Executive Summary

This study aimed to predict how frequently individuals wear masks in public settings using survey responses collected from students during the early stages of the COVID-19 pandemic. Each participant answered over 150 questions related to their beliefs, habits, and perceptions surrounding mask-wearing, social norms, and health behaviors.

To interpret the data, we first cleaned and standardized the responses, replacing missing values with averages or most common answers. The main outcome we aimed to predict was how often someone wears a mask: *rarely*, *most of the time*, or *always*.

After evaluating multiple models—including logistic regression, decision trees, and others—we selected **Random Forest** as our final model due to its strong accuracy and adaptability. Random Forests are especially flexible: they build many decision trees and combine their outputs, making them powerful even when the data is complex and includes many survey questions. We fine-tuned this model to increase accuracy using smart search techniques. The best-performing version used smaller trees and limited the number of features considered at each decision point.

Our findings suggest that **perceptions of others' mask-wearing behavior**, **personal beliefs about safety**, and **social norms** were the strongest predictors of an individual's own mask usage. Specifically, individuals who believe their peers always wear masks or that mask-wearing is expected by their community are more likely to report "always" wearing a mask themselves.

As a baseline, consider a typical student who thinks masks are helpful, believes their friends usually wear masks, and agrees that their community values safety. This student is expected to wear a mask *most of the time*, with similar students showing a wide range in behavior—from rarely to always—depending on how strong those beliefs are.

The top 5% most mask-compliant individuals tend to strongly agree that mask-wearing is both effective and socially expected, and they report high personal control over their decisions. On the other hand, students with moderate beliefs who still value comfort or convenience may be found wearing masks "most of the time," reflecting a trade-off.

Overall, the model helps us understand how a variety of beliefs and perceptions influence behavior. If you're trying to identify students who are most likely to wear masks consistently—or those who might benefit from targeted communication—this model can help spotlight the key beliefs that matter most.

Model Selection and Hyperparameter Optimization

1 Methodology

1.1 Data Preprocessing

The raw dataset contained both numerical and categorical variables, some of which included missing values. A custom imputation strategy was applied:

- **Numerical columns:** Missing values were replaced with the column mean.
- **Categorical columns:** Missing values were replaced with the column mode.

Additionally, categorical labels in the response variable were mapped to integers as follows:

$$\text{“Always”} \to 2, \quad \text{“Most of the Time”} \to 1, \quad \text{“Rarely”} \to 0$$

After label encoding, the target variable was separated from the feature matrix. The features were then standardized using `StandardScaler` to ensure each feature had zero mean and unit variance before model fitting.

1.2 Candidate Models and Evaluation Metrics

A total of six candidate classification models were selected and implemented using `scikit-learn`:

- Logistic Regression
- Random Forest Classifier
- Support Vector Classifier (SVC) with a linear kernel
- Decision Tree Classifier
- Linear Discriminant Analysis (LDA)
- K-Nearest Neighbors (KNN)

These models were evaluated using 10-fold stratified cross-validation. Multiple performance metrics were collected:

- **Accuracy**
- **Precision (weighted average)**
- **Recall (weighted average)**
- **F1-score (weighted average)**

- **Custom Accuracy:**

$$\text{Custom Accuracy} = 1 - \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|$$

- **Computational Cost:** Defined as the average sum of fit time and score time across folds.

This comprehensive evaluation framework enabled a data-driven comparison of models in terms of both predictive performance and computational efficiency.

1.3 Random Forest: Hyperparameter Optimization

Following the initial evaluation of candidate models, the Random Forest classifier emerged as one of the top-performing models in terms of classification accuracy. As a result, it was selected for further refinement through hyperparameter optimization.

To fine-tune the model, a grid search was conducted using `GridSearchCV` with 5-fold cross-validation. The optimization objective was to maximize classification accuracy on the validation folds. The following hyperparameters were explored:

- `max_depth`: $\{3, 5, 10, \text{None}\}$
- `min_samples_leaf`: $\{1, 2, 4\}$
- `max_features`: {‘‘`sqrt`’’, ‘‘`log2`’’, `None`}

The grid search tested all combinations of these parameters. For each configuration, a Random Forest classifier was trained and evaluated using cross-validation accuracy. The results were aggregated in a `DataFrame` for further analysis and visualization.

The best-performing configuration selected by cross-validation was:

`max_depth = 5, min_samples_leaf = 4, max_features = ‘‘sqrt’’`

Using this configuration, the model was retrained on the full training set and evaluated on the test set. The model achieved the following test performance:

$$\text{Err}_{\text{test}} = 0.1944$$

Note that mean squared error is reported here to remain consistent with the project's evaluation metric, even though the model is a classifier.

To better understand the effect of different hyperparameter settings, we visualized the mean cross-validated accuracy across all grid configurations:

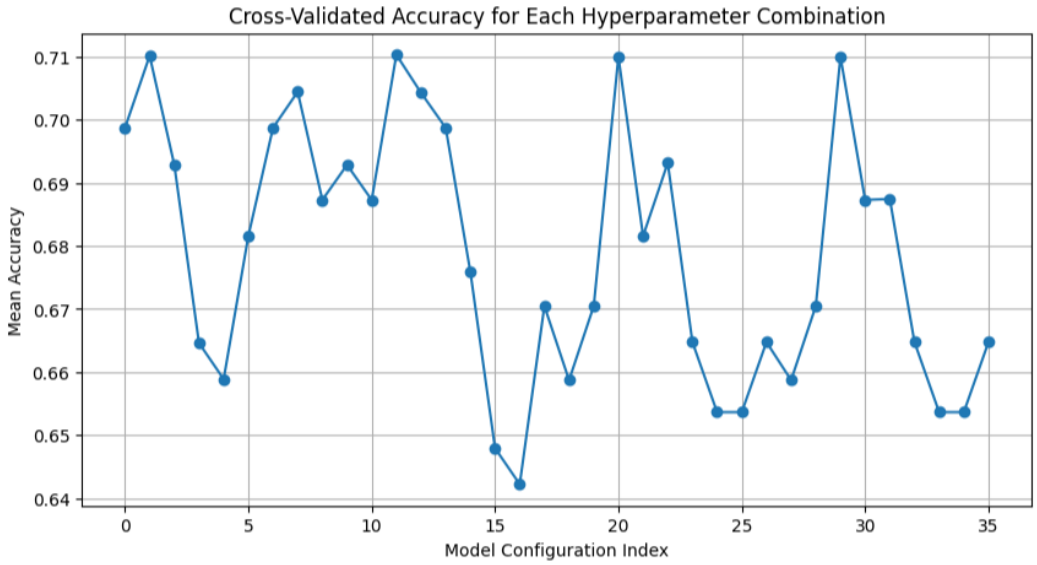

Figure 1: Cross-validated accuracy across hyperparameter configurations.

This optimization process confirmed that Random Forest, when properly tuned, can provide high classification accuracy while balancing model complexity and generalization performance.

1.4 Support Vector Classifier: Hyperparameter Optimization

The Support Vector Classifier (SVC) was also selected as a high-performing candidate based on initial cross-validation results. To further enhance its performance, a hyperparameter tuning procedure was conducted using grid search with 5-fold cross-validation.

Optimization Setup

The base model used a Radial Basis Function (RBF) kernel:

```
SVC(kernel='rbf', random_state=42)
```

The parameter grid for tuning was defined as:

- `C`: $\{0.01, 0.1, 1.0, 10.0\}$
- `max_iter`: $\{50, 100, 300, 1000, 5000\}$

The scoring metric for model selection was classification accuracy. After performing the grid search, the best configuration identified was:

```
C = 1.0,   max_iter = 50
```

This model achieved a test set classification accuracy of approximately:

$$\text{Accuracy}_{\text{test}} = 0.556$$

Visualization and Analysis

To understand the sensitivity of accuracy to different hyperparameter choices, two visualizations were created:

- A line plot of mean accuracy across all tested configurations.

- A heatmap showing accuracy scores across the `C` and `max_iter` grid.

These plots helped identify the regions in parameter space where the SVC performance was maximized.

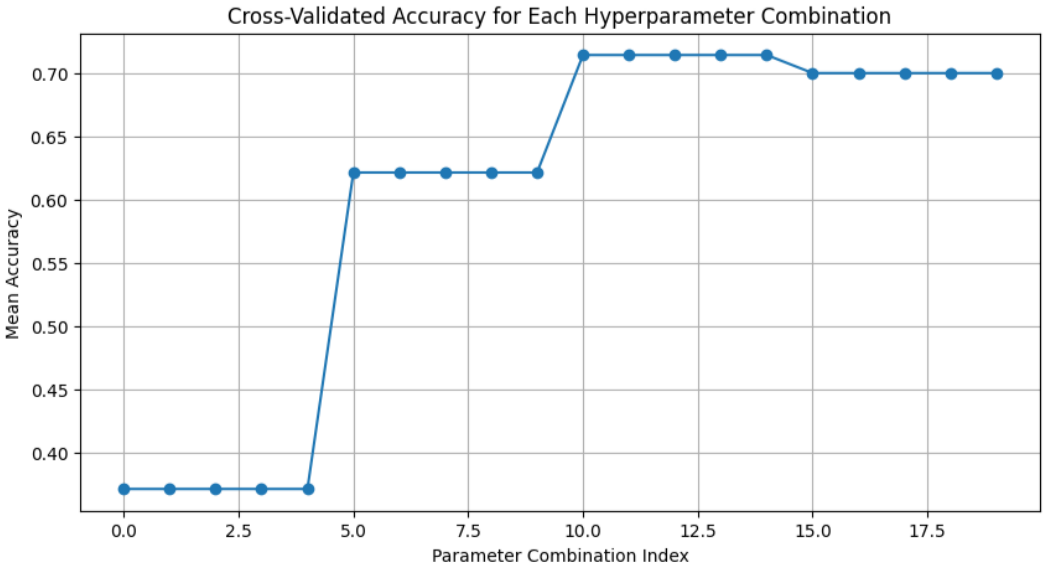

Figure 2: Cross-validated accuracy across SVC configurations.

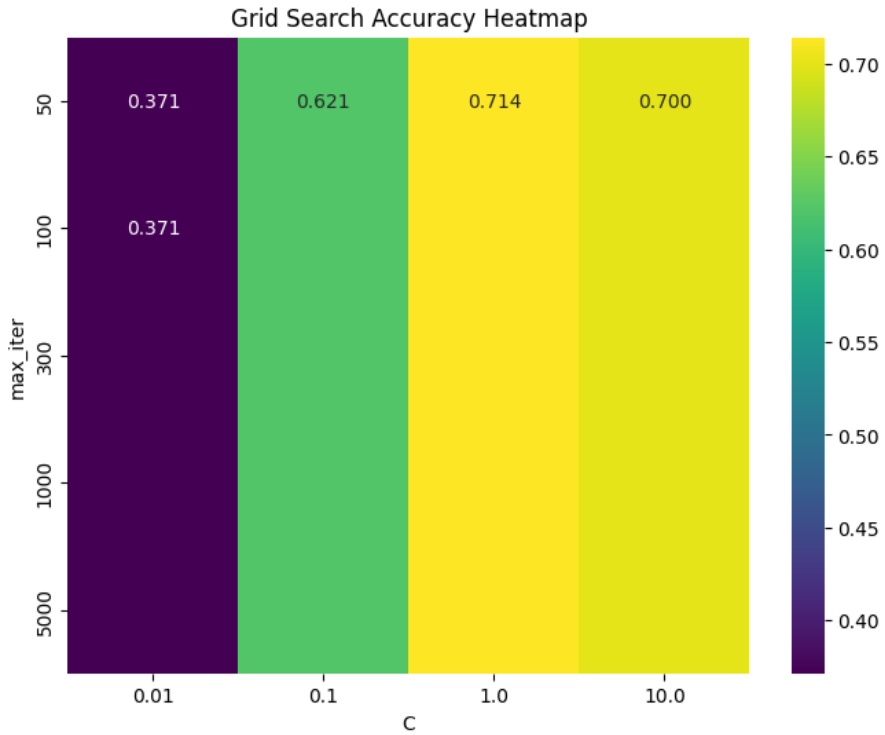

Figure 3: Heatmap of SVC accuracy for varying `C` and `max_iter`.

This optimization process refined the SVC model and established a reliable configuration for further evaluation and comparison.

1.5 Final Model Selection

Based on the hyperparameter optimization results, I selected the Random Forest classifier as the final model. The best-performing configuration identified through grid search was:

`max_depth = 5, max_features = ‘‘sqrt’’, min_samples_leaf = 4`

With other parameters set as their default in scikit-learn package. Random Forests were chosen due to their strong predictive performance and inherent flexibility. They can model complex nonlinear relationships, handle both numerical and categorical features, and are robust to overfitting when appropriately tuned. This flexibility is governed by hyperparameters such as tree depth, feature selection strategy, and leaf size, allowing the model to adapt effectively to various types of datasets. With the chosen parameters, the Random Forest model balances bias and variance, making it a suitable and interpretable choice for final deployment.