Classification Analysis on COVID-19 Data to Inform Future Pandemic Responses in Texas

Project 3

[Abdul Wasay, Om Sumedh Kharwadkar /Team 1]

Executive Summary:

This project aimed to classify COVID-19 risk levels (low, medium, high) across Texas counties using demographic, socioeconomic, and health metrics. Models like Random Forest and XGBoost performed best, achieving 57.14% accuracy and highlighting key factors such as median income, vaccination rates, and population density. However, quantile-based class definitions introduced significant overlap between classes, reducing the model's ability to draw clear decision boundaries. Despite this, the models provide valuable exploratory insights into risk patterns and key drivers, supporting stakeholders in resource allocation and public health strategies. Future efforts should focus on refining class definitions, enhancing feature engineering, and incorporating more granular data for improved separability and accuracy.

Table of Contents

Business Understanding

COVID-19, the deadly disease that gripped the world in 2019 and 2020, prompted us to explore its spread and mortality in relation to specific population metrics. Understanding these dynamics is crucial for addressing the global impact of COVID-19 and the critical need for insights into population behavior, healthcare resources, and socio-economic factors that influenced virus transmission and death rates. By analyzing key metrics such as infection rates, hospitalizations, and demographic trends, we aim to generate insights that can inform public health strategies for both ongoing pandemic recovery and future pandemic preparedness. This report is part 3 of this project and aims to use multiple modelling techniques to understand how different factors affect the overall outcome of our classifications.

Data Preparation

Class Definition & Reason for Classifying using Quantiles

Risk classes (low, medium, high) are created based on the number of deaths per 1,000 population (deaths_per_1000). This metric represents the number of deaths relative to every 1,000 individuals in the population, allowing for standardization across areas of varying population sizes.

Categorized into three risk classes:

- Low Risk: Values in the bottom third (0–33rd percentile).
- Medium Risk: Values in the middle third (33rd–66th percentile).
- High Risk: Values in the top third (66th–100th percentile).

After classification, the risk_class column is converted into a factor to represent categorical outcomes for the modeling task.

risk_class	Count	Proportion
high	87	34.3
medium	83	32.7
low	84	33.1

Table 1: Predictive Class Statistics for Modeling

We used quantiles so that the data can be divided into three evenly distributed risk groups. Quantiles work for us because we wanted to create balanced groups in this situation when there is no defined threshold for risk classification i.e there is no body that is classifying x number of fatalities is high / medium / low for a given population. This also helps eliminate some skews in the data which is the case with our Texas dataset containing health and socioeconomic variables with lots of small counties and few larger population counties.

Combined Dataset with Features for Modeling and Additional Features

The following predictive features were used for the modelling task, based on their relevance to COVID-19 risk and outcomes. Given that this is the same dataset as used before, we have moved the more detailed statistics regarding each of the features to the appendix and only the details of the derived features and the risk classes are shown.

Demographic Features:

- median_income: Median income of the population.
- median_age: Median age of the population.
- bachelors_degree_or_higher_25_64: Percentage of adults aged 25–64 with a bachelor's degree or higher.
- population_density_per_sqmi: Population density per square mile.

Socioeconomic Features:

- percent_below_poverty: Percentage of the population below the poverty line.
- percent_unemployed_CDC: Unemployment rate.

Health Metrics:

- percent_fair_or_poor_health: Percentage of people self-reporting fair or poor health.
- percent_smokers: Percentage of the population that smokes.
- percent_adults_with_obesity: Percentage of adults with obesity.
- percent_physically_inactive: Percentage of adults with insufficient physical activity.
- percent_with_access_to_exercise_opportunities: Percentage of people with access to facilities for physical activity.
- percent_excessive_drinking: Percentage of adults engaging in excessive drinking.
- percent_adults_with_diabetes: Percentage of adults diagnosed with diabetes.
- percent_vaccinated: Percentage of the population vaccinated against COVID-19.

Outcome Feature:

- risk_class: Target variable representing low, medium, or high risk.

Derived Features:

- health_risk_score: Combines health-related features into a composite metric: This average represents an overall health risk indicator for the population, summarizing poor health, smoking habits, and obesity.
- socioeconomic_score: Combines poverty and unemployment metrics into a single score: This average captures the socioeconomic challenges of the population, which may correlate with COVID-19 risks.

Derived Features	**Mean**	**Median**	**SD**	**Minimum**	**Maximum**	**Variance**

Health Risk Score	22.26	22.00	2.86	16.27	32.25	8.18
Socioeconomic Score	11.56	11.25	3.77	1.85	25.90	14.19

Table 2: Derived Features' Statistics and for Modeling

Given that we are working with a cleaned dataset prepared in Project 1, there are **no missing values**. *However a check is still performed so that there are no missing values before we feed this into the various models.*

Modeling

Data Prep for Training & Testing

The dataset is divided into a training set (80%) and a testing set (20%). This split provides enough data to train while reserving enough to test its real world performance on unseen data. We also split ensuring that the risk_class variable's class distribution is preserved through *stratified sampling*, ensuring balance across the class distribution of low, medium, and high risk categories.

5-fold cross-validation is used when assessing model performance and hyperparameters tuning. This means that the training data is divided into five folds, where in each iteration, one fold becomes the validation set while the others are used for training. This helps in ensuring that the model is performing similarly across each data point in our dataset. This will also help with detecting overfitting, as it is evaluating the model for its stability over the different subsets.

Hyperparameter tuning is used to find the best configuration of the models' various parameters that are not specific to the data that is being used for training. Each of the modeling techniques have various hyperparameters to be tuned. (Random Forest, SVM, XGBoost, Neural Networks) has specific hyperparameters to tune. For instance:

- Logistic Regression: Parameters like regularization strength (r C) and the type of regularization (L1, L2, or Elastic Net) are adjusted to prevent overfitting and balance model simplicity with predictive performance.
- Random Forest: Parameters like the number of trees and the number of features considered at each split are adjusted to balance model accuracy and computational efficiency.
- SVM: Parameters like the cost of misclassification and the kernel width control the trade-off between model complexity and margin width.
- XGBoost: Parameters such as learning rate, maximum tree depth, and number of boosting rounds optimize the model's predictive power while avoiding overfitting.
- Neural Networks: Factors like the number of neurons in the hidden layer and regularization weights influence the network's ability to generalize patterns without overfitting to training data.

The caret package automates this process by evaluating various combinations of hyperparameters during cross-validation and selecting the configuration that performs best on a specified metric, such as accuracy.

Modelling and Justifications

1. ***Regression:***

- Explanation: Regression (in this case, logistic regression) predicts the likelihood of different outcomes by analyzing how features are linked to each category. It identifies patterns to classify outcomes like "low," "medium," or "high" risk. Logistic regression estimates probabilities for each class, which can help in understanding the impact of each feature.
- Why Appropriate for our Classification Task: Logistic regression is appropriate because it provides interpretable results, such as coefficients showing the impact of each feature on the target variable (risk_class). It is effective for understanding relationships between predictors and the outcomes, making it a suitable baseline model. Logistic regression is straightforward and interpretable. It helps us understand how each feature influences the risk level and provides a solid baseline for comparing more complex models.

Metric	**Value**
Accuracy	53.06%
95% Confidence Interval	38.27% – 67.47%
Kappa	0.2961
P-Value (Acc > NIR)	0.0063

Table 3: Regression Evaluation Metrics

Statistic	**Class: High**	**Class: Low**	**Class: Medium**
Sensitivity	58.82%	56.25%	43.75%
Specificity	81.25%	81.82%	66.67%
Pos Pred Value	62.50%	60.00%	38.89%
Neg Pred Value	78.79%	79.41%	70.97%
Balanced Accuracy	70.04%	69.03%	55.21%

Table 4: Regression Class Statistics

Strengths:

- Balanced performance across all classes, with no extreme outliers in sensitivity or specificity.
- Strong balanced accuracy for the high and low classes (70.04% and 69.03%, respectively).

Weaknesses:

- Poor sensitivity for the medium class (43.75%) suggests challenges in identifying these cases.
- The model’s linear approach may oversimplify relationships in the data, limiting its effectiveness compared to more complex models.

2. ***Random Forest:***

- ○ Explanation: Random Forest builds many decision trees and combines their predictions to make a final decision. Each tree looks at different parts of the data, so the combined result is more accurate and less likely to be affected by noise or outliers.
- ○ Why Appropriate for our Classification Task: It is great for handling complex data with many interacting features. Random Forest also shows us which features are the most important for predicting risk levels, making it useful for understanding the data. It is robust to noise, can manage missing data, and provides feature importance scores, which are valuable for understanding the contribution of each feature to the model's predictions.

Metric	**Value**
Accuracy	57.14%
95% Confidence Interval	42.21% – 71.18%
Kappa	0.3553
P-Value (Acc > NIR)	0.0011

Table 5: RF Evaluation Metrics

Statistic	**Class: High**	**Class: Low**	**Class: Medium**
Sensitivity	70.59%	43.75%	56.25%
Specificity	71.88%	87.88%	75.76%
Pos Pred Value	57.14%	63.64%	52.94%
Neg Pred Value	82.14%	76.32%	78.12%
Balanced Accuracy	71.23%	65.81%	66.00%

Table 6: RF Class Statistics

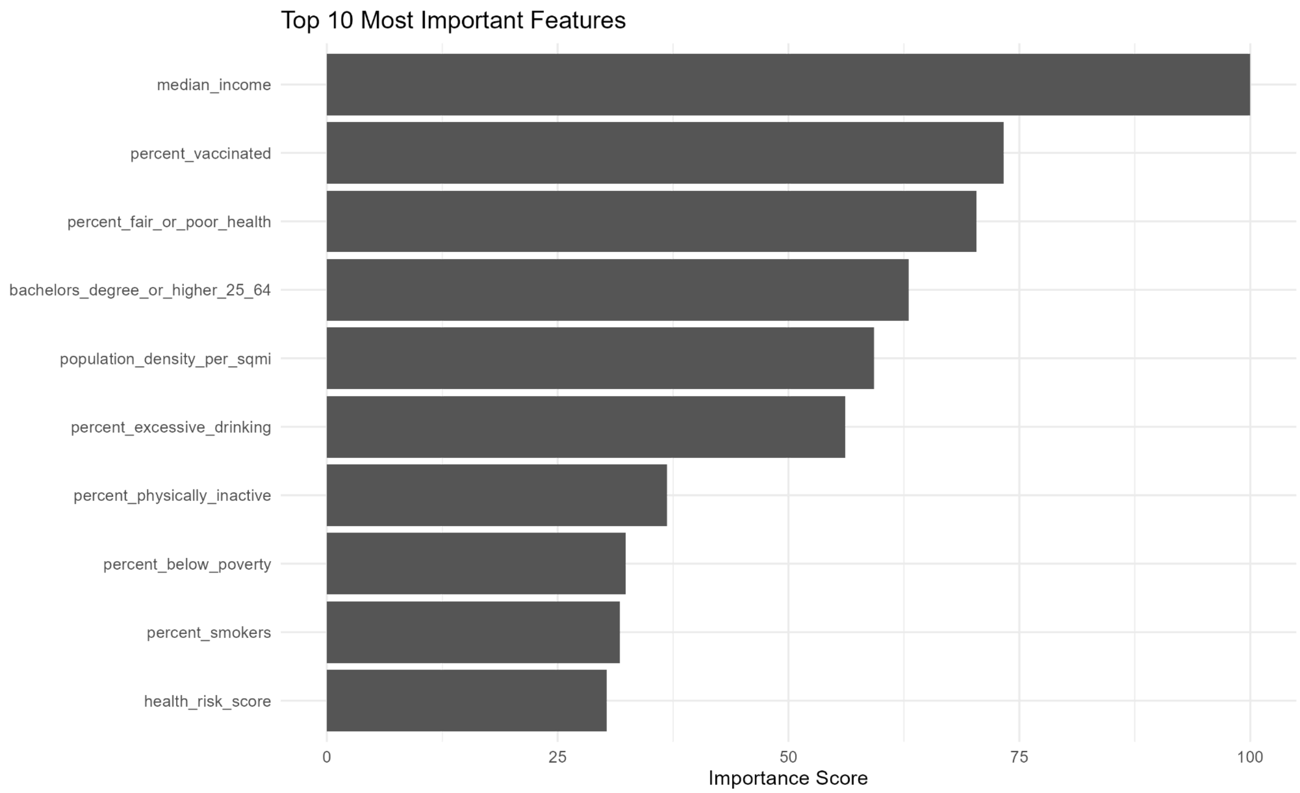


Illustration 1: RF Feature Importance

Strengths:

- The model performs best for the high class, with a balanced accuracy of 71.23% and a sensitivity of 70.59%, demonstrating its ability to correctly classify most high risk cases.
- Strong specificity for the low class (87.88%) ensures that false positives for this class are minimized.
- The feature importance analysis shows that median income, percent vaccinated, and population density per square mile are key drivers of performance, effectively capturing socioeconomic and health-related nuances.

Weaknesses:

- The model struggles with sensitivity for the low class (43.75%), missing many true cases.
- The medium class exhibits moderate sensitivity (56.25%), reflecting difficulty in distinguishing these cases from others.

3. ***Support Vector Machines:***

- Explanation: SVM works by drawing boundaries between different groups in the data, aiming to separate them as clearly as possible. It uses a mathematical technique to find the best boundary, even when the groups are not easily separable.

    - Why Appropriate for our Classification Task: SVM is good for finding patterns in data where the differences between groups are subtle or not immediately obvious. It can also handle more complex relationships while remaining efficient, making it suitable for our task.

Metric	Value
Accuracy	48.98%
95% Confidence Interval	34.42% – 63.66%
Kappa	0.2315
P-Value (Acc > NIR)	0.0275

Table 7: SVM Evaluation Metrics

Statistic	Class: High	Class: Low	Class: Medium
Sensitivity	70.59%	25.00%	50.00%
Specificity	65.62%	87.88%	69.70%
Pos Pred Value	52.17%	50.00%	44.44%
Neg Pred Value	80.77%	70.73%	74.19%
Balanced Accuracy	68.11%	56.44%	59.85%

Table 8: SVM Class Statistics

Strengths:

- SVM achieves strong sensitivity for the high class (70.59%), effectively identifying most true high risk cases.
- High specificity for the low class (87.88%) ensures minimal false positives for this category.

Weaknesses:

- Sensitivity for the low class is extremely poor (25.00%), indicating that the model struggles to identify true low risk cases.
- The medium class also shows moderate performance, with a balanced accuracy of 59.85%, reflecting difficulty in distinguishing this group from others.
- Overall accuracy (48.98%) and Kappa (0.2315) suggest suboptimal classification performance compared to other models.

4. ***XGBoost (Grad Task):***

- Explanation: XGBoost (Extreme Gradient Boosting) builds a series of small decision trees, where each tree corrects the errors made by the previous ones. By combining all the trees, it makes strong predictions that are both fast and accurate.
- Why Appropriate for our Classification Task: They work really efficiently for large datasets and can capture complex relationships between features. They can also optimize for accuracy while keeping overfitting in check, making it an appropriate choice for our risk classification task.

Metric	**Value**
Accuracy	57.14%
95% Confidence Interval	42.21% – 71.18%
Kappa	0.3553
P-Value (Acc > NIR)	0.0011

Table 9: XGBoost Evaluation Metrics

Statistic	**Class: High**	**Class: Low**	**Class: Medium**
Sensitivity	64.71%	50.00%	56.25%
Specificity	68.75%	81.82%	84.85%
Pos Pred Value	52.38%	57.14%	64.29%
Neg Pred Value	78.57%	77.14%	80.00%
Balanced Accuracy	66.73%	65.91%	70.55%

Table 10: XGBoost Class Statistics

Strengths:

- XGBoost achieves the best balanced accuracy for the medium class (70.55%), demonstrating its ability to effectively identify true medium risk cases while minimizing false positives.
- Balanced performance across the high and low classes, with balanced accuracies of 66.73% and 65.91%, respectively.
- The model excels in capturing interactions between features, leveraging variables like median income, population density, and percent vaccinated to make nuanced predictions.

Weaknesses:

- Moderate sensitivity for the low class (50.00%) and specificity for the high class (68.75%) suggest opportunities for improvement in distinguishing these categories.

5. ***Neural Network (Grad Task):***

- ○ Explanation: A Neural Network uses layers of interconnected "neurons" to assess the data passing through them and find patterns within them. They are built in layers that build on top of the previous layer passing over "weights" that is the learned pattern allowing us to capture complex relationships.
- ○ Why Appropriate for our Classification Task: Neural Networks are very useful when finding hidden patterns in the dataset. They work really well when there are many variables that interact with each other in non-linear ways, which makes them an appropriate choice for predicting risk levels.

Metric	**Value**
Accuracy	53.06%
95% Confidence Interval	38.27% – 67.47%
Kappa	0.2903
P-Value (Acc > NIR)	0.0063

Table 11: NN Evaluation Metrics

Statistic	**Class: High**	**Class: Low**	**Class: Medium**
Sensitivity	88.24%	12.50%	56.25%
Specificity	56.25%	96.97%	75.76%
Pos Pred Value	51.72%	66.67%	52.94%
Neg Pred Value	90.00%	69.57%	78.12%
Balanced Accuracy	72.24%	54.74%	66.00%

Table 12: NN Class Statistics

Strengths:

- Exceptional sensitivity for the high class (88.24%) ensures that most true high risk cases are identified.
- High specificity for the low class (96.97%) minimizes false positives.

Weaknesses:

- Extremely poor sensitivity for the low class (12.50%) indicates difficulty in identifying true cases.

- Moderate performance for the medium class (Balanced Accuracy: 66.00%) reflects the model's struggles with this category.

Hyperparameter Tuning

For this project, the 5-fold cross-validation method, from the ‘caret’ package, was used as it provided various combinations of hyperparameters systematically. The grid search technique was applied during cross-validation to test different configurations of hyperparameters for each classification method.

Below, the table summarizes the best-performing hyperparameters for each model, as identified during this process. These values represent the configurations that produced the highest accuracy or other evaluation metrics during training.

Model	**Hyperparameter(s)**	**Optimal Value(s)**
Logistic Regression	decay	decay = 1e-04
Random Forest	mtry	16
SVM	sigma, C	sigma = 0.05724155, C = 0.25
XGBoost	nrounds, max_depth, eta, gamma, colsample_bytree, min_child_weight, subsample	nrounds = 100, max_depth = 2, eta = 0.4, gamma = 0, colsample_bytree = 0.8, min_child_weight = 1, subsample = 0.75
Neural Network	size, decay	size = 5, decay = 0.1

Table 13: Best Hyperparameters from Modeling

Accuracy was chosen as the primary metric for hyperparameter tuning because it provides a straightforward and comprehensive measure of overall model performance across all classes. Since the dataset is relatively balanced, accuracy is an effective metric for evaluating how well the model predicts the correct class for most instances. It also aligns with the goal of finding a model that performs well across all risk levels (low, medium, high).

Evaluation and Recommendations

Speed / Efficiency Comparison

Model	User Time (seconds)	System Time (seconds)	Elapsed Time (seconds)
Logistic Regression	0.28	0.00	0.47
Random Forest	1.34	0.02	1.72

Support Vector Machine (SVM)	0.64	0.03	0.78
XGBoost	13.25	18.33	68.03
Neural Network	0.94	0.00	1.07

Table 14: Modelling Speed Comparison

- Speed vs. Complexity: Logistic Regression and SVM are the quickest, making them ideal for quick baselines or when computational resources are limited.
- Trade-Off for Accuracy: XGBoost is the slowest but is often the most accurate for complex tasks, making it a valuable choice when time isn't a constraint. We will see if it pays off when we check for accuracy later.
- Balanced Choices: Random Forest and Neural Networks strike a balance between training time and predictive power, offering efficient yet robust performance for most datasets.

Model Performance

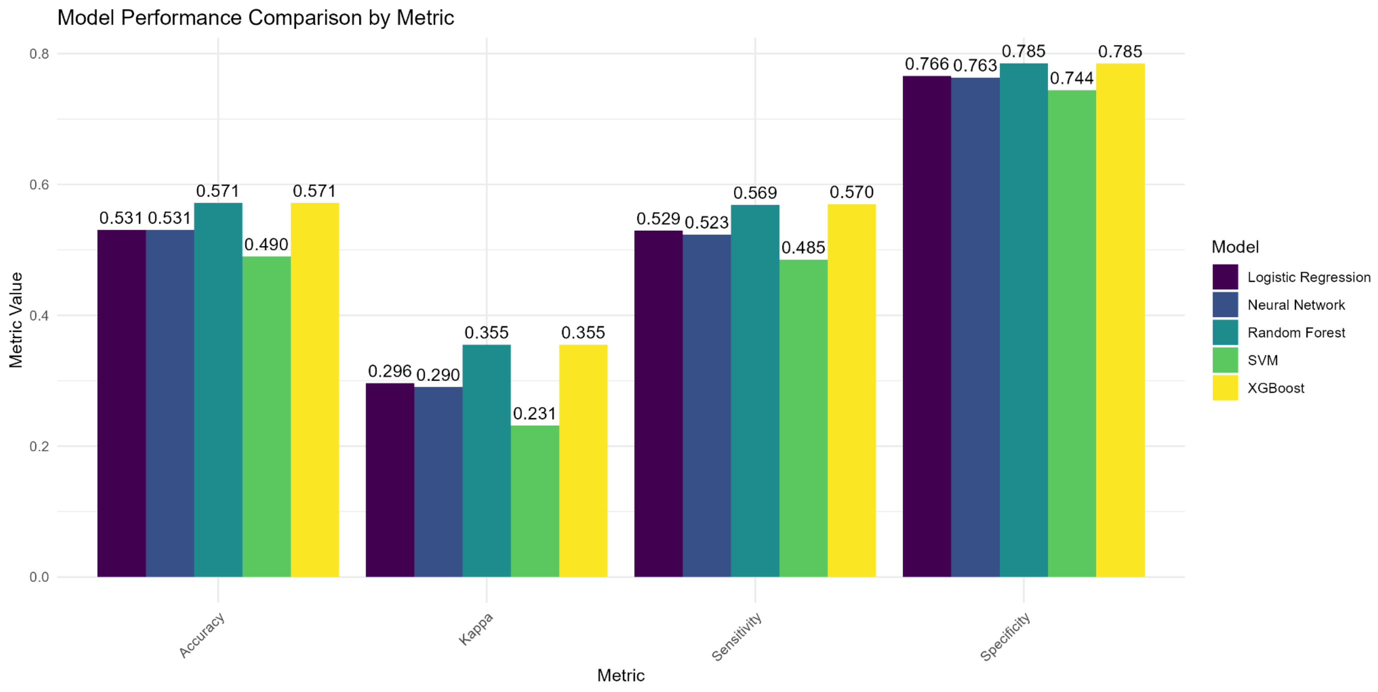


Illustration 2: Modelling Comparison Metrics

1. Accuracy: Random Forest and XGBoost achieved the highest accuracy (0.571), correctly classifying 57.1% of cases. SVM had the lowest accuracy (0.490), indicating limited overall effectiveness in classifying the data.

2. Kappa: Random Forest and XGBoost achieved the highest Kappa score (0.355), showing moderate agreement between predicted and actual classes, accounting for chance. SVM had the lowest Kappa (0.231), indicating poorer reliability compared to other models.

3. Sensitivity: Random Forest (0.569) and XGBoost (0.570) had the highest sensitivity, performing well in identifying positive cases. SVM had the lowest sensitivity (0.485), reflecting difficulties in correctly identifying positive cases.

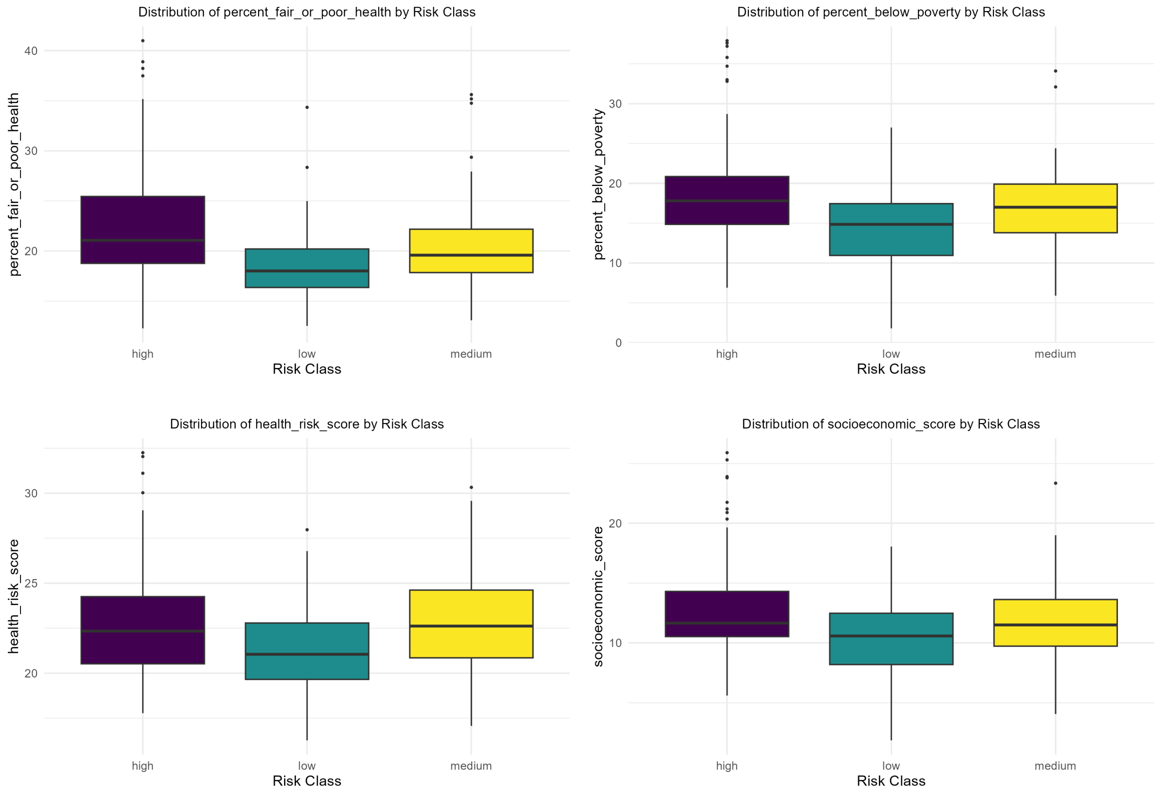
4. Specificity: Random Forest and XGBoost had the highest specificity (0.785), excelling in identifying negative cases. SVM had the lowest specificity (0.744), performing worse than other models in identifying negative cases.

Overall Evaluation

Random Forest and XGBoost are the most effective for this classification task, providing the best accuracy, Kappa, sensitivity, and specificity. Both models excel in handling non-linear relationships and interactions between features, which likely helped them achieve superior performance in this dataset.

While they were the best performers, their overall accuracy (57.1%) and moderate Kappa score (0.355) indicate that even these models struggled with the dataset. The way we created the predicted classes—by dividing the data into equal groups to eliminate class imbalance—likely contributed to the observed performance issues. While this approach ensured balanced class distributions, it may have inadvertently introduced significant overlap between the classes. This overlap makes it harder for the models to differentiate between them effectively, leading to moderate performance even for the best models like Random Forest and XGBoost.

Feature Distributions - Understanding Performance of our Models



We divided the dataset into three risk classes (low, medium, high) based on quantiles of a calculated metric (e.g., deaths per 1000 population). This approach ensured roughly equal-sized groups but did not guarantee that the groups were well-separated in terms of their feature values.

Many features have overlapping distributions across the classes, as seen in the descriptive statistics. For median_income, (for example) the ranges for high, medium, and low overlap significantly.

- high: Min = 24,794, Max = 71,389
- medium: Min = 33,125, Max = 73,655
- low: Min = 26,063, Max = 93,645

This overlap made it difficult for the model to learn clear decision boundaries between the classes.

The calculated composite features (health_risk_score, socioeconomic_score) also had limited separation:

For health_risk_score means were nearly identical for high (22.77), medium (22.74), and low (21.26). This means that the composite feature did not amplify meaningful differences between the classes. The same can be said about the socioeconomic_score where the ranges overlap significantly, and the means (high = 12.77, medium = 11.60, low = 10.27) are too close for effective classification.

Usefulness of Models for Stakeholders & Assessing Value

The model's utility for the stakeholder is limited by the significant overlap between the risk classes, which reduces its ability to reliably predict low, medium, and high risk cases. However, its value can still be assessed in specific contexts and with targeted improvements:

- Redefine Classes:
  Revising the risk class definitions to minimize overlap could significantly enhance model utility. Using domain-specific thresholds or clustering methods might better align the classes with real-world distinctions.
- Feature Engineering:
  Develop new features or interactions that better capture differences between risk classes. For example: Ratios of economic to health indicators or Regional clustering or geographic features.
- Additional Data:
  Incorporating more granular data or external datasets could improve class separation and model performance.

Use Case and Updates

The model, in its current form, provides limited value for precise risk classification but is still a useful exploratory tool for identifying trends and prioritizing areas of concern.

- Exploratory Tool:
  The model is most useful as an exploratory tool to identify patterns and high-level trends rather than providing definitive risk classifications. Stakeholders can use it alongside other qualitative or quantitative methods for decision-making.
- Potential Applications:
  - Broad Risk Categorization: The model can help identify areas likely to fall into a high risk category based on overarching trends.
  - Resource Allocation: While imperfect, the model could provide a rough prioritization framework for allocating resources or initiating further analysis.

Graduate Work

The 2 extra modeling techniques for the graduate portion were: XGBoost and Neural Network and can be found on page 11 and 12 respectively. They were included in the evaluations along with all the other modelling methods.

Student Contributions

Both members worked on equal contributions of the code and the report. The initial data handling and setup of the modeling part was handled by Wasay and the results of the models along with their interpretations was conducted by Om. We used Google colab for the R file and wrote the report using Google Docs.

Appendix Contributions

The following are the details of the features taken from Project 2.

Feature	**Mean**	**Median**	**Variance**	**Range**	**Mode**
Hispanic Population	42,023.26	5,068.5	2.97E+10	12 - 1,910,535	0
Median Income	49,894.34	48,311	1.47E+08	24,794 - 93,645	42,500
Percent Below Poverty	16.87	16.5	33.06	1.8 - 37.9	17.2
Percent Fair or Poor Health	20.60	19.73	23.85	12.29 - 40.99	4.74
Percent Smokers	14.98	14.9	2.34	10.64 - 19.87	14.9
Percent Adults with Obesity	31.49	30.7	27.45	21.6 - 47.3	28.6
High School Diploma	14,130.69	3,281	2.28E+09	3 - 559,393	695

Bachelor’s Degree	12,947.94	1,289.5	2.59E+09	11 - 237,000	237
Percent Vaccinated	37.03	39	90.27	9 - 55	38

Table 15: Features, Statistics for Modeling