Additional code written is all at the bottom:  
There should be a comment that looks like this:  
  
below which we have the new code

The new statistics to use for the write up should be the final final block, it should be below a comment that looks like this:  
  
  
In the new code, we essentially just re-pull the data and then we perform SMOTE to increase the amount of data we have (explained below), and then we just run the 3 new models.

So not much actual new code!

**Some explanations:**

**Synthetic Minority Oversampling Technique/SMOTE – additional preprocessing method used to get extra data (might not even need to mention this):**SMOTE is a method used to address the problem of imbalanced datasets (we have way less days with insider trades than non insider trades), where one class significantly outnumbers another. Instead of simply duplicating data from the minority class, SMOTE creates new synthetic examples by interpolating between existing minority samples. It selects a minority class (insider trades) sample and finds its nearest neighbors, then generates new data points along the line segments connecting these neighbors.

This approach helps balance the dataset and provides more variety in the synthetic samples, improving model performance without overfitting to repeated examples.

**Random Forest**

The Random Forest algo is an (ensemble) learning method that builds multiple decision trees together during training and aggregates their predictions to improve accuracy and control overfitting.

Each tree is constructed from a random subset of features and samples - so the model is robust to noise and variance.

In our experiment, Random Forest achieved an **accuracy of 93%**, with a perfect recall for class 1 (22/22) and a high precision for class 0 (16/19). This indicates that the model is highly effective at detecting positive cases – which are windows with insider trades (class 1).

**XGBoost**

XGBoost, (eXtreme Gradient Boosting), is a gradient boosting algorithm optimized for speed and performance.

It builds an ensemble of trees sequentially, where each tree focuses on correcting errors from the previous one – essentially it is therefore a self improving Random Forrest.

This model achieved the **highest accuracy of 95%** among the three models. It had a slight improvement over Random Forest in precision and recall, particularly for class 0 (17/19 correctly classified). This result shows XGBoost’s ability to minimize errors by iteratively refining predictions. The model's strong performance across all metrics makes it the most reliable option for insider trading prediction.

**LSTM**

LSTM involves an extra step of preprocessing where it takes different intervals as being sequential. This is how we are able to create time-series data.

Long Short-Term Memory (LSTM) model is a type of recurrent neural network (RNN) which is designed to capture temporal patterns and dependencies in sequential data – i.e. it can analyze and understand relationships between data points that occur in a specific order over time. In this experiment, the LSTM achieved an **accuracy of 80%**, which is lower than Random Forest and XGBoost.

While its performance was fairly balanced between classes, with recall values of 83% and 77% for classes 0 and 1, respectively, it struggled slightly with precision for class 0 and had a higher misclassification rate.

**ANALYSIS:**

**WHY LSTM lower:**

LSTM could potentially be worse at predictions because insider trades may not be easily predicted by the time when they occur, and are instead easier to predict if we know what the status of the stock is at the time (which is how Random Forrest and XGBoost works).

**What some of the statistics mean:**

**1. Precision**

* **Definition**: Precision measures like how many of the predicted positive cases are actually positive.
* **Formula**: A black text on a white background

  Description automatically generated
* **Interpretation**:
  + High precision means few false positives.
  + Low precision means the model is overpredicting positives.

**Scores**:

* **Random Forest**:
  + Precision for class 0 = 1.00 (perfect).
  + Precision for class 1 = 0.88 (very good, but a few false positives).
* **XGBoost**:
  + Precision for class 0 = 1.00 (perfect).
  + Precision for class 1 = 0.92 (excellent, slightly better than Random Forest).
* **LSTM**:
  + Precision for class 0 = 0.78 (lower, highlighting more false positives).
  + Precision for class 1 = 0.82 (decent but not as high as the other models).

**2. Recall**

* **Definition**: Recall measures how many of the actual positive cases the model correctly identified.
* **Formula**: A black text on a white background

  Description automatically generated
* **Interpretation**:
  + High recall means few false negatives.
  + Low recall indicates the model is missing positives.

**Scores**:

* **Random Forest**:
  + Recall for class 0 = 0.84 (some false negatives for class 0).
  + Recall for class 1 = 1.00 (perfect, no false negatives for class 1).
* **XGBoost**:
  + Recall for class 0 = 0.89 (better than Random Forest, fewer false negatives).
  + Recall for class 1 = 1.00 (perfect).
* **LSTM**:
  + Recall for class 0 = 0.83 (some false negatives).
  + Recall for class 1 = 0.77 (lower than the other models, missing more positive cases).

**3. F1-Score**

* **Definition**: F1-score is essentially a mix (or a harmonic mean) of precision and recall. It provides a balance between the two metrics and is especially useful when the class distribution is imbalanced.
* **Formula**:

A close-up of a sign

Description automatically generated

* **Interpretation**:
  + High F1-score indicates a good balance between precision and recall.
  + Low F1-score suggests poor performance in one or both metrics.

**Scores**:

* **Random Forest**:
  + F1-score for class 0 = 0.91 (strong balance).
  + F1-score for class 1 = 0.94 (excellent).
* **XGBoost**:
  + F1-score for class 0 = 0.94 (better than Random Forest).
  + F1-score for class 1 = 0.96 (highest among the models).
* **LSTM**:
  + F1-score for class 0 = 0.81 (decent but lower than tree-based models).
  + F1-score for class 1 = 0.79 (lower, showing weaker performance).

**4. Support**

* **Definition**: Support is the number of actual occurrences of each class in the test dataset. It provides context for the evaluation metrics.
* **Interpretation**:
  + Larger support means the metric is more robust (based on more samples).
  + Smaller support can lead to less reliable metrics due to fewer samples.

**Scores**:

* **Random Forest**: Support for class 0 = 19, class 1 = 22. Both classes are fairly balanced, allowing for reliable metrics.
* **XGBoost**: Same as Random Forest (support is the same for all models).
* **LSTM**: Support for both classes = 30 each. With a larger dataset, this model might perform better, but currently, it lags behind.

**Summary of Model Results:**

1. **Random Forest**:
   * Strong overall performance with good balance between precision and recall.
   * Slightly lower recall for class 0, meaning it misses a few negative cases.
   * A solid choice but slightly behind XGBoost.
2. **XGBoost**:
   * Best performer overall with the highest precision, recall, and F1-scores for both classes.
   * Particularly excels in handling class 0 better than Random Forest.
   * Ideal model for insider trading pred.
3. **LSTM**:
   * Performs reasonably well but struggles to match the tree-based models.
   * Lower precision and recall indicate room for improvement, potentially with more data or hyperparameter tuning.
   * May still be useful for capturing time based/temporal patterns, but not optimal in this specific setup as described above.