

# Bitcoin Classifier

DSI - 13 Capstone



Julian Broche - East Coast

# Background

I'm a quantitative researcher for a large asset manager and was tasked to build a model, which generates buy or sell decisions for BTC

# DS Problem

# Predicting Time Series Data is Hard

# Hypothesis

Leveraging blockchain,  
search and social media  
data allows for predictions  
of BTC's daily close better  
than a baseline model

# Methodology

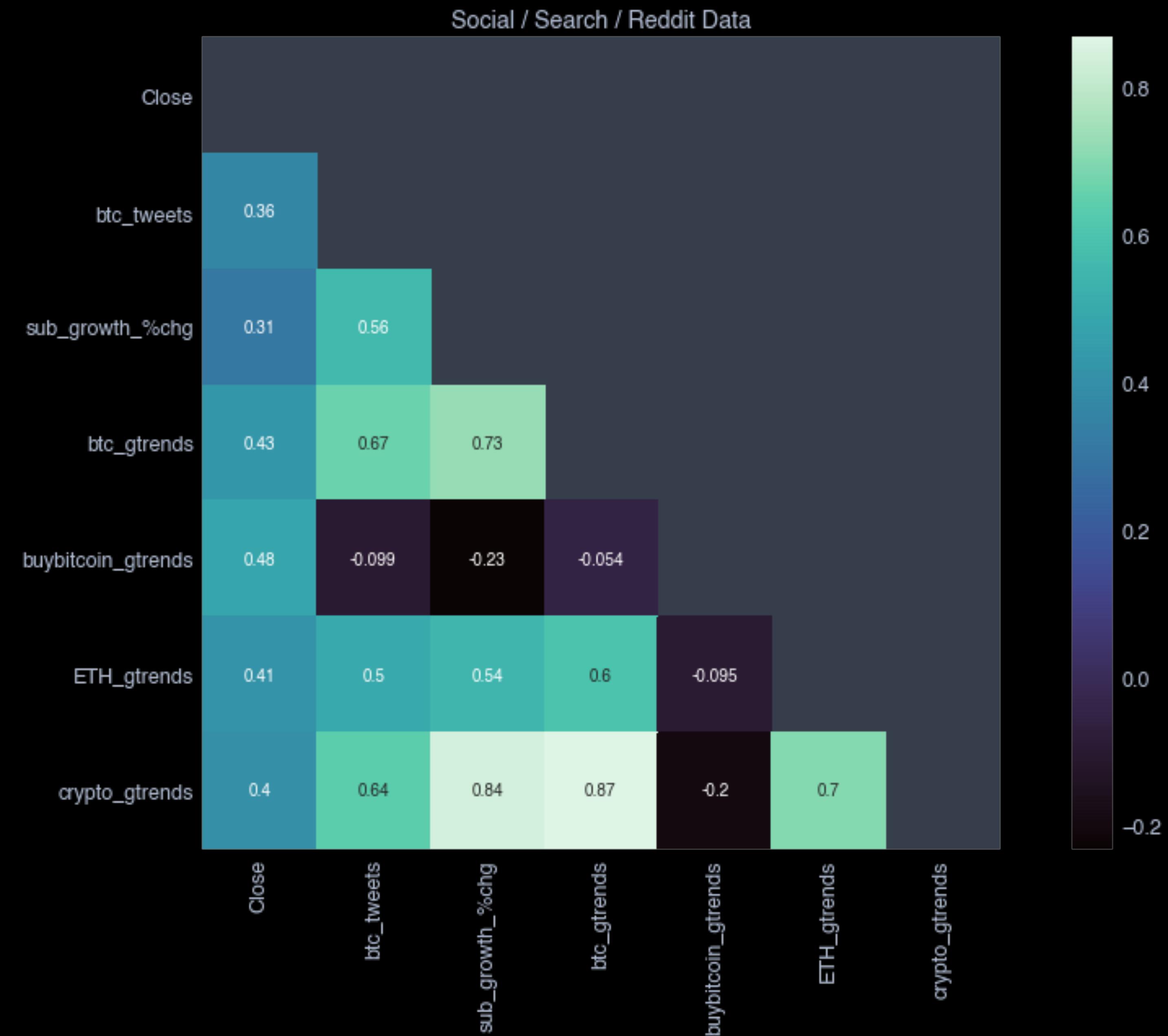


- **Cryptory - Data Extraction**
  - Google Search Trends
  - Twitter
  - Reddit
  - Blockchain Metrics
  - Yahoo Finance
- **Feature Engineering - Clean/Transform**
  - Remove Dups, Generate New Features
  - Scaling, PCA
  - Polynomial Features
- **Modeling - Feed Models**
  - SVM & XGB
  - Neural Networks

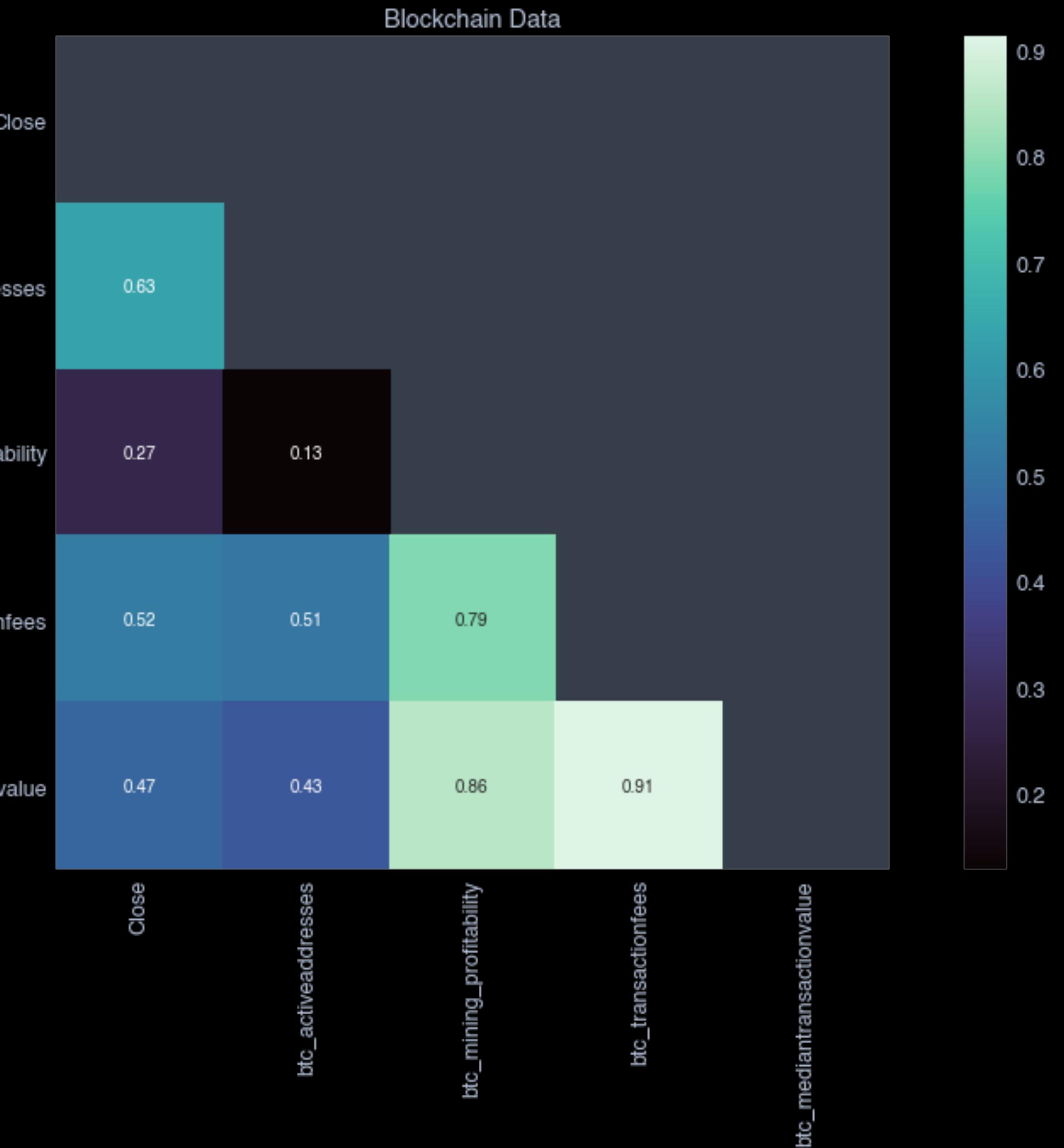
# Quantitative findings



# Proxies for Retail Demand, Overall Engagement

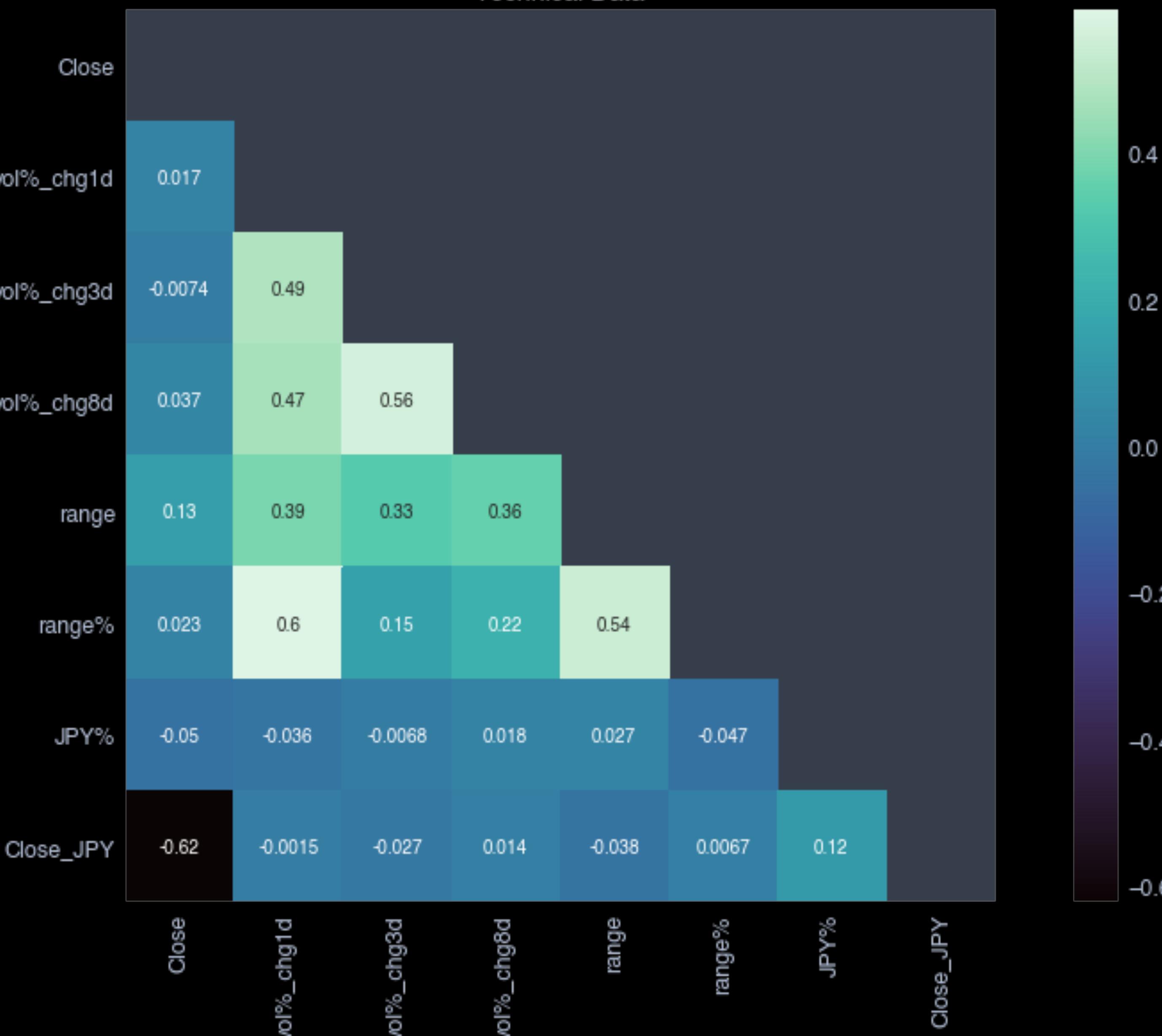


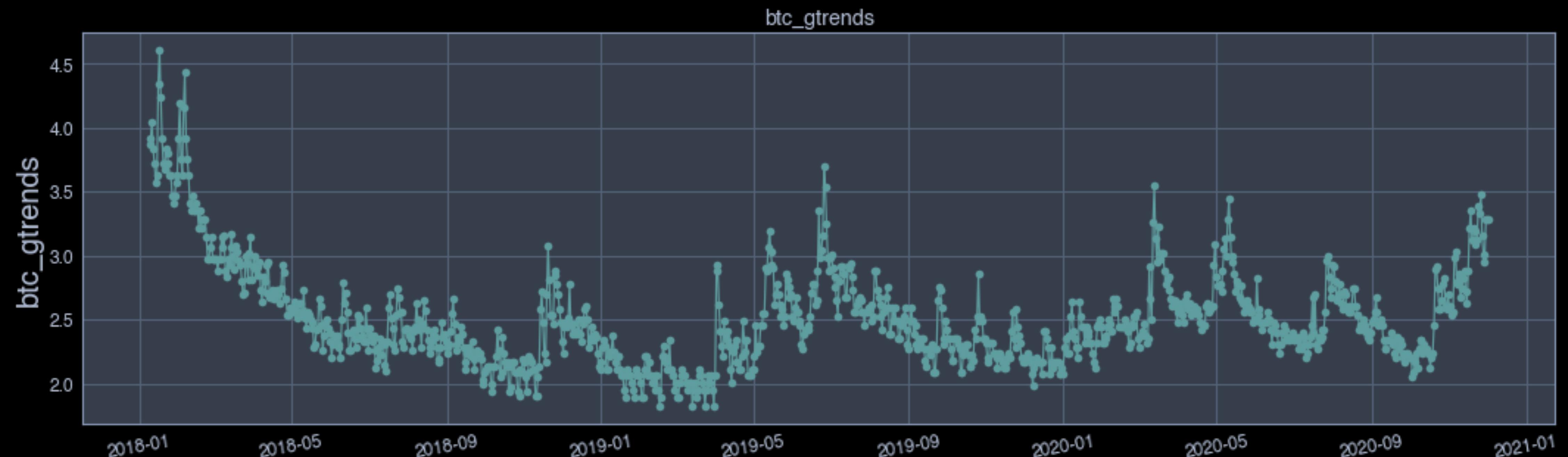
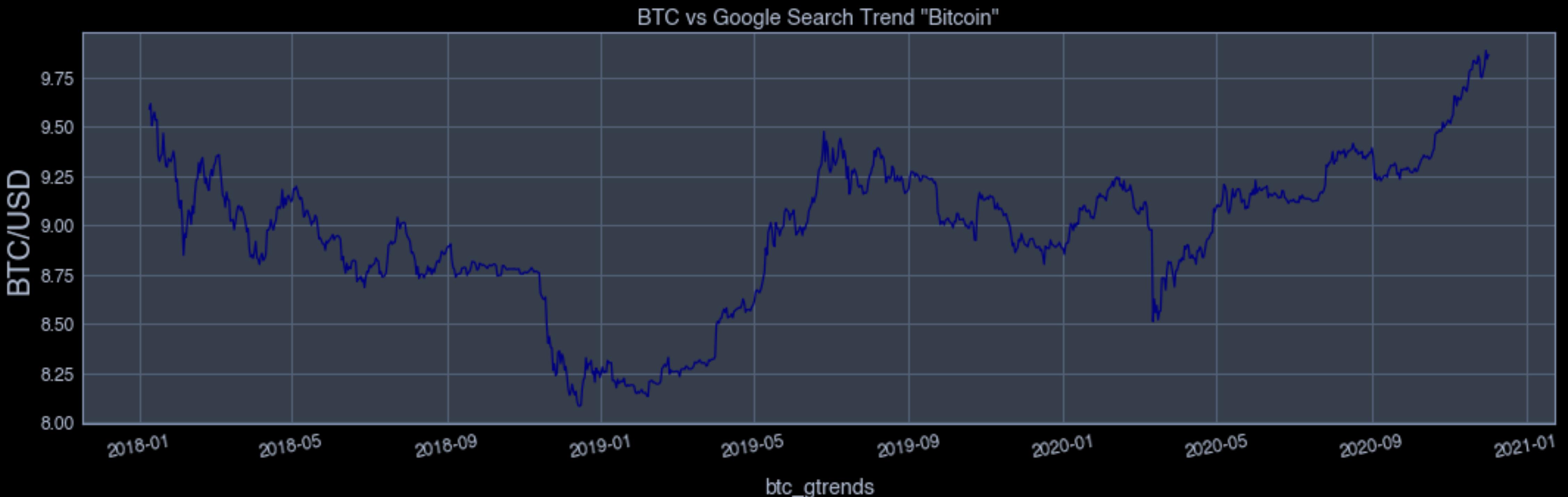
# Proxies for Crypto Adoption, Miner Supply, Overall Network Health



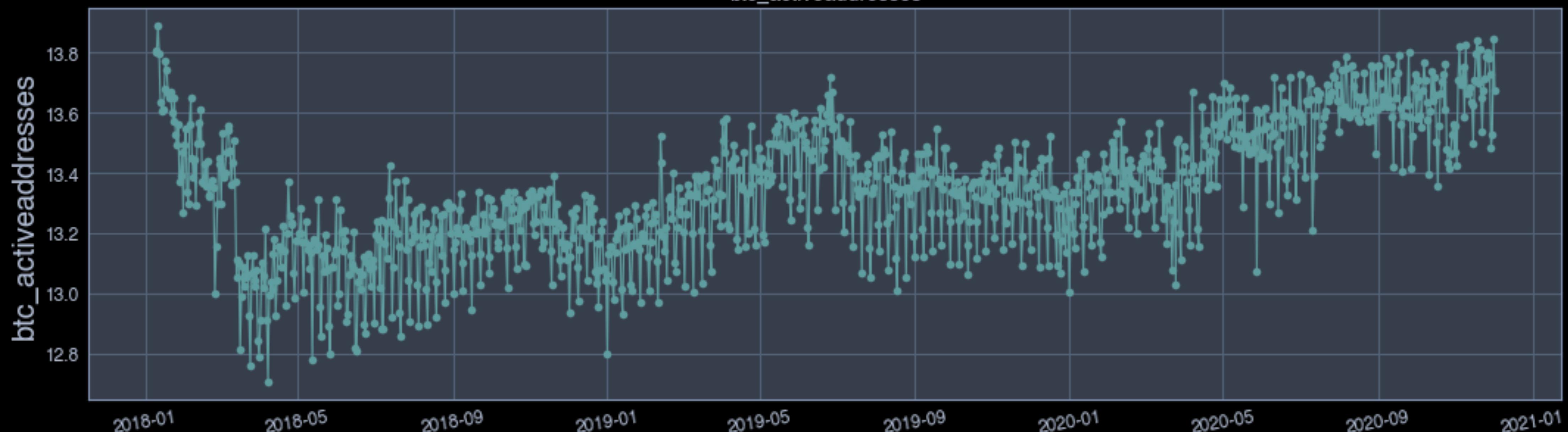
# Proxies for Technical Trading Signals

Technical Data

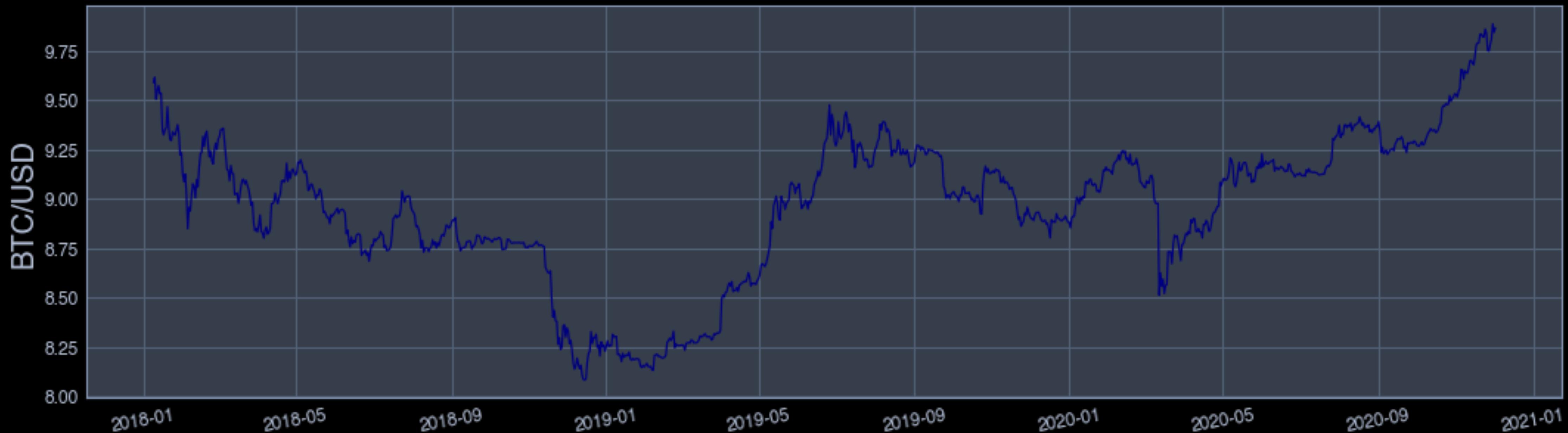




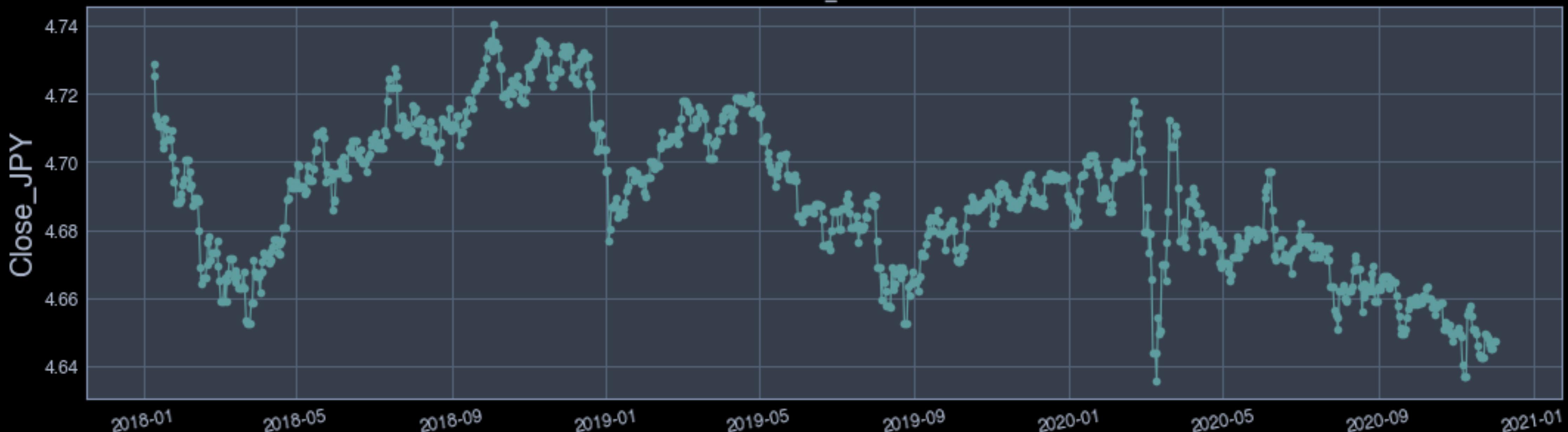
BTC Active Addresses



BTC vs JPY



Close\_JPY



# Lets Talk Supervised Models



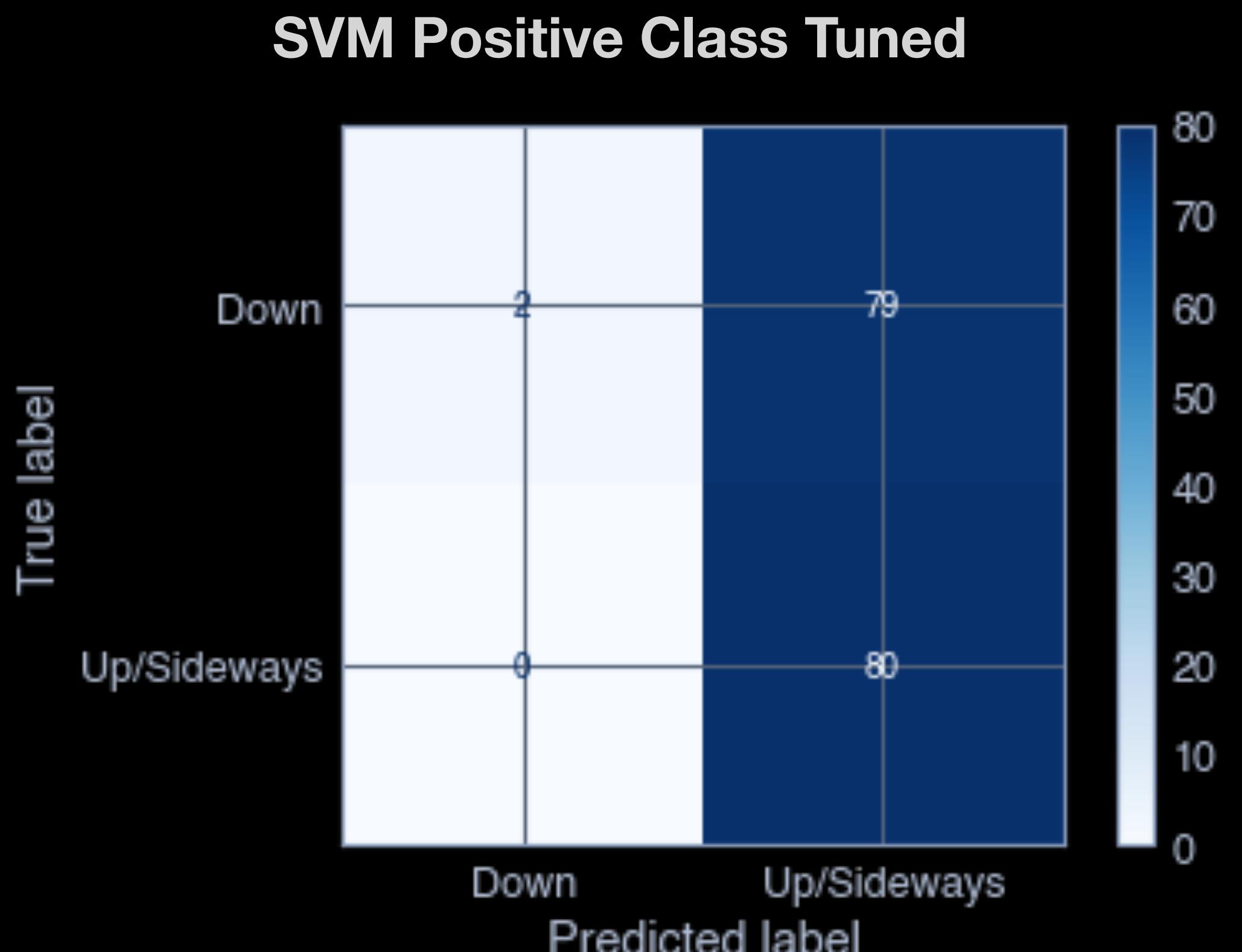
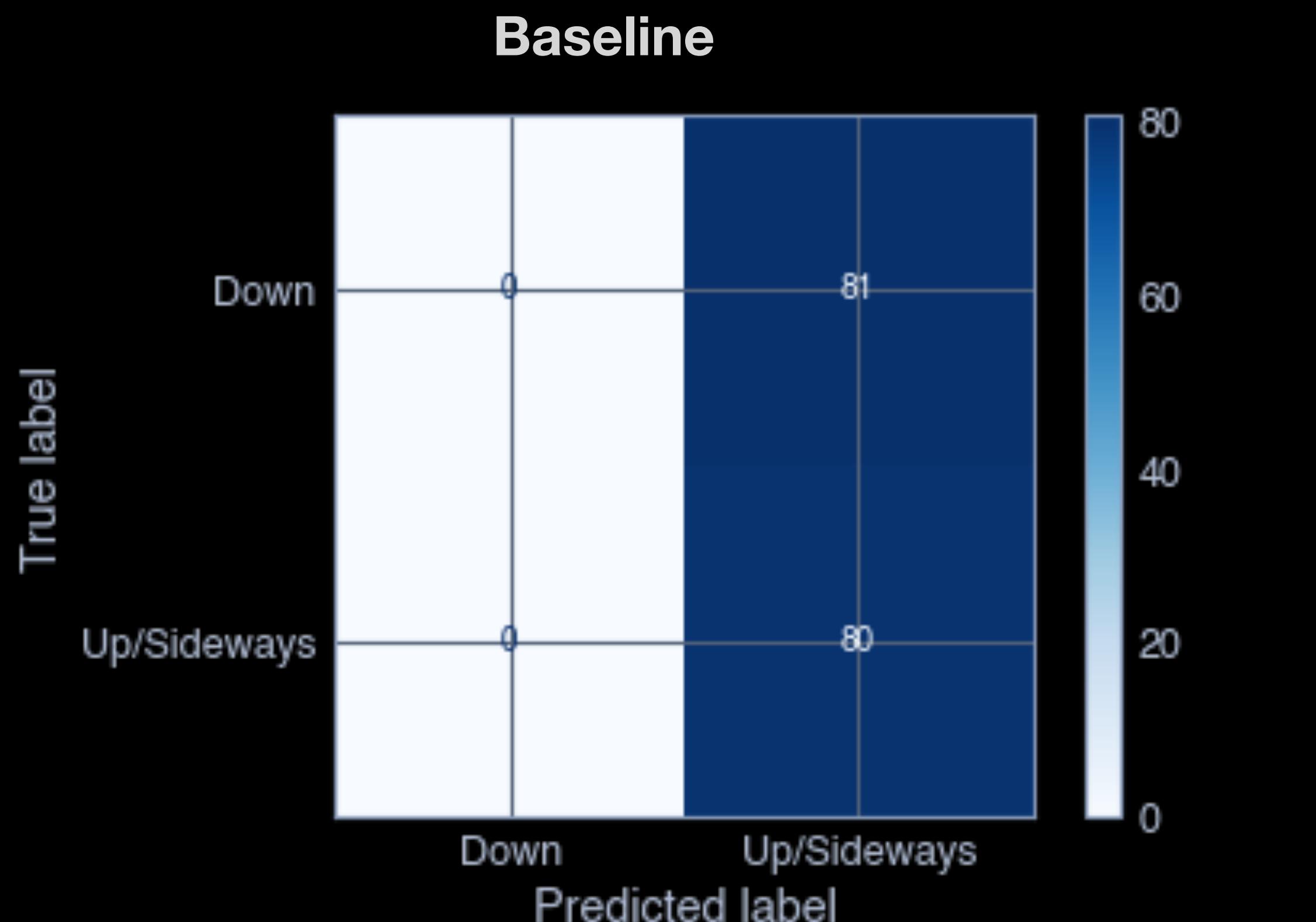
# Baseline Model Comp

(n\_estimators=100, random\_state=42)

	train_accuracy	test_accuracy	ROC_AUC	test_recall	test_precision	F1	train_test_diff
<b>Most Frequent</b>	0.526953	0.496894	0.500000	1.0000	0.496894	0.663900	0.030058
<b>SVM</b>	0.633663	0.496894	0.534105	0.7875	0.496063	0.608696	0.136769
<b>Bagging Classifier</b>	1.000000	0.540373	0.538194	0.6500	0.530612	0.584270	0.459627
<b>GradientBoosting Classifier</b>	0.959296	0.521739	0.518364	0.6250	0.515464	0.564972	0.437557
<b>ExtraTrees Classifier</b>	1.000000	0.527950	0.499846	0.5750	0.522727	0.547619	0.472050
<b>Nearest Neighbors</b>	0.775578	0.503106	0.491821	0.6000	0.500000	0.545455	0.272472
<b>xgboost</b>	1.000000	0.527950	0.495525	0.5625	0.523256	0.542169	0.472050
<b>Random Forest</b>	1.000000	0.496894	0.522840	0.5750	0.494624	0.531792	0.503106
<b>DecisionTree Classifier</b>	1.000000	0.509317	0.509259	0.5000	0.506329	0.503145	0.490683
<b>AdaBoost Classifier</b>	0.855886	0.465839	0.506636	0.5125	0.465909	0.488095	0.390047
<b>Logistic Regression</b>	0.634763	0.428571	0.418673	0.5250	0.437500	0.477273	0.206192

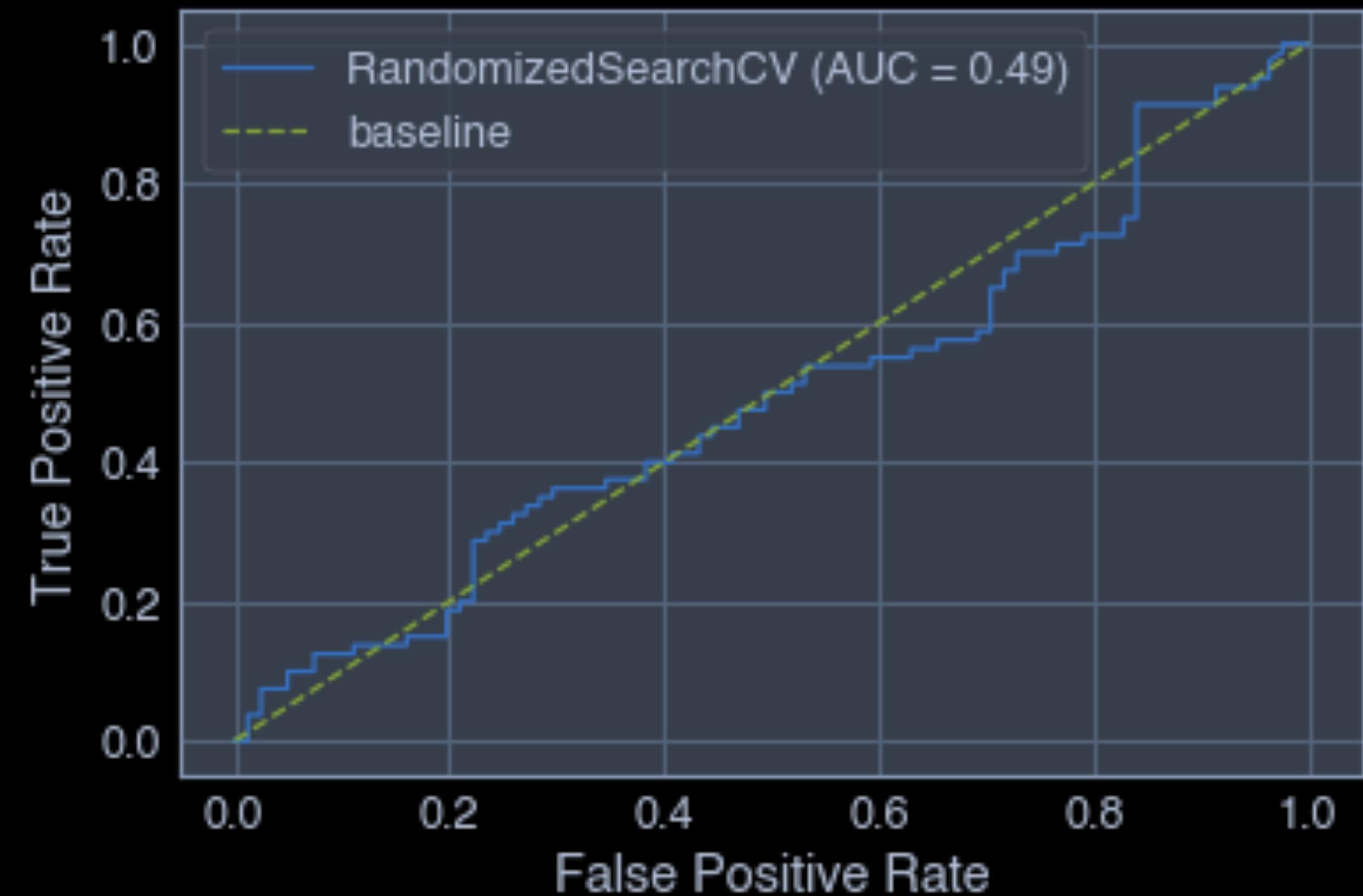
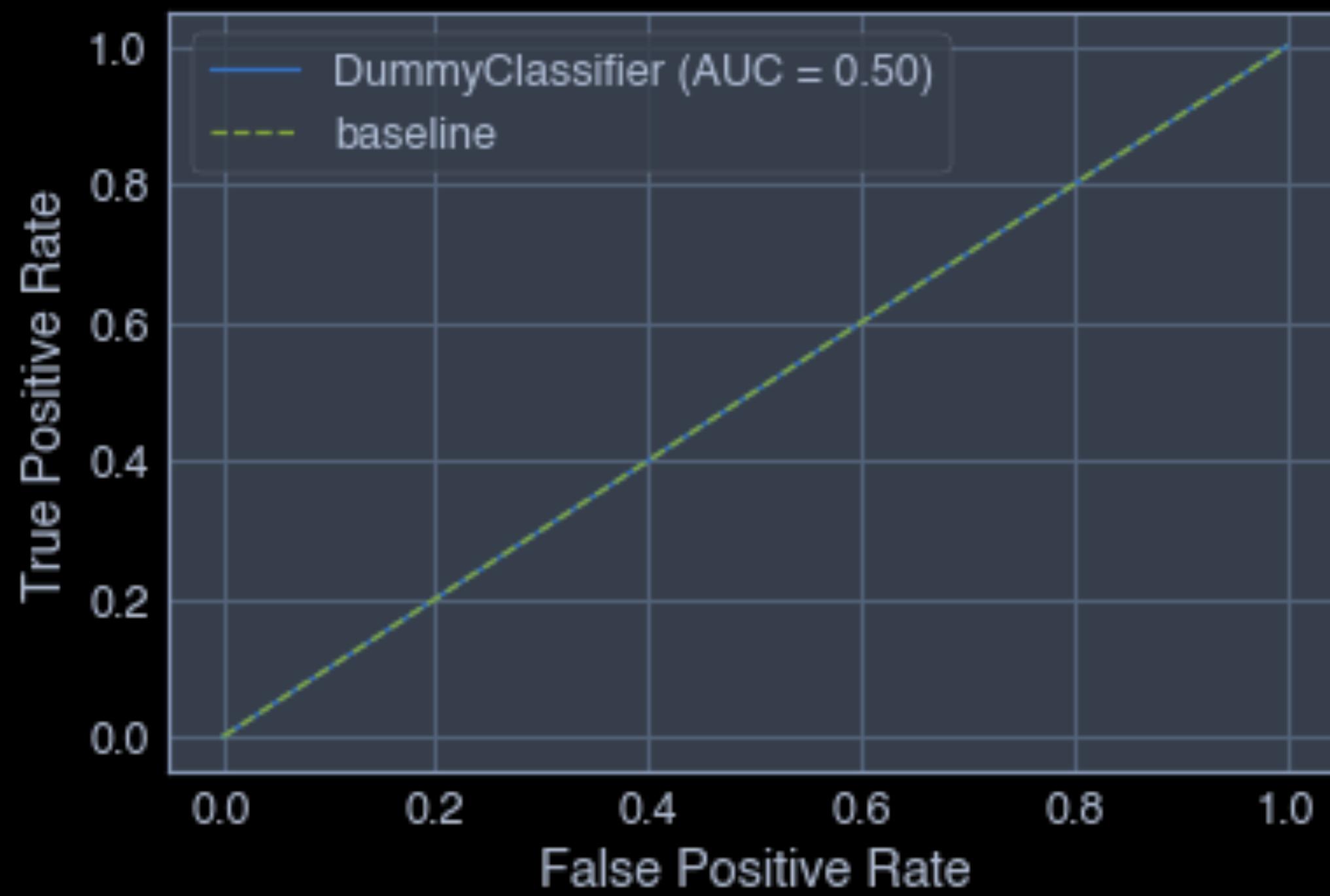
# Tuned Model Matrix Comp

	train_accuracy	test_accuracy	ROC_AUC	test_recall	test_precision	F1	train_test_diff
SVM_Rs	0.536854	0.509317	0.491744	1.0	0.503145	0.669456	0.027537
Most Frequent	0.526953	0.496894	0.500000	1.0	0.496894	0.663900	0.030058



# Tuned Model ROC\_AUC Comp

	train_accuracy	test_accuracy	ROC_AUC	test_recall	test_precision	F1	train_test_diff
SVM_Rs	0.536854	0.509317	0.491744	1.0	0.503145	0.669456	0.027537
Most Frequent	0.526953	0.496894	0.500000	1.0	0.496894	0.663900	0.030058



# Tuned Models Comp

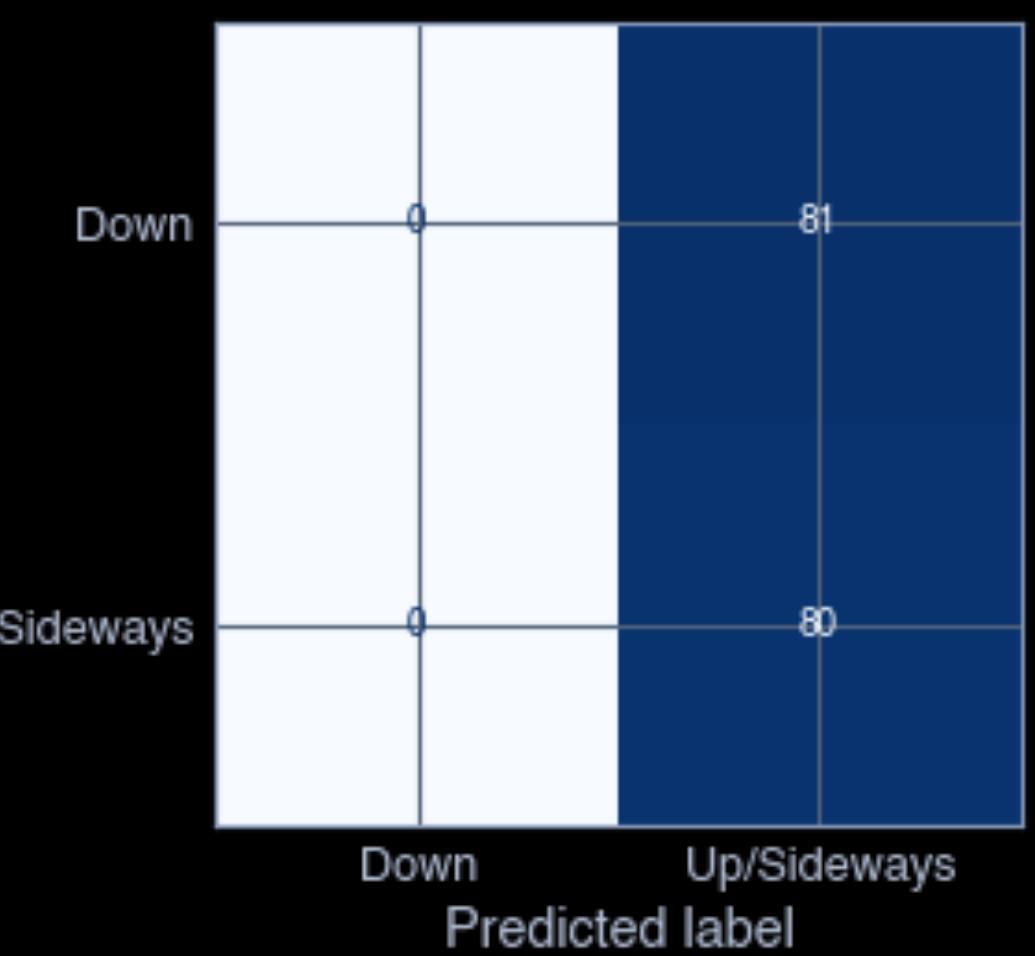
	<b>train_accuracy</b>	<b>test_accuracy</b>	<b>ROC_AUC</b>	<b>test_recall</b>	<b>test_precision</b>	<b>F1</b>	<b>train_test_diff</b>
<b>SVM_Recall Tune</b>	0.539054	0.509317	0.482253	1.0000	0.503145	0.669456	0.029737
<b>Most Frequent</b>	0.526953	0.496894	0.500000	1.0000	0.496894	0.663900	0.030058
<b>SVM_Balanced Tune</b>	0.924092	0.571429	0.543210	0.8375	0.544715	0.660099	0.352664

**Tune for Balance or for High Recall/Sensitivity??**

# Confusion Matrix Comp

**Baseline**

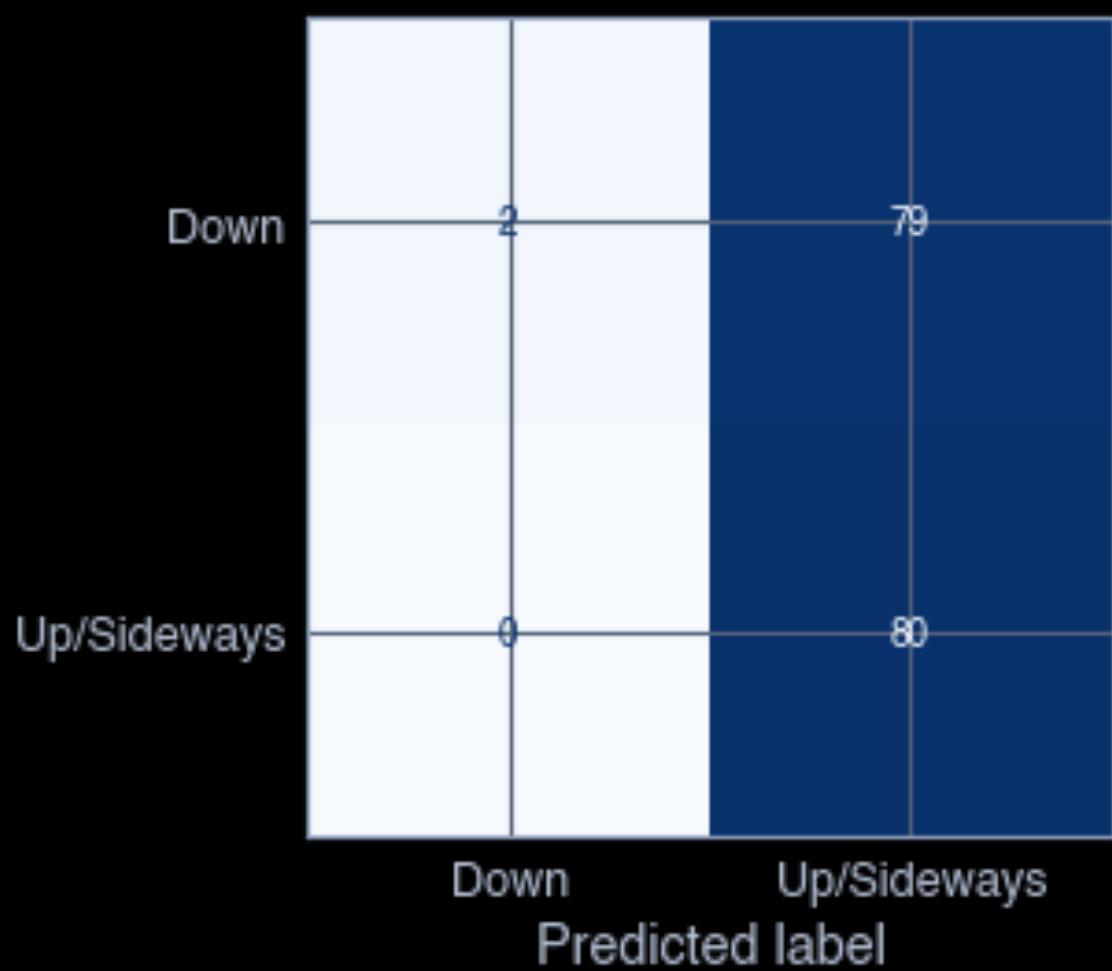
True label



Accuracy Score: 0.496%

**SVM\_Recall Tune**

True label

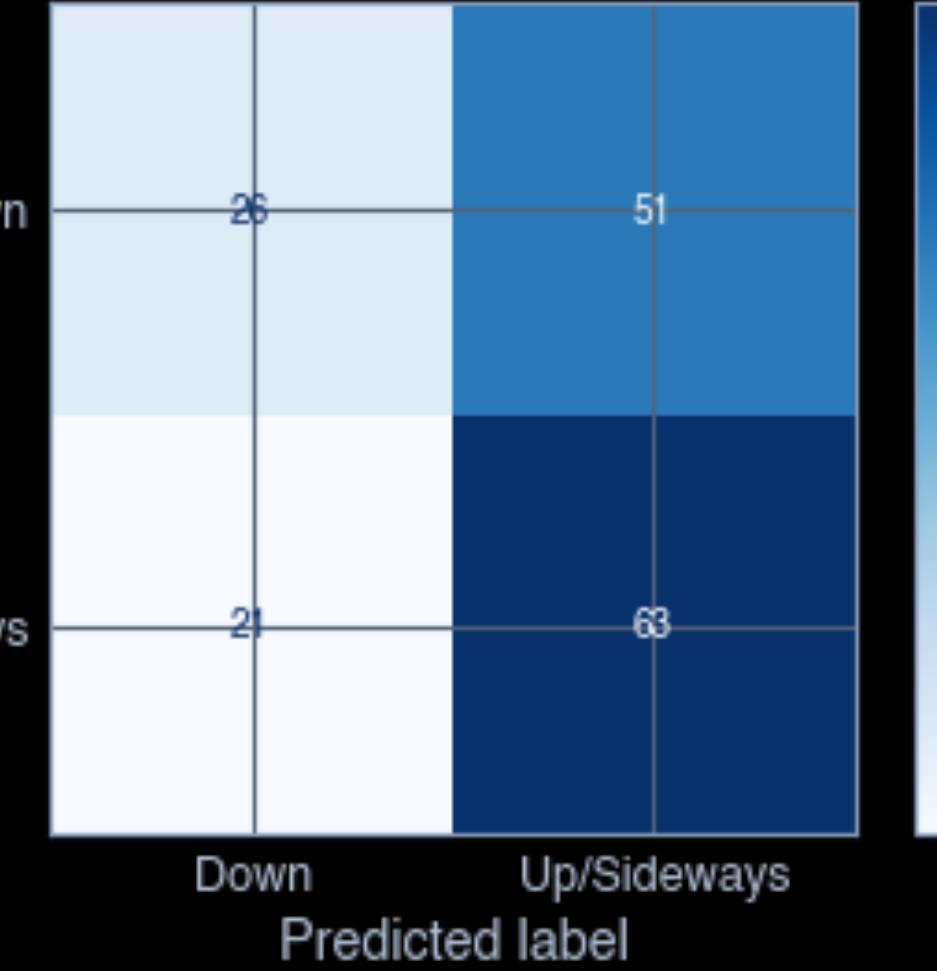


Accuracy Score: 0.509%

Gs Best\_Score: 0.534%

**SVM\_Balance Tune**

True label

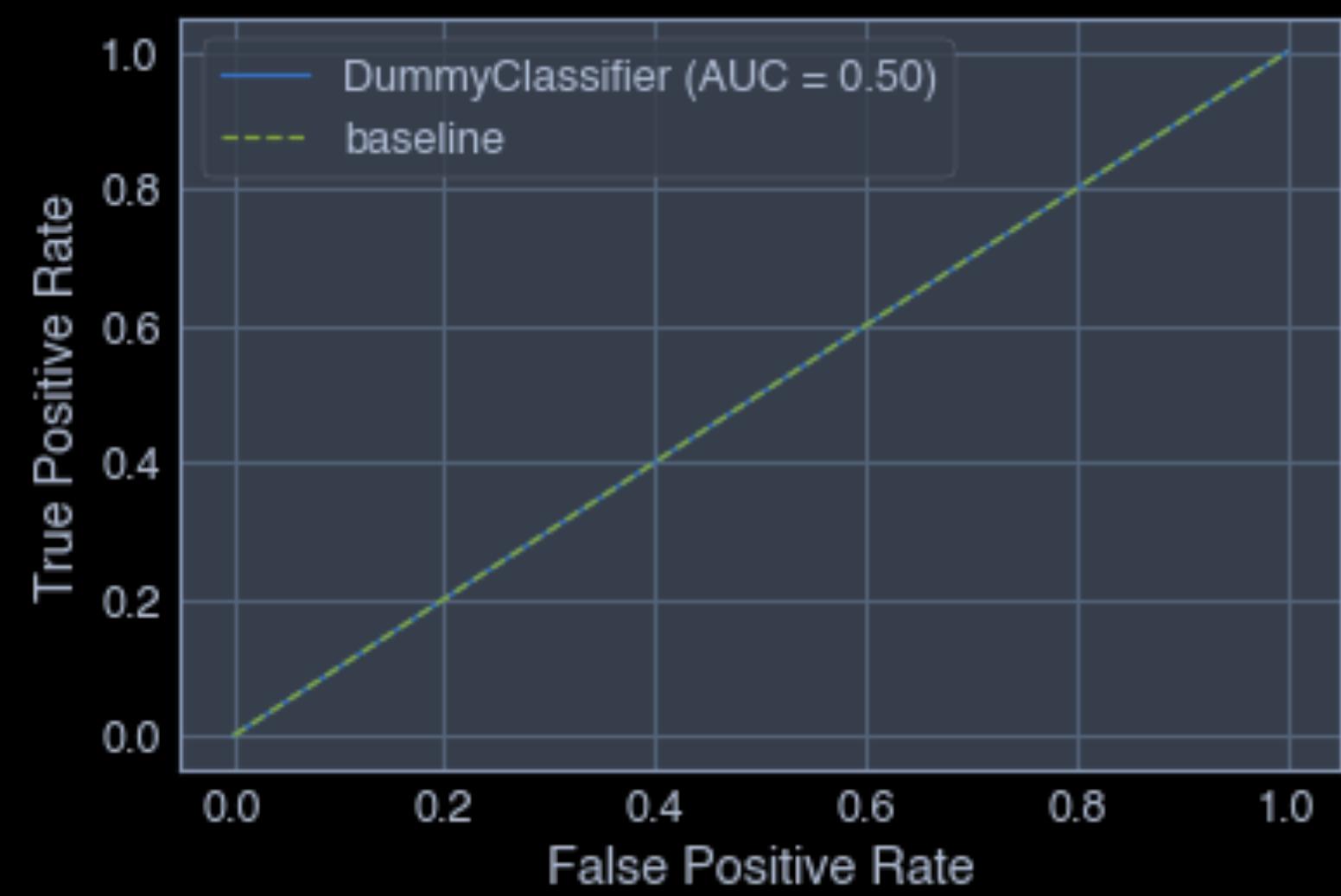


Accuracy Score: 0.552%

Gs Best\_Score: 0.551%

# ROC\_AUC Curve Comp

## Baseline



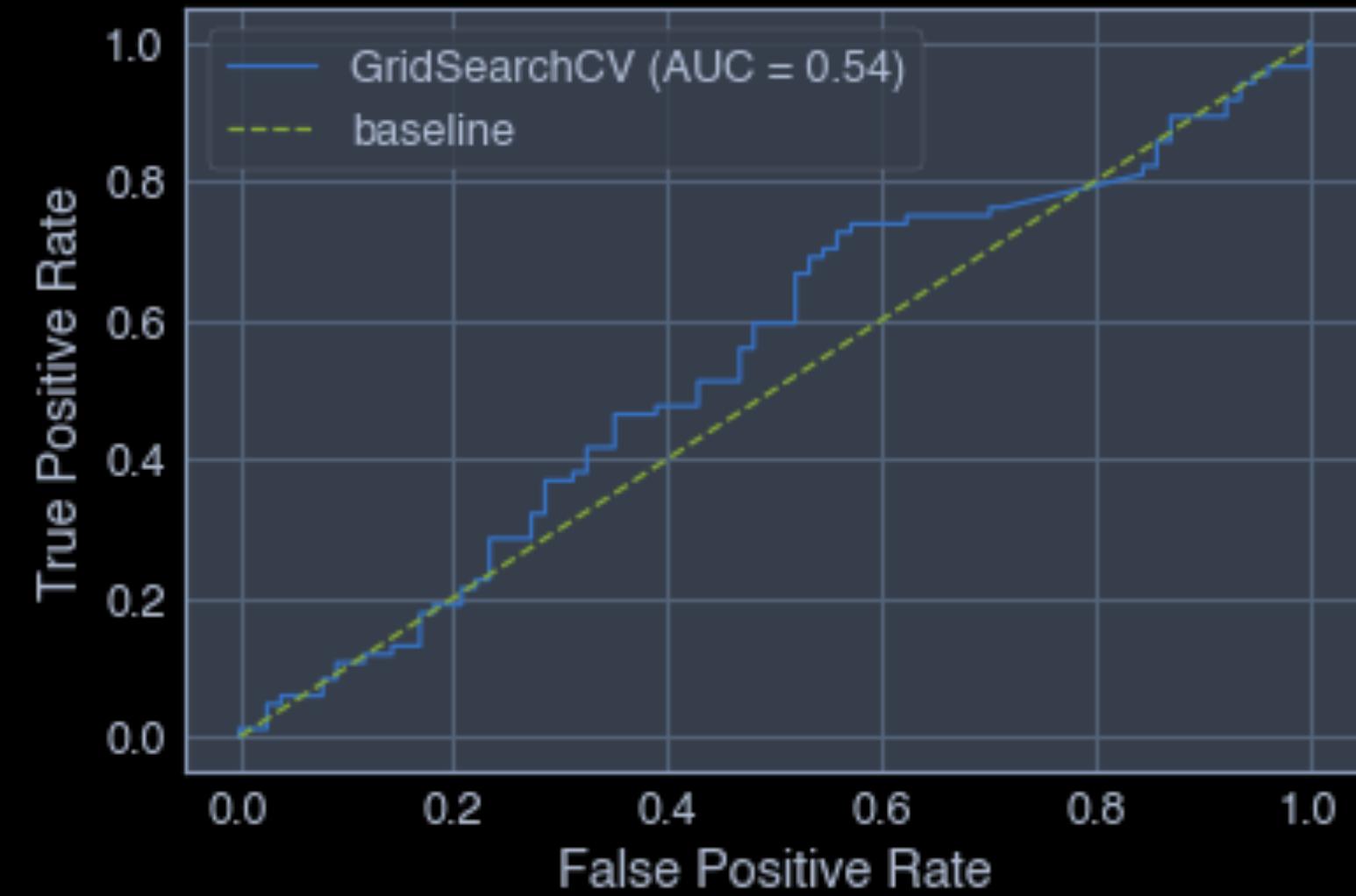
ROC\_AUC Score: 0.5%

## SVM Recall Tune



ROC\_AUC Score: 0.493%

## SVM\_Balance Tune



ROC\_AUC Score: 0.541%

# Lets talk Unsupervised Models

# NN Model Comp

	train_accuracy	test_accuracy	ROC_AUC	test_recall	test_precision	F1	train_test_diff
Neural Network	0.523604	0.590062	0.586038	0.686	0.593718	0.632481	-0.066458
SVM_Recall Tune	0.539054	0.509317	0.482253	1.000	0.503145	0.669456	0.029737
Most Frequent	0.526953	0.496894	0.500000	1.000	0.496894	0.663900	0.030059
SVM_Balanced Tune	0.924092	0.571429	0.543210	0.838	0.544715	0.660099	0.352663

## NN Balance Tune



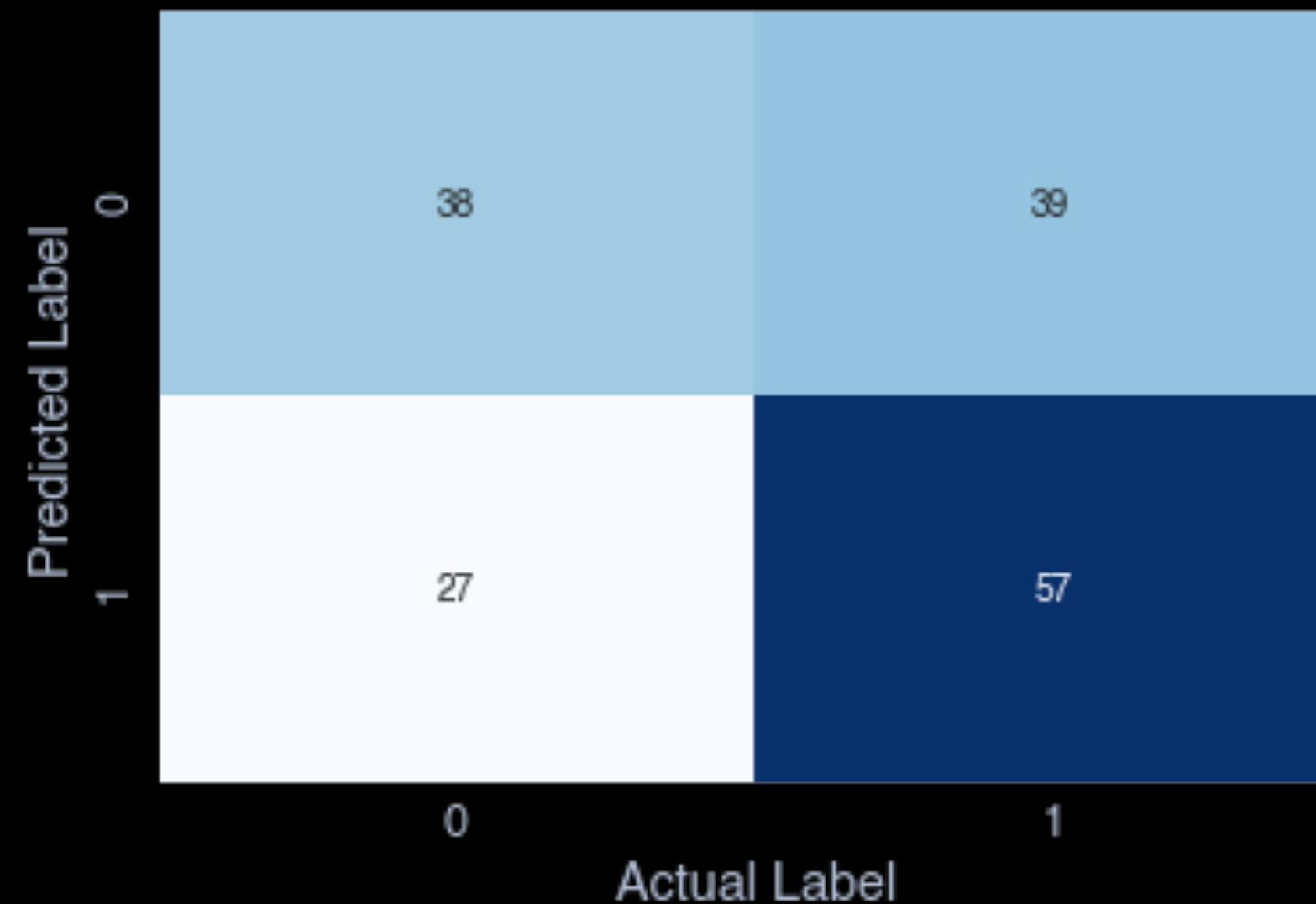
## NN Negative Class Recall Tune



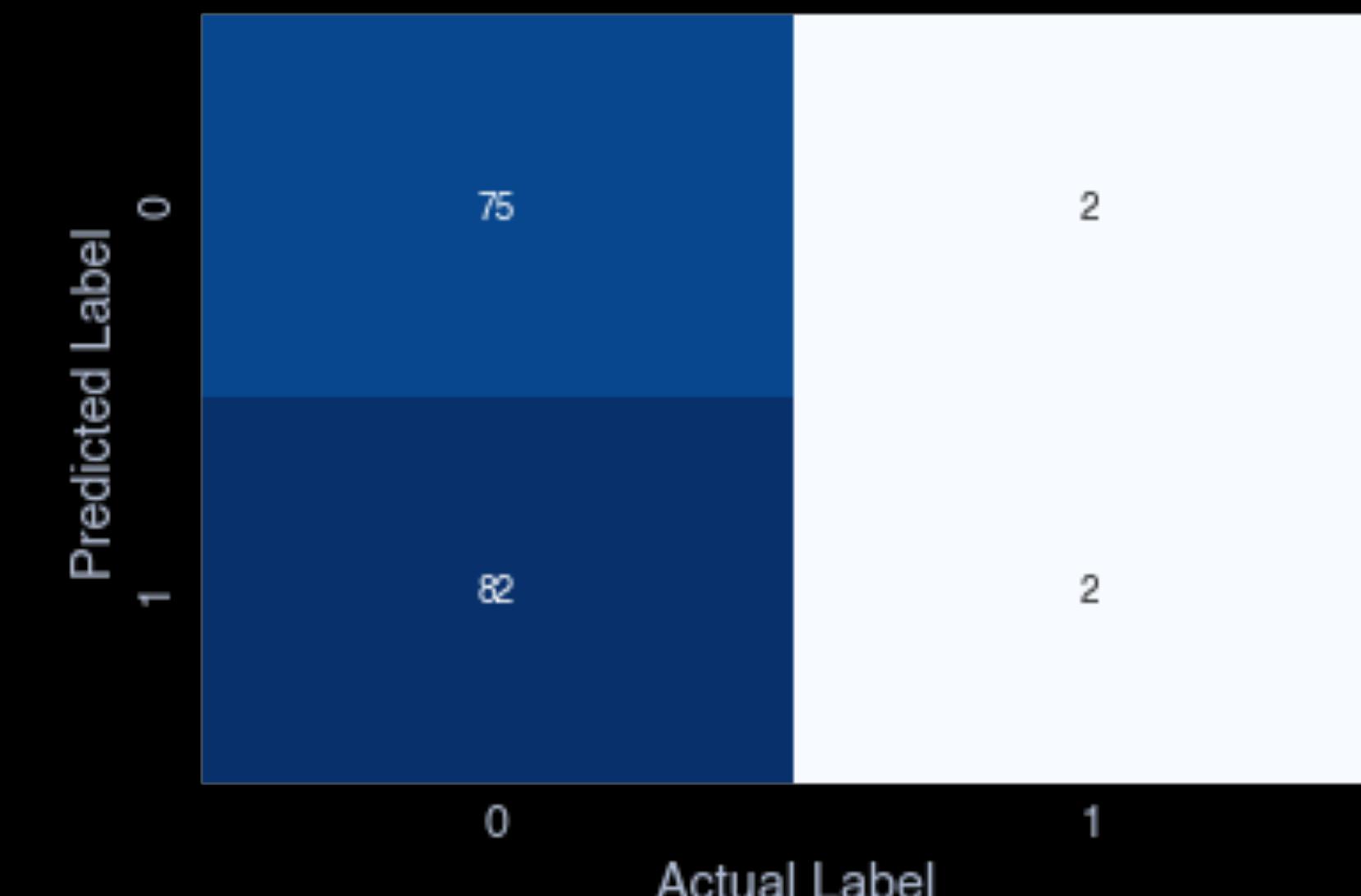
# NN Model Comp

	train_accuracy	test_accuracy	ROC_AUC	test_recall	test_precision	F1	train_test_diff
Neural Network	0.523604	0.590062	0.586038	0.686	0.593718	0.632481	-0.066458
SVM_Recall Tune	0.539054	0.509317	0.482253	1.000	0.503145	0.669456	0.029737
Most Frequent	0.526953	0.496894	0.500000	1.000	0.496894	0.663900	0.030059
SVM_Balanced Tune	0.924092	0.571429	0.543210	0.838	0.544715	0.660099	0.352663
Neural Network Neg Tune	0.521405	0.478260	0.498917	0.020	0.503114	0.646110	0.043145

NN

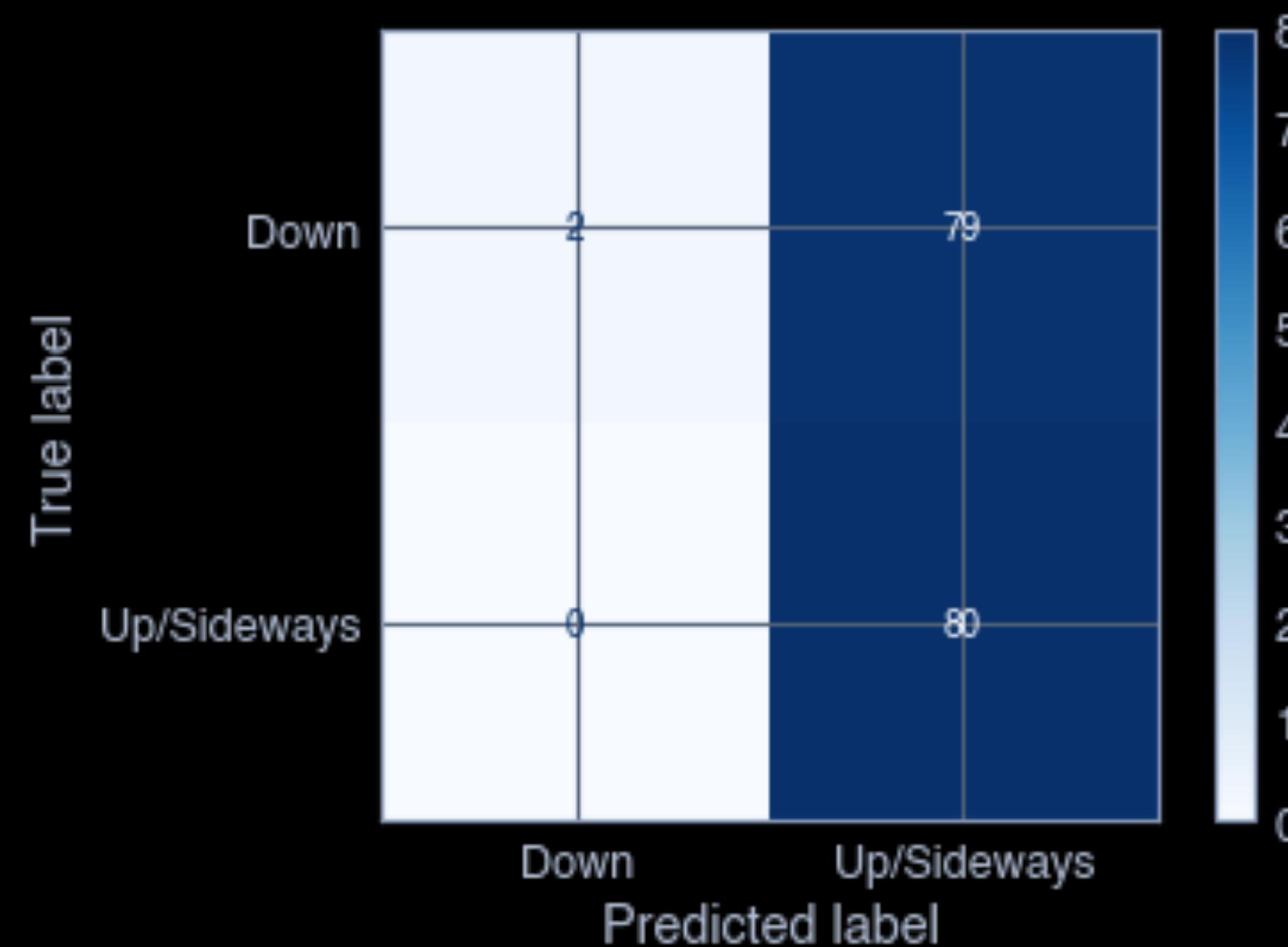


NN Negative Class Tune



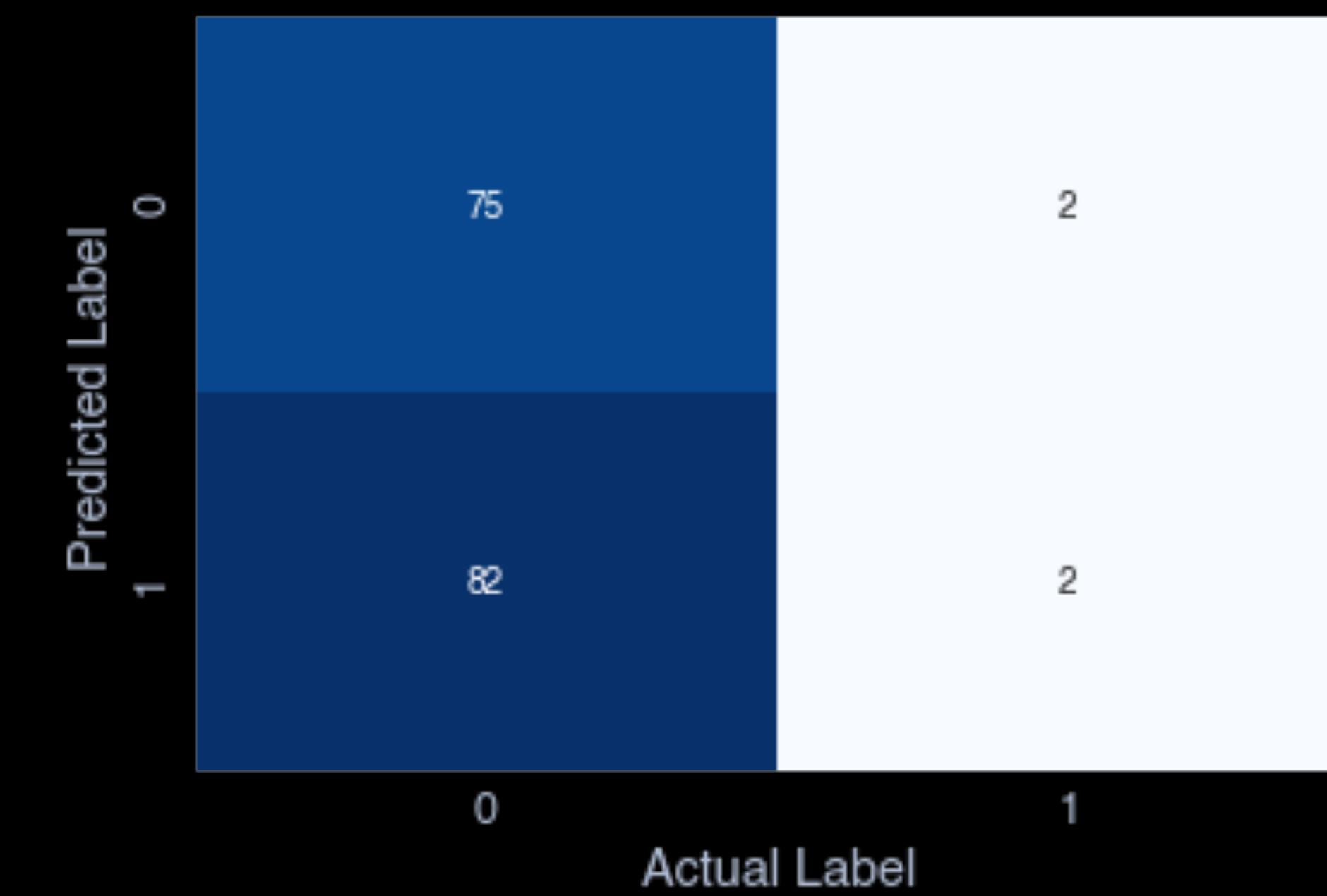
# Confusion Matrix Unsupervised vs Supervised Comp

**SVM Positive Class Tune**



Accuracy Score: 0.509%

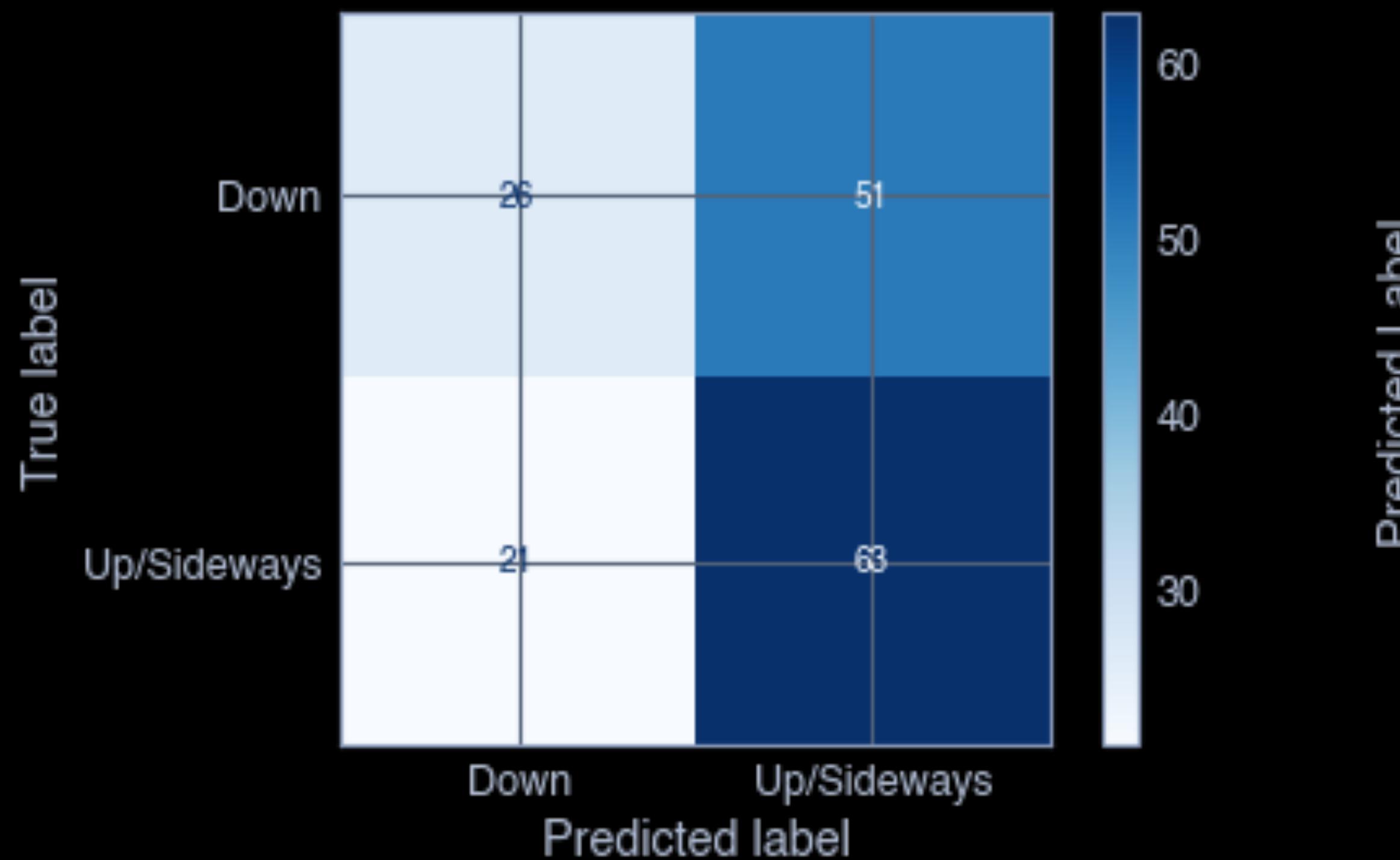
**NN Negative Class Tune**



Accuracy Score: 0.478%

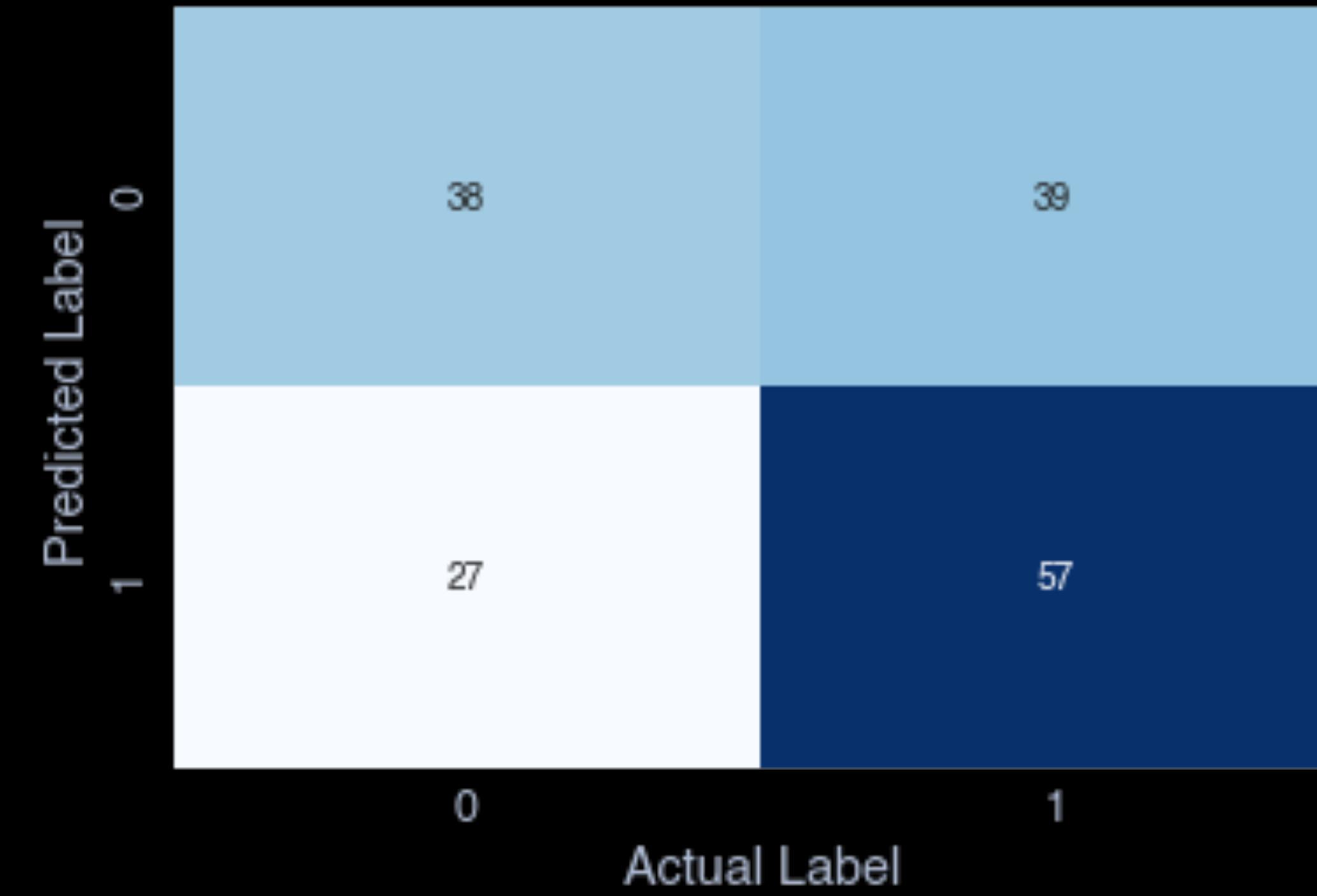
# Confusion Matrix Unsupervised vs Supervised Comp

SVM Balance Tune



Accuracy Score: 0.552%

Neural Network Balance Tune

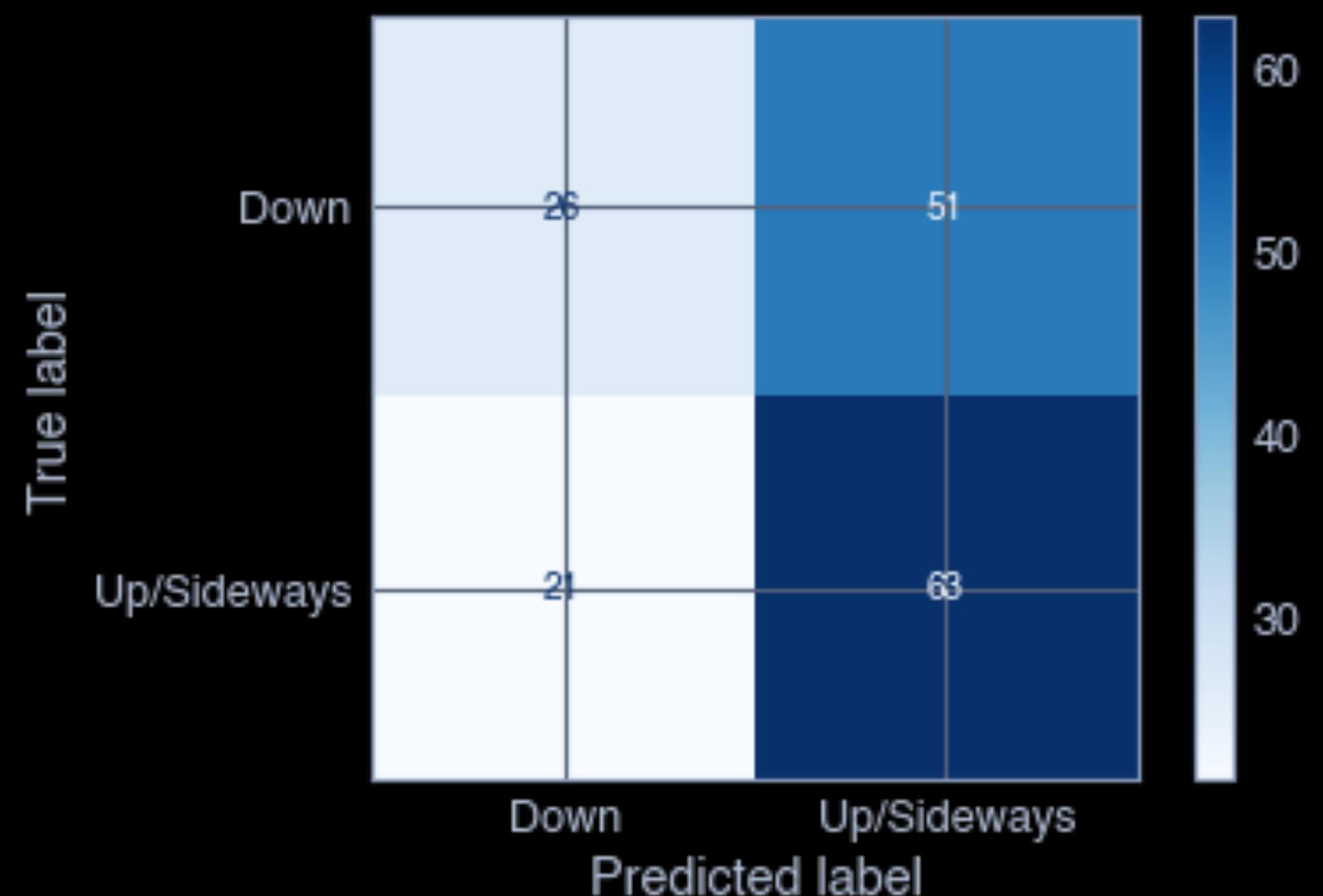


Accuracy Score: 0.590%

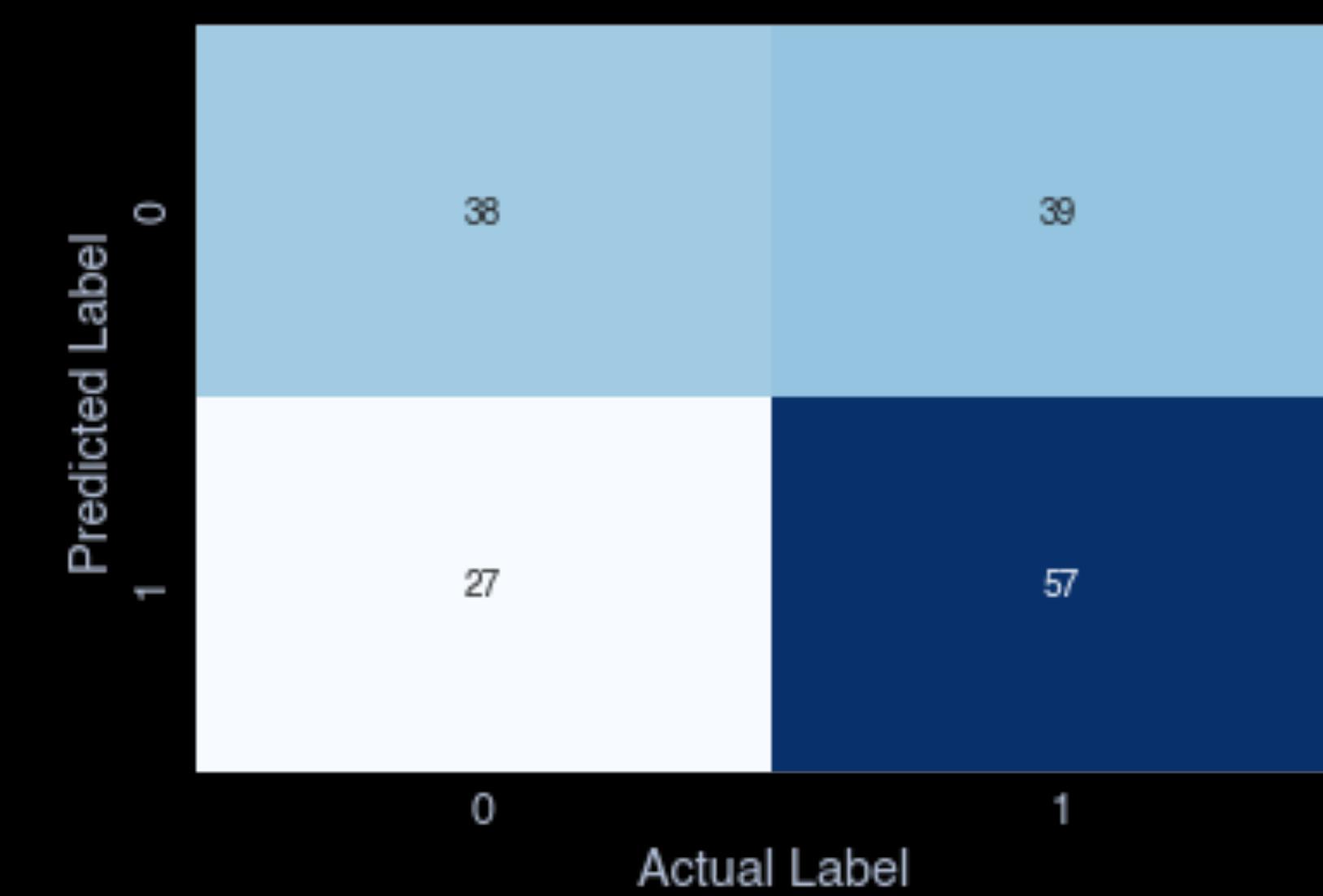
# Classifiers vs NN Models Comp

	train_accuracy	test_accuracy	ROC_AUC	test_recall	test_precision	F1	train_test_diff
Neural Network	0.523604	0.590062	0.586038	0.686	0.593718	0.632481	-0.066458
SVM_Recall Tune	0.539054	0.509317	0.482253	1.000	0.503145	0.669456	0.029737
Most Frequent	0.526953	0.496894	0.500000	1.000	0.496894	0.663900	0.030059
SVM_Balanced Tune	0.924092	0.571429	0.543210	0.838	0.544715	0.660099	0.352663
Neural Network Neg Tune	0.521405	0.478260	0.498917	0.020	0.503114	0.646110	0.043145

## SVM Balanced Tune



## NN Balanced Tune



# Recommendation

- Get moAr Data
- Use this model predictions in conjunction with a Multi-Factor

Trading model

# To do



- Spend more FE time and sourcing new data
- Try Regression Models or CNN/LTSM
- Backtest and Prove concept

#DSIdone



Thanks for listening!

# Sources

Reddit r/BTC

bitinfocharts.com

Google Trends

Yahoo Finance

Cryptory - Dashee87 Github

