GROUP WORK PROJECT # _3_
Group Number: ____7073________

MScFE 600: FINANCIAL DATA

FULL LEGAL NAME	LOCATION (COUNTRY)	EMAIL ADDRESS	MARK X FOR ANY NON-CONTRIBUTING MEMBER
Lawal Nimotalai Abduganiyu	Nigeria	nabduganiyu@gmail.com	
Shivansh Kumar	India	Shivansh.business23@gmail.com	
Pulkit Gaur	India	pulkit.gaur.iit@gmail.com	

Statement of integrity: By typing the names of all group members in the text boxes below, you confirm that the assignment submitted is original work produced by the group (excluding any non-contributing members identified with an "X" above).

Team member 1	**Lawal Nimotalai Abduganiyu**
Team member 2	**Shivansh Kumar**
Team member 3	**Pulkit Gaur**

Use the box below to explain any attempts to reach out to a non-contributing member. Type (N/A) if all members contributed.
Note: You may be required to provide proof of your outreach to non-contributing members upon request.

N/A

Issues

- Issue 1: Optimizing Hyperparameters
- Issue 2: Optimizing the Bias-Variance Tradeoff
- Issue 3: Applying Ensemble Learning—Bagging, Boosting, or Stacking

Step 2

Issue 1: Optimizing Hyperparameters

Technical Section:

- **Model Technicalities**
    - **Model Names**
    - **List of Hyperparameters**
    - **Importance of Each Hyperparameter**
- **Hyperparameter Optimization Techniques**
    - **Grid Search**
    - **Random Search**
    - **Bayesian Optimization**
- **Performance Metrics**
    - **Evaluation Metrics**
        - **Accuracy**
        - **Precision**
        - **Recall**
        - **F1-Score**
        - **ROC-AUC**
    - **Results**
- **Validation Strategies**
    - **Cross-Validation**
- **Determining Optimality**
    - **Comparative Analysis**
- **Interpretations and Recommendations**
    - **Interpretations**
    - **Recommendations**

Non-Technical Section:

- **Stakeholder Communication**
  - **Explanation of Hyperparameters**
  - **Optimization Process Overview**
  - **Validation and Optimality Criteria**
- **Risk Management**
  - **Identifying Risks in Optimization**
  - **Mitigation Strategies**
  - **Contingency Plans**
- **Ethical Considerations**
  - **Data Privacy in Optimization**
  - **Bias and Fairness in Results**
  - **Compliance with Regulations**
- **Business Impact Analysis**
  - **ROI Calculation for Optimization**
  - **Strategic Alignment with Business Goals**
  - **Long-term Vision for Model Performance**

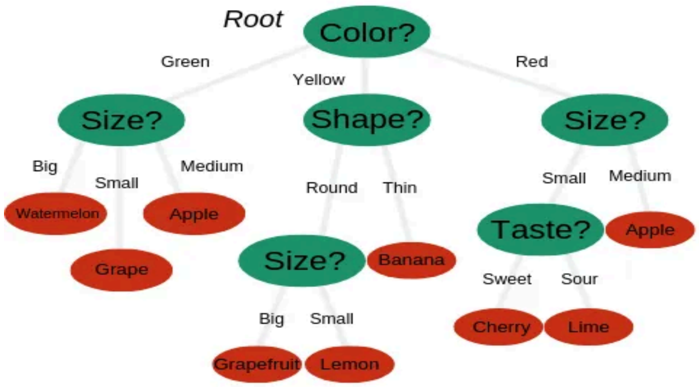
Step 3

1. Optimizing Hyperparameters

- ## Technical Section
  - Model Technicalities
    - Model Name: Random forest

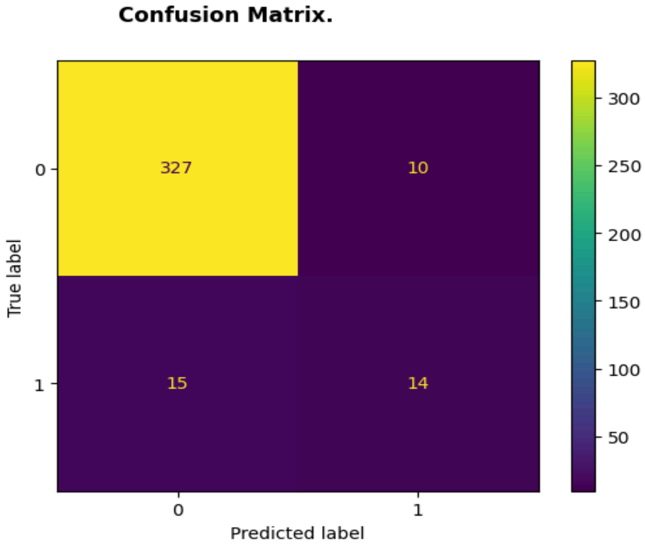
      - For our implementation, we are using a Random forest model. The random forest model is an supervised learning algorithm that includes multiple decision trees in which each tree has the same nodes but uses different data that leads to other leaves. It composes all the decisions of multiple decision trees together to solve the given problem which is the average of all the decision trees. It uses labeled data to learn and solve classification problems from its learning. (Schott)

      - Here in the figure below, we can see how multiple decision trees are used to classify different fruits. (Schott)



  - List of Hyperparameter and it’s Importance (Arya)
    - n_estimators: Controls the ensemble size and generalization ability.
    - max_depth: Influences the model's complexity and potential overfitting.
    - min_samples_split: Affects the granularity of the splits, balancing between underfitting and overfitting.
    - min_samples_leaf: Ensures that each leaf has a sufficient number of samples, preventing overfitting.
    - max_features: Determines the diversity of the trees, influencing both bias and variance.
    - criterion: Determines the function used to measure the quality of a split, influencing the tree's structure and performance
      - In Random Forest we have 2 criterion Gini and entropy. (Schott)
        - Gini : $1 - \sum_{i=1}^{C} (p_i)^2$
        - Entropy = $\sum_{i=1}^{C} - p_i * log_2(p_i)$
- Hyperparameter Optimization Techniques
  - Grid Search
    - Grid Search involves exhaustively searching through a specified subset of hyperparameters. For each combination, the model is trained and evaluated. The combination with the best performance is selected.
  - Random Search
    - Random Search samples hyperparameters from a distribution. It is less computationally expensive than Grid Search and can often find good hyperparameters faster.
  - Bayesian Optimization
    - Bayesian Optimization uses a probabilistic model to predict the performance of different hyperparameter settings. It iteratively updates the model to focus on promising regions of the hyperparameter space.
- Performance Metrics
  - Evaluation Metrics
    - Accuracy : Proportion of correctly classified instances.
      - Accuracy = Number of correct prediction / Total Number of Predection
    - Precision : Proportion of true positive predictions among all positive predictions.
      - Precision = True Positives / (True Positives + False Positives)
    - Recall : Proportion of true positive predictions among all actual positives
      - Precision = True Positives / True Positives + False Negatives

- F1-Score : Harmonic mean of Precision and Recall.
  - F1-Score = 2 X $(precision\ *\ Recall\ /\ Precision\ +\ Recall)$
- ROC-AUC : Area under the Receiver Operating Characteristic curve, measuring the model's ability to distinguish between classes.
  - ROC-AUC = $\int_{0}^{1} TPR(FPR)dFPR$

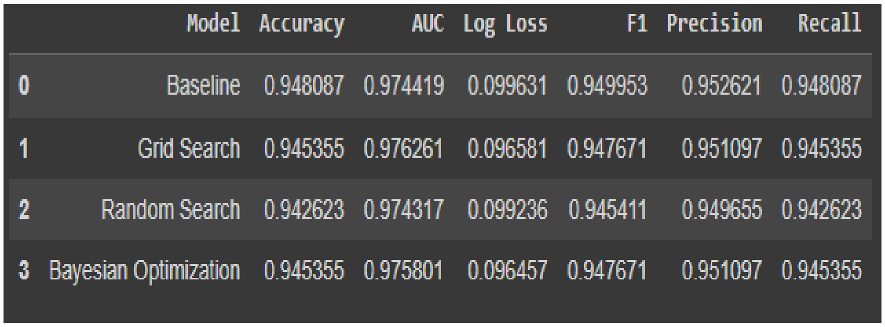
- Validation Strategies
  - Cross-Validation: A technique where the dataset is split into k folds, and the model is trained and validated k times, each time using a different fold as the validation set.
    - CV = $1/k \sum_{i=1}^{k} Performance\ Metric_i$
- Determining Optimality
  - Comparative Analysis
    - Confusion Matrix Analysis
      - A confusion matrix is a table used to describe the performance of a classification model on a set of data for which the true values are known. It provides a detailed breakdown of the model's predictions versus the actual outcomes.
      - 

- Implementation
  - Interpretations & Results
    - Steps of implementation:
      - Feature engineering: We started with collection 5 years data of BTC-USD daily OHLCV from 2019-2024, and done some feature engineering buy calculating some technical indicator and a bullish strategy as our goal is Price Movement Classification to predict whether a cryptocurrency's price (e.g., Bitcoin) will move

up(1) or down(0) based on historical price data and technical indicators.

- ○ Data preprocessing: After all this we make our data ready by adding the target variable and doing some interpolation for some missing data.
- ○ Training Baseline model: As we are doing classification doing Random Forest classifier we first need to train a baseline model with default parameters for comparative analysis with other model with optimized hyperparameters.
- ○ Multiple Optimized model: we then trained multiple hyperparameter optimized models using Grid Search, Random Search, Bayesian optimization.
- ○ Evaluation performance: We then use cross-validation to evaluate the model's performance on each set of hyperparameters.
- ○ Confusion Matrix Analysis: For each set of hyperparameters, compute the confusion matrix and derive the performance metrics.
- ○ Selecting best optimized model: At last we choose the best hyperparameter optmized model and method based on our decided performance metrics (i.e. Accuracy, Precision, Recall, F-1 Score, ROC-AUC, Log loss).

- Results



	Model	Accuracy	AUC	Log Loss	F1	Precision	Recall
0	Baseline	0.948087	0.974419	0.099631	0.949953	0.952621	0.948087
1	Grid Search	0.945355	0.976261	0.096581	0.947671	0.951097	0.945355
2	Random Search	0.942623	0.974317	0.099236	0.945411	0.949655	0.942623
3	Bayesian Optimization	0.945355	0.975801	0.096457	0.947671	0.951097	0.945355

- Interpretations
    - The results indicate that the baseline model performs slightly better than the optimized models across most metrics, with an accuracy of 0.948087, AUC of 0.974419, and F1 score of 0.949953. Both Grid Search and Bayesian Optimization models achieve similar performance, with accuracies of 0.945355 and AUCs of 0.976261 and 0.975801, respectively. The Random Search model lags behind with an accuracy of 0.942623 and a slightly lower AUC of 0.974317.however, the log loss is lowest for the Grid Search and Bayesian Optimization models, suggesting better generalization. Overall, while the optimized models show marginal improvements in certain metrics, the baseline model remains competitive and can be preferred for its simplicity and comparable performance.

Non-Technical Section:

- Stakeholder Communication
    - Explanation of Hyperparameters:
        - Hyperparameters are configuration settings that are not learned from the data but are set prior to training the model. In the context of a Random Forest, common hyperparameters include:
        - Number of Trees (n_estimators): The total number of decision trees in the forest.
        - Maximum Depth (max_depth): The maximum depth of each tree, which controls the complexity of the model.
        - Minimum Samples Split (min_samples_split): The minimum number of samples required to split an internal node.
        - Minimum Samples Leaf (min_samples_leaf): The minimum number of samples required to be at a leaf node.
        - Maximum Features (max_features): The number of features to consider when looking for the best split.
    - Optimization Process Overview:
        - The optimization process involves systematically searching for the best combination of hyperparameters that maximize the model's performance. This is typically done using techniques like Grid Search, Random Search, or Bayesian Optimization. The process includes:
        - Defining the Search Space: Specifying the range of values for each hyperparameter.

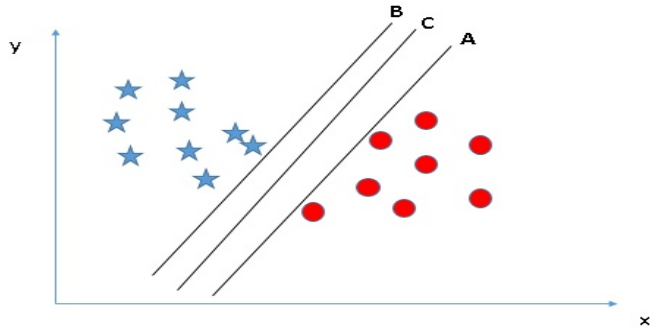
        - Selecting an Optimization Technique: Choosing the method to explore the hyperparameter space like Grid Search, Random Search, bayesian optimization.
        - Training and Validation: Running the model with different hyperparameter combinations and evaluating performance using a validation set.
        - Selecting the Best Configuration: Identifying the hyperparameter set that yields the highest performance metric (i.e., accuracy, F1 score,precision, Recall etc).
    - Validation and Optimality Criteria:
        - Cross-Validation: Splitting the data into multiple folds and training the model on different subsets.
        - Optimality criteria typically include metrics such as accuracy, precision, recall, F1 score, and ROC-AUC. The goal is to find a balance between model complexity and performance, avoiding overfitting while maximizing predictive power.
- Risk Management
    - Risks in Optimization
        - Overfitting: The model performs well on training data but poorly on unseen data.
        - Underfitting: The model is too simplistic and fails to capture the underlying patterns.
        - Resource Intensive: The optimization process may require significant computational resources and time.
        - Bias in Hyperparameter Selection: The chosen hyperparameters may introduce bias, leading to unfair or biased results.
    - Mitigation Strategies
        - Regularization Techniques: Use techniques like early stopping or pruning to prevent overfitting.
        - Data Augmentation: Increase the diversity of the training data to improve generalization.
        - Parallel Processing: Utilize parallel computing to speed up the optimization process.
        - Bias Detection: Regularly assess the model for bias and fairness issues.
    - Contingency Plans
        - Backup Plans: Maintain backups of the model and data in case of failures.
        - Alternative Hyperparameters: Have a set of alternative hyperparameters ready to switch to if the primary optimization fails.
        - Continuous Monitoring: Implement monitoring tools to track model performance and detect anomalies.

- Ethical Considerations
  - Data Privacy in Optimization
    - We need to ensure that the optimization process complies with data privacy regulations (e.g., GDPR, CCPA). This includes anonymizing data, obtaining necessary consent, and ensuring secure storage and transmission of data.
  - Bias and Fairness in Results
    - Bias Detection: Use fairness metrics to identify and mitigate bias in the model's predictions.
    - Fairness Audits: Conduct regular audits to ensure that the model treats all groups fairly.
    - Transparency: Be transparent about the model's limitations and potential biases.
- Business Impact Analysis
  - ROI Calculation for Optimization
    - Cost Analysis: we still nedd to estimating the costs associated with data collection, model training, and optimization.
    - Benefit Analysis: after that we need to quantify the benefits, such as improved accuracy, reduced errors, and increased efficiency.
    - ROI Calculation: As we have just built an toy model we have not considered ROI but comparing the costs to the benefits can determine the overall ROI.
  - Strategic Alignment with Business Goals
    - Business Objectives: As our main objective is to use the models with right parameters, we have tried to answer all the issues addressed by our strategists like How do we know our models have right parameter, which hayperparameters are used, How have we optimized them and how we know they are optimal?
    - Stakeholder Input: We have tried to solve every issues but still valuable feedback from stakeholders are much need to incorporate it in our business needs.
  - Long-term Vision for Model Performance
    - Long-term vision: We need to also consider the long term impact of these kind of optimization approach in our investment models and how they can impact our PnL.
    - We also need to work on regularly update and refine the model to adapt the change in micro to macro environment also we need to make sure how we are going to scale it on a larger scale.

- How can ML tools add value?
    - ML is not a magic potion but a tool that we need to leverage for solving the hyperparameter optimization issue, Optimizing the hyperparameter and using the optimal parameters can significantly enhance the accuracy and robustness of predictive models, thereby adding substantial value. By fine-tuning hyperparameters in models like Random Forests, we can improve the precision of their forecasts, reduce prediction errors, and better manage risk. For instance, optimized models can more accurately predict market trends, identify profitable investment opportunities, and detect early warning signs of market downturns. This leads to more informed decision-making, potentially increasing returns and minimizing losses. Additionally, the ability to systematically explore and select the best hyperparameters can streamline the model development process, making it more efficient and scalable. Ultimately, the integration of advanced ML tools in hyperparameter optimization can provide a competitive edge, enabling investors to stay ahead in a dynamic and complex financial landscape.

2. Optimizing the Bias-Variance Tradeoff

● Technical Section

- Model Technicalities
  - Model Name: Support Vector Machine (SVM)
    - For our implementation, we are using an SVM model. The SVM model is a supervised learning algorithm which is mostly used for classification tasks. It tries to find a hyperplane which best separates different classes in the feature space. SVM allows both linear and non- linear data by using kernel functions which can transform data into higher dimensional spaces.
    - Here in the figure below, we can see how SVM is classifying different kinds of data using hyperplanes
    - 

  - Bias Variance Tradeoff:
    - **Bias:** When the model is too simple and actually can't fit the data properly. It will fail to capture important patterns
    - **Variance:** When the model is too complex, it fits the data too closely and will also fit the noise in the data, leading to overfitting and poor accuracy.
- **Bias-Variance Tradeoff in SVM**
  - High Bias (Underfitting): The SVM model is using a large margin and using simple linear boundaries, so it's not able to properly capture and classify the data
  - High Variance (Overfitting): The SVM model is using a very small margin or is using non-linear boundaries that are highly sensitive to the training data as well as the noise present

- **Kernel Trick and its impact on Bias and Variance:**
    - **Linear Kernel**: This is used for data which is linearly separable. Has low variance and high bias
    - **RBF (Radial Basis Function) Kernel:** This is useful for non-linear data but may overfit sometimes. It has medium to high variance and low bias.
    - **Polynomial Kernel:** It controls its complexity with the degree of the polynomial that its using. It has medium variance and medium bias.
- Determining Optimality
    - Optimizing the Hyperparameters that influence Bias and Variance in SVM
        - **Regularization Parameter(C):** Low C increases bias and higher C increases variance
        - **Kernel Type:** By changing, the kernel type, we can determine which kernel fits our data the best
        - **Kernel Coefficient for RBF/Polynomial:** It will control the influence of individual data point
    - K-fold Cross Validation:
        - We will split our dataset into k folds and the train on k-1 folds and test on remaining data. Thus we will be able to find the best option which has low variance in performance across every fold which suggests the optimal bias-variance tradeoff.
    - Learning Curves:
        - Learning curves will help us visualize how model's performance changes as the data size increases
    - Bayesian Optimization:
        - We can build a probabilistic model using Bayesian Optimization to focus on the promising regions of hyperparameter space and it will reduce both bias and variance efficiently.

- Model Complexity vs. Generalization:
    - Low Bias and Low Variance: After using the above techniques to optimize bias-variance tradeoff. We should be able to find the optimal model that works well on both training and testing data.

Non-Technical Section:

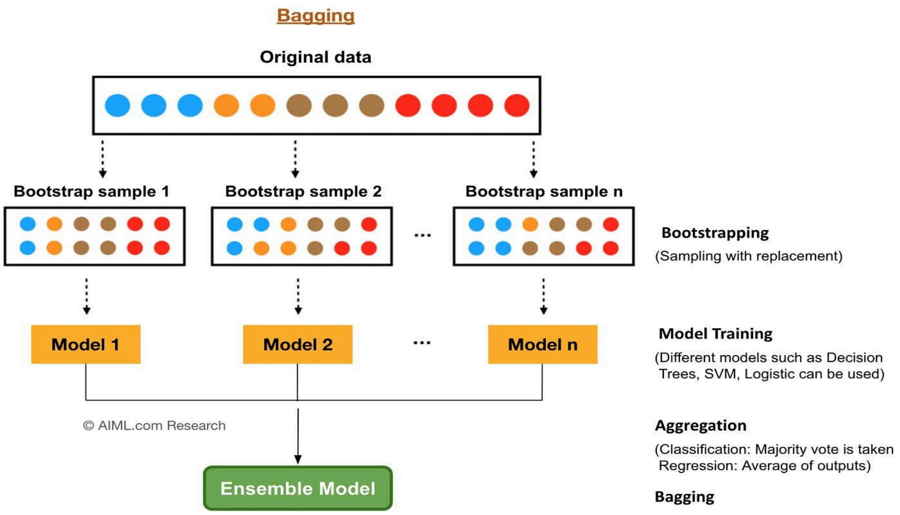
- Stakeholder Communication
  - Bias-Variance Tradeoff:
    - Bias happens when the model makes overly simple assumptions and then underperforms on both the training and the test data
    - Variance happens when the model is very sensitive to the smallest variations and noises in the training data. Thus, it can't generalize to newer data.
  - Hyperparameters and their impact:
    - C: Controls how closely the model fits the data. Smaller C means higher bias and larger C will mean lower bias and higher variance
- Risk Management
  - Risks in Bias-Variance Optimization
    - Overfitting: A complex SVM model will perform well on training data but will not work that well on test data
    - Underfitting: A simple SVM model will fail to capture important patterns and won't be as accurate
    - Computational Cost: Complex kernels and large value of C might make our SVM model computationally expensive.
  - Mitigation Strategies
    - Regularization Techniques: Adjust parameter C such that it will control margin size, generalization and will help balance complexity.
    - Cross-Validation: Reduces variance by making sure that the model performs well on different subsets of data
    - Early Stopping: We can use it to prevent overfitting during training by stopping it when the performance begins to degrade
- Ethical Considerations
  - Data Privacy in Optimization
    - We need to ensure that the optimization process complies with data privacy regulations (e.g., GDPR, CCPA). This includes anonymizing data, obtaining necessary consent, and ensuring secure storage and transmission of data.
  - Bias and Fairness in Results
    - Bias Detection: Use fairness metrics to identify and mitigate bias in the model's predictions.
    - Fairness Audits: Conduct regular audits to ensure that the model treats all groups fairly.
    - Transparency: Be transparent about the model's limitations and potential biases.

- Business Impact Analysis
    - ROI Calculation for Optimization
        - Cost Analysis: we still need to estimate the costs associated with data collection, model training, and optimization.
        - Benefit Analysis: after that we need to quantify the benefits, such as improved accuracy, reduced errors, and increased efficiency.
        - ROI Calculation: As we have just built a toy model we have not considered ROI but comparing the costs to the benefits can determine the overall ROI.
    - Strategic Alignment with Business Goals
        - Business Objectives: As our main objective is to use the models with right parameters, we have tried to answer all the issues addressed by our strategists like How do we know our models have the right parameter, which optimize the bias-variance tradeoff.
        - Stakeholder Input: We have tried to solve every issue but still valuable feedback from stakeholders is much needed to incorporate it in our business needs.
    - Long-term Vision for Model Performance
        - Long-term vision: We need to also consider the long term impact of this kind of optimization approach in our investment models and how they can impact our PnL.
        - We also need to regularly update and refine the model to adapt the change in micro to macro environment. We also need to make sure how we are going to scale it on a larger scale.

- How can ML tools add value?
    - ML for optimizing the bias-variance tradeoff can help us to do enhanced decision-making by predicting outcomes, identifying trends and optimizing the whole model to best predict the data that we are dealing with. By assessing the predictions, we can make informed decisions and better manage the risk associated with our financial investments and portfolios. SVM's ability to handle both linear and non linear data with kernel functions, combining with proper tuning of the C parameter will allow financial analysts to model complex and nonlinear behavior in the asset prices in various market conditions as well as it will allow them to capture subtle patterns in the data that simpler tools might not be able to capture. Thus, ML can help with portfolio management, risk assessment and algorithmic trading.

3. Applying Ensemble Learning - Bagging

- ## Technical Section
  - **Model Technicalities**
    - **Model Name: Bagging**
      - Bagging also known as bootstrap aggregating is a machine-learning ensemble method used to combine multiple models. The algorithm involves creating multiple subsets of the training dataset through bootstrapping (i.e random sampling with replacement), a model is then trained on each subset and then an aggregate of the models' predictions is taken. The aggregate could be chosen to be the average or the majority vote of the models' predictions.
      - An illustration of the bagging algorithm is shown with the following:



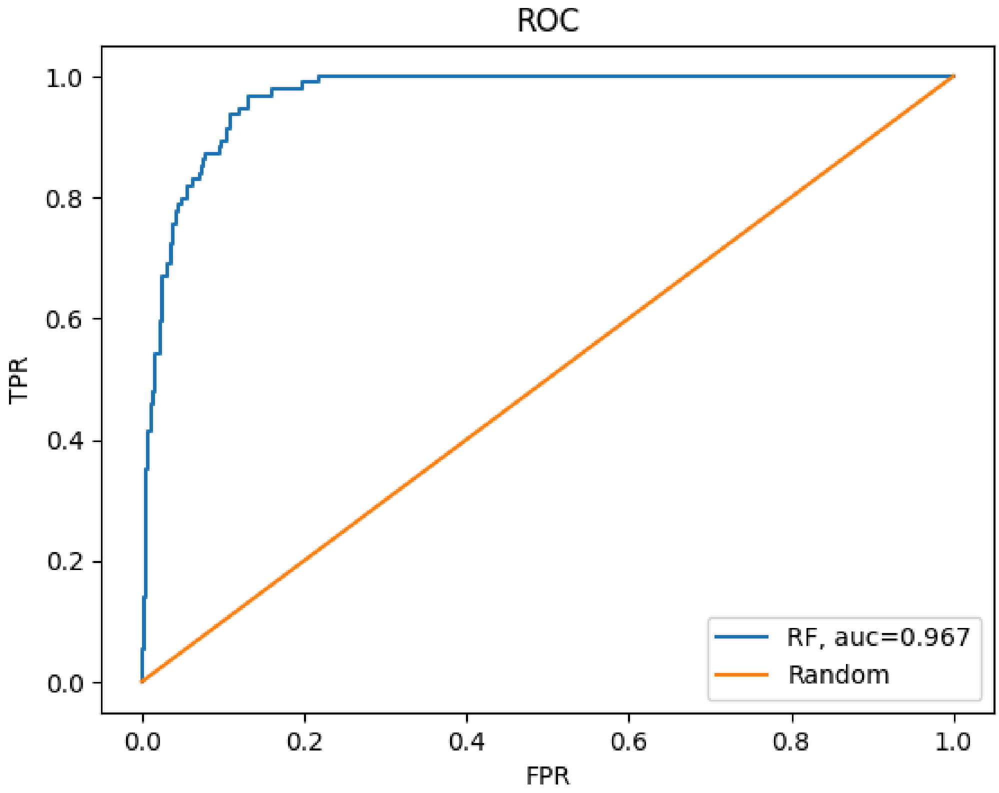
  - **List of Hyperparameters and their importance**
    - **The choice of the base model:** We are going to implement bagging with the random forest classifier as the base model.
    - **Number of trees:** A larger number of trees will generally improve performance but might make the model more computationally costly.
    - **Number of sample splits at each node:** This determines the diversity of the models created. The larger the number of splits, the more diverse the model and the lesser the risk of overfitting.
    - **Maximum number of features:** The number of features randomly selected at each node during tree construction. A small number of features reduces the risk of overfitting but this can affect the performance of the model if the number of features is too little.

        - **Aggregation method:** This determines how the individual decision trees are combined. As mentioned above, it could be an average or a majority vote. Our choice is the majority vote.

- **Hyperparameter Optimization Techniques**
    - **Grid Search:** This is an optimization technique that involves defining a grid of values for the hyperparameters and searching through all possible combinations of the hyperparameters to determine the optimal values of hyperparameters.

- **Performance Metrics**
    - **Evaluation Metrics**
      Since our problem is a classification problem, we will use the following metrics to evaluate the model's performance:
        - **Accuracy:** The proportion of predictions that were correct out of all the predictions.
        - **Precision:** The proportion of positive predictions that are truly positive.
        - **Recall:** The proportion of actual positive instances that were correctly predicted as positive.
        - **F1-Score:** The harmonic mean of precision and recall.
        - **ROC-AUC:** We can plot the Receiver Operating Characteristic (ROC) curve and the Area Under the Curve (AUC).
        - **Confusion matrix:** Summarizes the performance of the model.
        - **Out-of--bag score:** It is used to estimate the generalization error of the bagging method. The OOB score takes values between 0 and 1.
          For each sample, we use the subset of trees that did not include that sample to make a prediction. We calculate the errors of the predictions and then average the errors over all samples to obtain the OOB score.

    - **Results:**
        - Accuracy of the model after hyperparameter tuning and optimization increased from 92.29% to about 93%
        - Precision score shows that Class 0 (indicating returns less than 2%) is correctly predicted by the model 96% of the time while Class (indicating returns higher than 2%) is correctly predicted by the model 77% of the time.
        - The F1-score (indicating balance between prediction and recall) shows that Class 0 (at 96%) performs better than class 1 (at 78%).
        - Overall performance shows that the model is better at predicting instances where the returns on NVDA stocks are less than 2%.



- The ROC curve indicates that the bagging model with random forest classifier has more prediction benefits over a random model.

Non-Technical Section

- **Stakeholder communication**
  - **Model Combination Benefits (using bagging):**
    - Combining many models reduces the variance of predictions from the resulting model increasing its accuracy.
    - Using ensembles makes the resulting model less biased as it combines different strengths and weaknesses.
    - Bagging can improve model generalization by creating a more diverse set of models, which allows it to better capture the underlying patterns in the data.
    - Bagging can help to prevent overfitting because the models are created on different subsets of the dataset ensuring it to better generalize to new data.
  - **Explanation of hyperparameters and their impacts:**
    - **Number of trees:** This is the number of individual models created. In our case, we started with 10 trees. The more trees, the better the predictions. However, too many trees may increase the time of computation without adding significant improvements. After optimization, we got 20 as the optimal number of trees.
    - **Maximum Features**: This controls how many features each tree can use when making decisions. Limiting the number of features ensures that each tree can

focus on different aspects of the data. This can enhance diversity and improve overall predictions. Using too many features in each tree can cause them to be similar and reduce diversity of the model.

Tuning the maximum features hyperparameter for our data resulted in 6 features as the best estimate.

- **Sample Size:** THis determines the number of data points to be included in each tree. Using too few data points will make the trees unable to capture some of the patterns in the data. Too many data points per tree will make the trees too similar.

  After optimization, we determined the optimal sample split to be 4 samples which represents (2982/4=) about 745 data points per sample.

It is important to optimize the model by determining the optimal values of the hyperparameters.

- **Risk Management**
  - **Identifying challenges in model combination using bagging**
    - **Overfitting risk:** If the individual models capture noise or are too complex the resulting model may not handle new data accurately. The random forest classifier is a pretty simple model and is less subject to this challenge.
    - **Risk of model similarity:** The individual models should be diverse. Similar models (using similar data samples), diminishes the benefits of model combination.
    - **Hyperparameter tuning:** Tuning hyperparameters for ensemble models can be complex and may require more computational cost.
    - **Dependency on choice of base model:** The results of bagging relies heavily on the chosen base model
    - Quality of data: Bagging relies heavily on the data used. Erroneous predictions from a noisy or biased dataset can be made worse with bagging.

  - **Mitigation Strategies for the Challenges**
    - **Limiting model complexity:** Using simpler base models keeps the ensemble from being too complex.
    - **Controlling the number of trees:** Starting with a small number of trees and increasing gradually to arrive at the optimal value.
    - **Increase Model Diversity**: Using different base models (e.g., combining trees with other algorithms) or adjusting the features used by each model can enhance diversity and improve the resulting model's overall performance.
    - **Hyperparameter Tuning**: Use techniques like grid search to find the best hyperparameters for both the ensemble and the individual models
    - **Improve Data Quality**: Data cleaning and preprocessing to reduce noise and bias.

- **Ethical Considerations**
    - **Data Privacy**
        - **Secure data storage:** Storing training data securely and preventing unauthorized access to data to protect sensitive information that can be accessed by business competition.
        - Regular data audits should be conducted to ensure compliance with data protection regulations like GDPR or CCPA.
        - Monitoring the model predictions for any signs that they might reveal sensitive information.
    - **Bias and Fairness in Results**
        - **Training Data Diversity:** Using training data that is diverse and representative can help prevent bias
        - **Sampling bias:** The bootstrap sampling process may favor certain data points over others if the dataset is too small or unbalanced leading to some data points being overrepresented while others are underrepresented.
- **Business Impact Analysis**
    - **Cost implications, efficiency and added business value**
        - Investment in the bagging algorithm for business decision making involves incurring additional costs such as computational cost, data preparation costs, development time, monitoring and maintenance costs.
        - However, the benefits of more informed and more efficient decisions may well outweigh these costs especially in the long-term.
        - Improved prediction accuracy will help the business to be more confident and make faster decisions that will yield better returns and result in added business value.
        - Higher predictive accuracy and reliability can lead to better business outcomes, such as improved customer retention, more effective marketing campaigns, and enhanced product recommendations.
        - Generally, specific business needs, alignment with business strategy, data availability, resources and interpretability should be main considerations when using bagging for business decisions.
- **How ML can add value**
    - **ML** models are versatile and can be applied to different fields such as finance, health, governance and so on. Bagging which has diverse choices of base models offers a lot of flexibility in application.
    - Bagging can handle large datasets and allows the model to be scaled up as the organization expands.
    - Reducing error in predictions can help manage business risks associated with decision making such as customer credit rating, taking investment positions and portfolio management.
    - More accurate models provide insights into market trends and aid strategic decision making.

Step 4

Student Reviews:-

Student	**Wrote**	**Reviewed**
Team member A	**Issue 3**	**Issue 2**
Team member B	**Issue 1**	**Issue 3**
Team member C	**Issue 2**	**Issue 1**

Team member B (Issue 3 review):

- Accuracy and technical Correctness:
  - Issue 3 is written by Team Member A in which the Bagging model is used to implement the ensemble learning model, and the member well explains the model description. The concept of creating multiple subsets of the training dataset through bootstrapping and aggregating the predictions is correctly outlined.
  - The next Important thing which is hyperparameters for the bagging model especially when using a Random Forest as the base model is clearly explained.
  - The team members have also used hyperparameter techniques like grid search which is accurate and relevant for hyperparameter tuning.
  - The choice of evaluation metrics is very comprehensive and suitable for a classification problem. The explanation of the Out-of-Bag score is accurate.
- Clarity And Readability
  - The text is clear and easy to understand and all the concepts like bagging, performance metrics etc are well articulated.
  - Both the technical and non-technical parts structure are logically explained one after another forming an intuitive separation.

- Spelling & Grammar
  - There are no spelling errors as such in the text.
  - And the grammar is also mostly correct in most of the part with some minor issues like The phrase "the lesser the risk of overfitting" could be rephrased for clarity. A more concise version could be "the lower the risk of overfitting."
- Suggestions for improvement

    - Clarity: Some paraphrasing can be done to improve clarity and make it a publication level report.
    - Consistency: consistency in the use of terms like "model" and "classifier" throughout the text. For example, if you refer to the base model as a "classifier" in one place, maintain that term consistently.
    - Detailed Explanation: While the explanations are generally good, adding a brief example or analogy for concepts like bootstrapping or the Out-of-Bag score could enhance understanding for readers who are new to these concepts.

Team member A (Issue 2 review)

- Accuracy and technical Correctness:
    - Issue 2 is written by Team Member C in which the Support Vector Machine model is used to optimize bias-variance tradeoff. The member provided an accurate definition and well-explained model description. The concept of achieving the optimal hyperplane that correctly separates the data into two different classes and dimension reduction were well explained
    - The model hyperparameters were listed with their importance and impacts well outlined.
    - Team member C also used hyperparameter tuning techniques such as grid search, random search and Bayesian optimization to obtain best estimates for the hyperparameters.
    - The choice of evaluation metrics is very comprehensive and suitable for a classification problem.
    - The team member used illustrative diagrams and gave concise summary and interpretation of results
- Clarity And Readability
    - Both the technical and non-technical parts structure are logically explained one after another forming an intuitive separation.

- Spelling & Grammar
    - Some spelling and grammatical errors were identified and corrected accordingly
- Suggestions for improvement
    - Clarity: Some paraphrasing can be done to improve clarity and make it a publication level report.
    - More reference to the modelled data: In interpreting results, more reference should be made to the data used for the modelling exercise to draw more practical inferences from the reports.

Team Member C (Issue 1 review)

- Accuracy and technical Correctness:
  - Issue 1 is Written by Team B, the model description of random forest is correct and the explanation of how the Random Forest model works, including the use of multiple decision trees and aggregation of their decisions, is accurate.
  - The list of Hyperparamters and it's imporatnce is also clear and explained properly.
  - The Optimization techniques mentioned like Grid Search, Random Search, and Bayesian Optimization are relevant for hyperparameter tuning.
  - Again the performance metrics chosen is also correct and explained well as we all have chosen the same performance metrics.
  - Also Team member B have also provided implementation steps as well as result interpretations which make the whole issue easy to understand.
- Clarity And Readability
  - The language used by Team member B is easy to understand not only for technical bu also for Non-technical members.
  - As I already mentioned the structure of the explanation is very clear explained and the the necessary information is provides for both technical and non-technicla members.
- Spelling and Grammar
  - I have not found any spelling and grammatical error.
- Suggestions for improvement
  - Some sentences can be paraphrased to be more precise and clear.
  - Some generic examples can be given to explain some concepts.

References

1. Arya, Nisha. “Tuning Random Forest Hyperparameters.” *KDnuggets*, 22 August 2022, https://www.kdnuggets.com/2022/08/tuning-random-forest-hyperparameters.html. Accessed 15 October 2024.
2. Schott, Madison. “Random Forest Algorithm for Machine Learning | by Madison Schott | Capital One Tech.” *Medium*, 25 April 2019, https://medium.com/capital-one-tech/random-forest-algorithm-for-machine-learning-c4b2c8cc9feb. Accessed 15 October 2024.