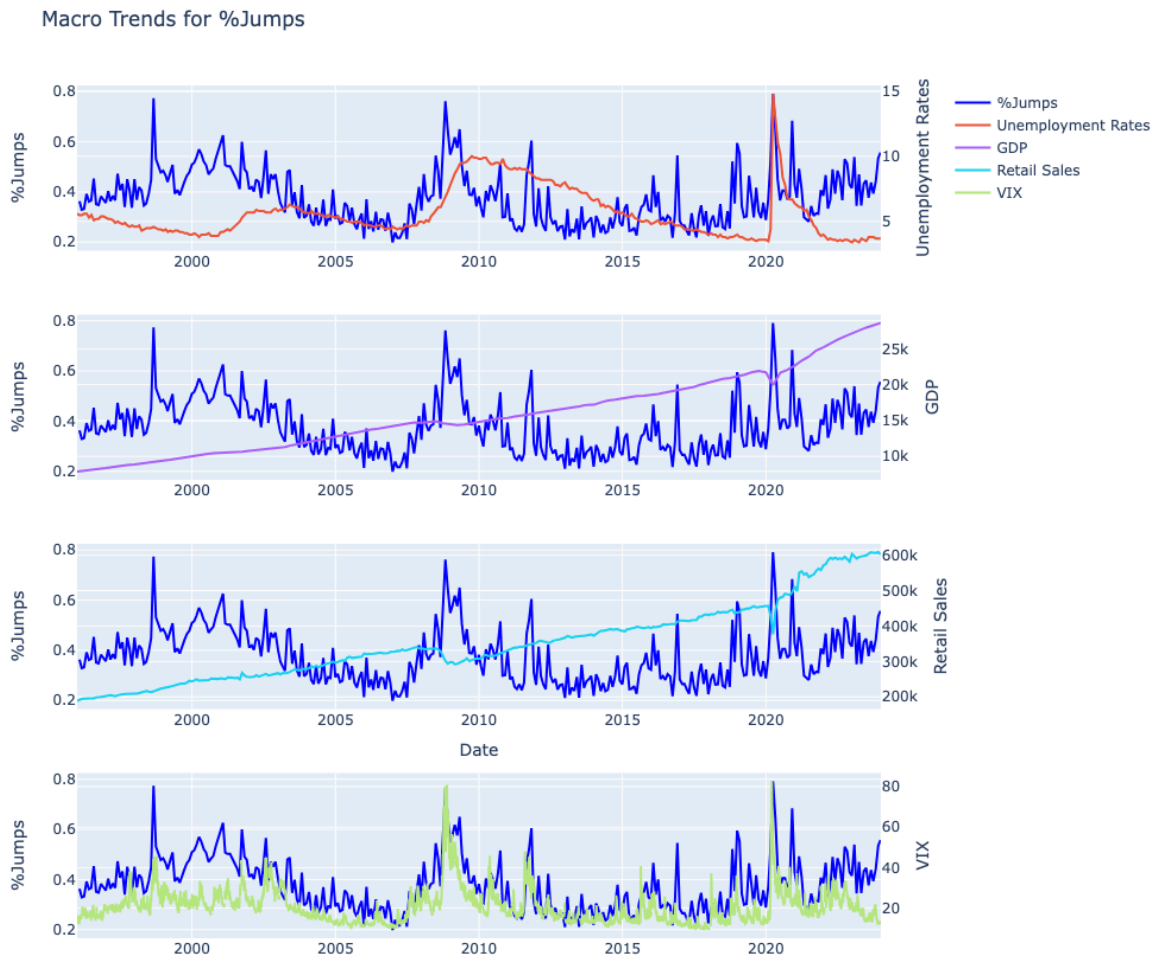
AS6 Report Naman Kedia

Macroeconomic indicators vs Percentage Jumps

Indicators:

- Retail Sales, VIX, Unemployment Rates, GDP



- Therefore we can see that the VIX is a good indicator of the %jumps as other macro's don't directly impact the stocks on a continuous basis.
- However we should note that the Unemployment rates are also loosely showing correlation ie around the 2008 and 2020 era. Both these times due to a major financial crisis, Covid trading activity might have increased leading to more volatility and hence the trends.

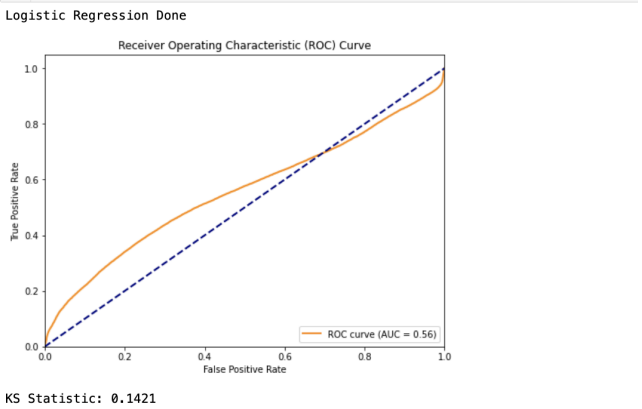
List of covariates selected based on model as possible indicators of strong prediction including those from previous assignments

- ['PERMNO', 'RET', 'vwretd', 'beta', 'industry_code', 'PRC', 'return', 'momentum', 'volatility', 'VIX', 'year']

Choosing the best window among the 3 - simple, rolling and fixed I choose the siple window since it was marginally better than the other 2. [*Graph*: *Logistic Simple vs Rolling vs Fixed window*]

AS6 Report Naman Kedia

Simple Logistic Regression Plot



Final AUC Metrics and misclassification table

Metric|Model	Logistic	Lasso	Ridge	Knn	XGboost	ANN
AUC	.5557	.56	.554	.967	1.00	0.998
Misclassification	.3586	.3595	.3618	.0081	0	0
Hyperparameter	-	λ =1251	λ =1000	K=5	Num = 100	-

Based on the table provided, the following analysis of the model performances and accuracies:

- AUC Comparison:
  - KNN, XGBoost, and ANN models show the highest AUC of 0.99, indicating high performance of advanced models as compared to linear models
  - Logistic Regression, Lasso, and Ridge models have significantly lower AUC values (0.5557, 0.56, and 0.554 respectively), suggesting much weaker predictive performance.
- Hyperparameters:
  - For Lasso and Ridge, relatively high lambda values (1251 and 1000) suggest strong regularization was needed on variables
  - KNN's best performance was achieved with a relatively small number of neighbors (K=5), indicating local patterns in the data.
  - XGBoost required only 100 boosting rounds, which is relatively low, suggesting it quickly captured the important patterns.
- Missing Information:
  - Misclassification rates are high for simple logistic, lasso and ridge. We can see that the data misclassificaiton is highest for actual 1 values. This is bc of a smaller sample set observed in training data for y=1. We can improve this potentially by taking a better split of data based on y values as opposed to a simple random split.
- Model Complexity:
  - More complex models (KNN, XGBoost, ANN) significantly outperformed simpler linear models, suggesting the underlying relationships in the data are likely non-linear.
  - The poor performance of linear models might indicate that individual features are not strongly predictive on their own, and that interactions or non-linear combinations of features are crucial.

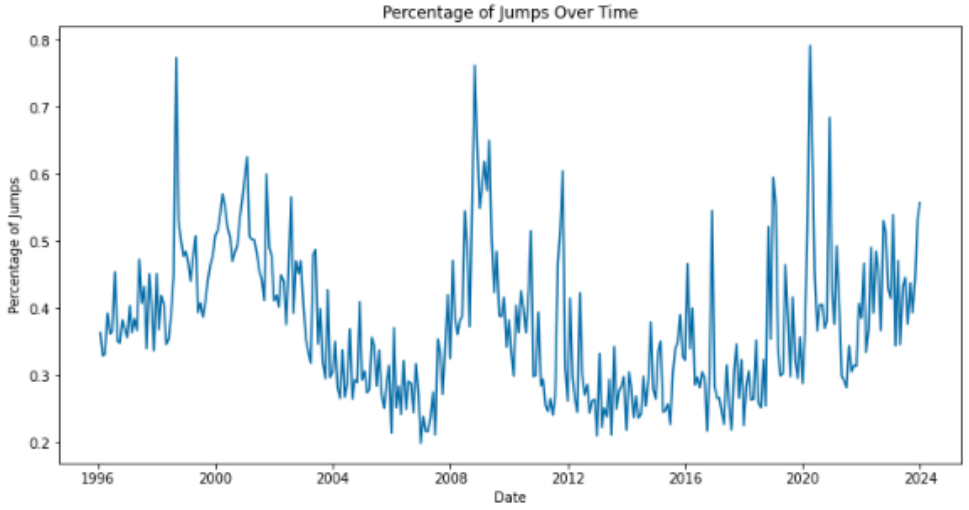
Final Analysis:

XGBoost, ANNs and KNN's have the lowest misclassification rates respectively. They excel at capturing complex, non-linear relationships in data even if the data is non-uniformly split. They can model intricate patterns that simpler models like logistic regression might miss due to non-linearity or sampling bias

**********End of Report Analysis**********

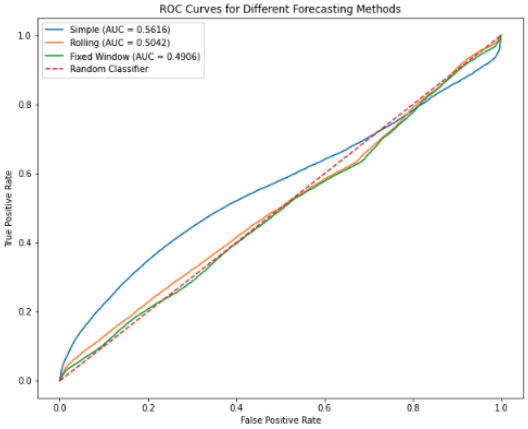
AS6 Report Naman Kedia

Other Graphs and metric plots:



Logistic Simple vs Rolling vs Fixed window

```
AUC (Simple): 0.5616
AUC (Rolling): 0.5042
AUC (Fixed Window): 0.4906
```



ANN RoC Performance

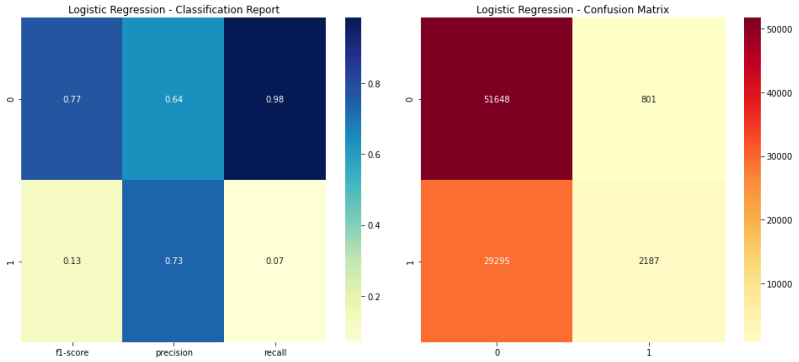
```
ANN Performance:
              precision    recall  f1-score   support

           0       0.99      1.00      0.99     52426
           1       1.00      0.99      1.00     16041
           2       1.00      0.97      0.98     15464

    accuracy                           0.99     83931
   macro avg       0.99      0.99      0.99     83931
weighted avg       0.99      0.99      0.99     83931

Confusion Matrix:
[[52365     6    55]
 [  122 15919     0]
 [  531     0 14933]]
AUC-ROC (multi-class): 0.9998
```

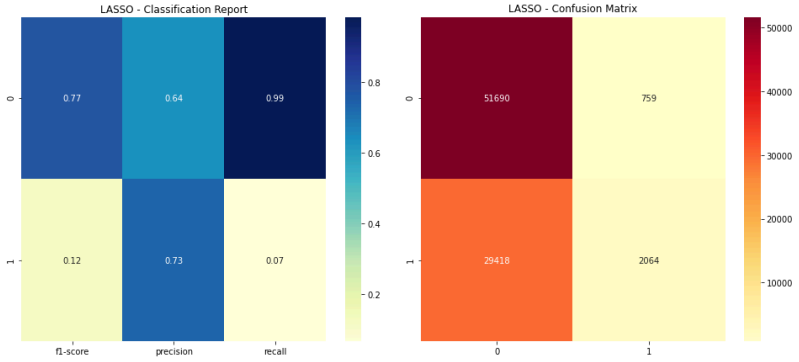
AS6 Report Naman Kedia

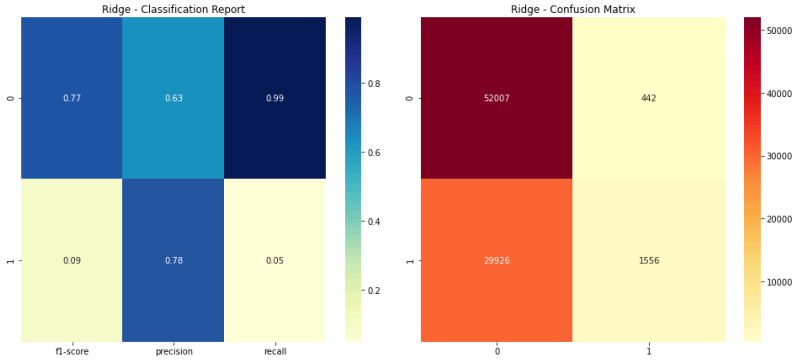
All ROC tables along with confusion metrics and their visualisation (Simple vs Lasso vs Ridge vs KNN vs XgBoost Confusion Matrices)

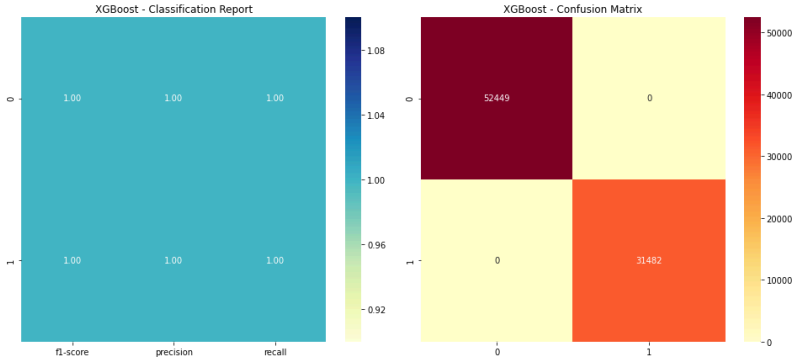
```
Logistic Regression Performance:
              precision    recall  f1-score   support

           0       0.64      0.98      0.77     52449
           1       0.73      0.07      0.13     31482

    accuracy                           0.64     83931
   macro avg       0.69      0.53      0.45     83931
weighted avg       0.67      0.64      0.53     83931

Confusion Matrix:
[[51648   801]
 [29295  2187]]
AUC-ROC: 0.5557

LASSO Performance:
              precision    recall  f1-score   support

           0       0.64      0.99      0.77     52449
           1       0.73      0.07      0.12     31482

    accuracy                           0.64     83931
   macro avg       0.68      0.53      0.45     83931
weighted avg       0.67      0.64      0.53     83931

Confusion Matrix:
[[51690   759]
 [29418  2064]]
AUC-ROC: 0.5562

Ridge Performance:
              precision    recall  f1-score   support

           0       0.63      0.99      0.77     52449
           1       0.78      0.05      0.09     31482

    accuracy                           0.64     83931
   macro avg       0.71      0.52      0.43     83931
weighted avg       0.69      0.64      0.52     83931

Confusion Matrix:
[[52007   442]
 [29926  1556]]
AUC-ROC: 0.5546

KNN Performance:
              precision    recall  f1-score   support

           0       0.91      0.98      0.94     52449
           1       0.96      0.85      0.90     31482

    accuracy                           0.93     83931
   macro avg       0.94      0.91      0.92     83931
weighted avg       0.93      0.93      0.93     83931

Confusion Matrix:
[[51229  1220]
 [ 4795 26687]]
AUC-ROC: 0.9673

XGBoost Performance:
              precision    recall  f1-score   support

           0       1.00      1.00      1.00     52449
           1       1.00      1.00      1.00     31482

    accuracy                           1.00     83931
   macro avg       1.00      1.00      1.00     83931
weighted avg       1.00      1.00      1.00     83931

Confusion Matrix:
[[52449     0]
 [    0 31482]]
AUC-ROC: 1.0000
```

AS6 Report Naman Kedia









References

- *ChatGPT used for generating a few visualisation portions for eg.visualize_model_performance() and some variable selection metrics in make_features()*