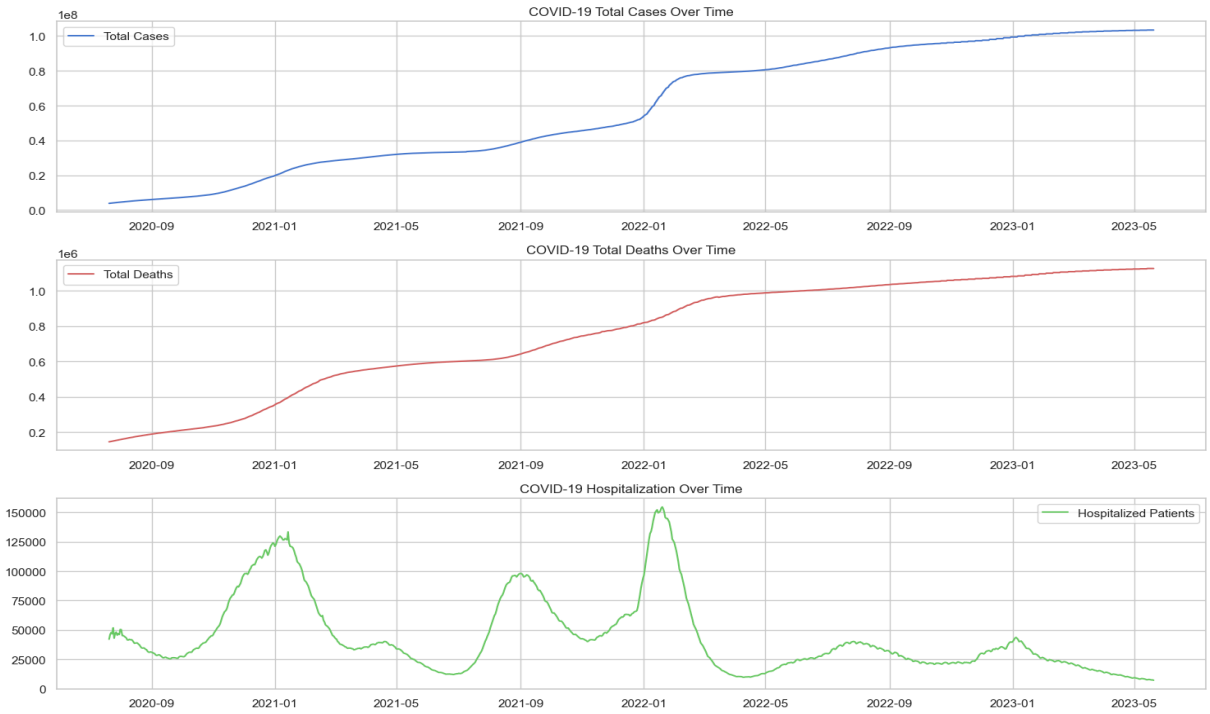
Group Members:
Chongyi Wang - 2244409
Nathan Tran - 2049679
Victor Ha - 2018028

Comprehensive Analysis and Predictive Modeling of Global COVID-19 Trends

In this project, we worked with a COVID-19 dataset that contains 1,034 rows and 19 columns after removing duplicate entries. We found that most columns have no missing values, but a few columns, such as new_deaths, daily_case_change_rate, daily_death_change_rate, 7day_avg_new_cases, and 7day_avg_new_deaths, have some missing data. Overall, the amount of missing data is small compared to the full dataset. The data covers a timeline from July 21, 2020, to May 20, 2023, and includes important information like total cases, new cases, deaths, hospitalizations, and ICU admissions. Based on the basic statistics, we observed about 96,367 new cases and 953 new deaths reported daily on average. We also noticed a large spread in the data, showing that the pandemic situation changed a lot over time.

When reviewing the dataset, we found that missing values were limited to a few columns. The number of missing entries was relatively low, with the highest being 13 missing values in the 7day_avg_new_deaths column. Since the missing data was minimal and spread across a few columns, we decided that it would not significantly impact the overall analysis. To maintain consistency, we handled missing values using appropriate techniques such as interpolation or by removing rows when necessary, depending on the analysis requirements. After addressing the missing data, we proceeded with descriptive statistical analysis to summarize the main characteristics of the dataset.

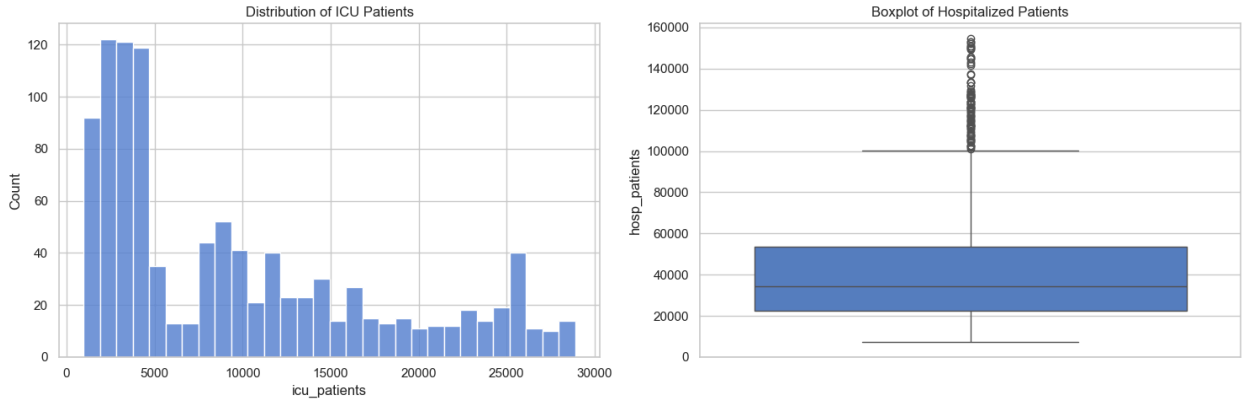


In the first graph, showing **Total Cases Over Time**, we see a steady increase in the number of cases from mid-2020 to mid-2023. There are a few periods where the curve becomes much steeper, especially around late 2021 and early 2022, suggesting major waves of infections during that time. After these spikes, the growth rate appears to slow down slightly but still continues to rise steadily.

In the second graph, **Total Deaths Over Time**, a similar pattern is visible. Deaths increased consistently throughout the pandemic, with steeper rises during the same periods where total cases grew sharply. However, compared to the total cases, the death curve is smoother and shows a slight flattening toward the end of the timeline, which could be due to improvements in treatment, vaccination efforts, or better healthcare management over time.

The third graph shows **Hospitalized Patients Over Time** and it has a different pattern. Here, we notice clear waves of hospitalizations, with sharp peaks followed by sharp declines. There were major hospitalization peaks around early 2021 and early 2022, matching the major COVID-19 waves seen in the first two graphs. After the peak in early 2022, hospitalization numbers dropped significantly and although there were smaller spikes afterward, the overall trend is downward by 2023.

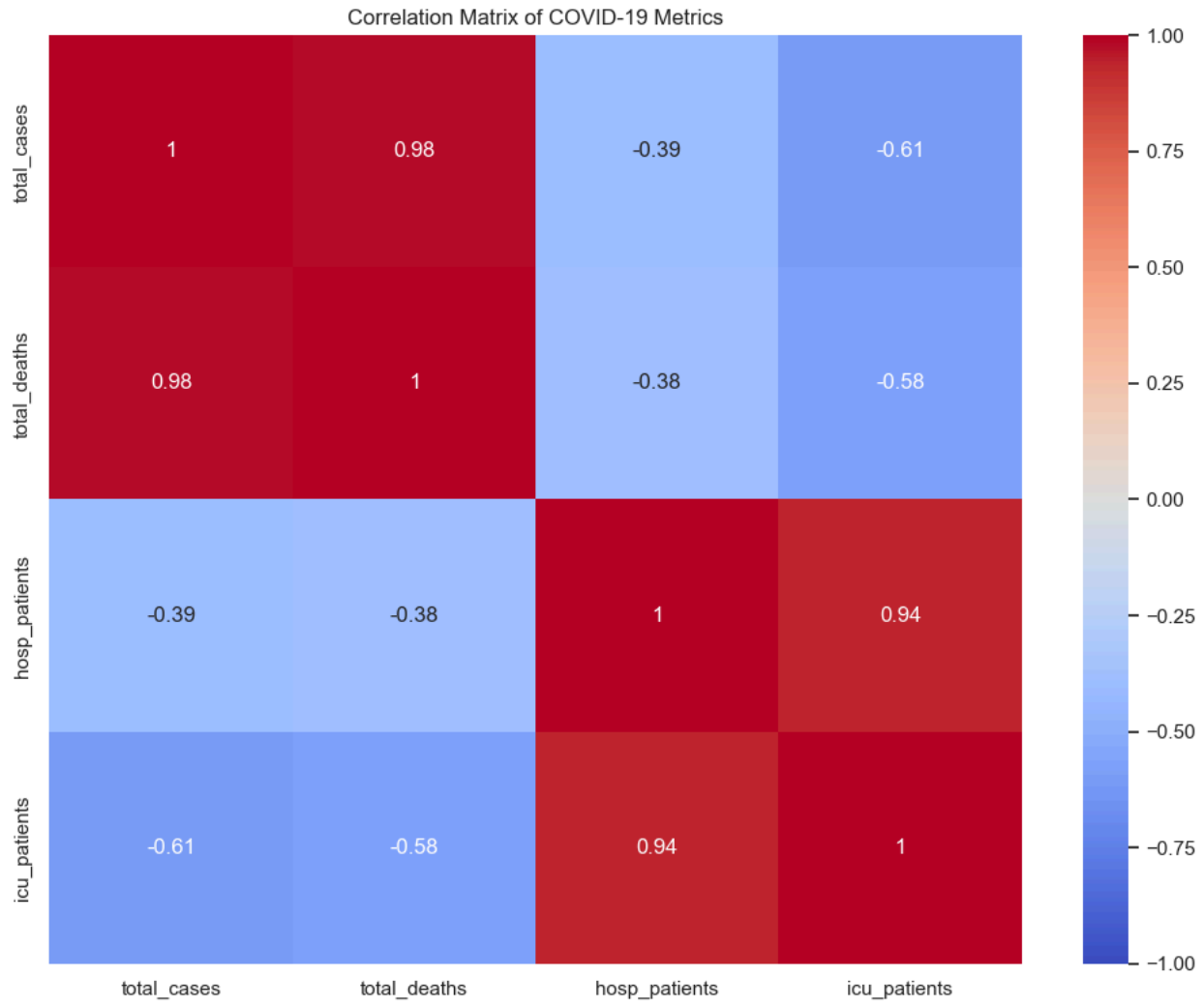
Overall, these graphs together show that COVID-19 spread rapidly during certain periods, causing big increases in hospitalizations and deaths. However, over time, while total cases continued to grow, hospitalizations and deaths seem to have become more controlled, possibly due to public health interventions like vaccines, better treatments, or natural immunity.



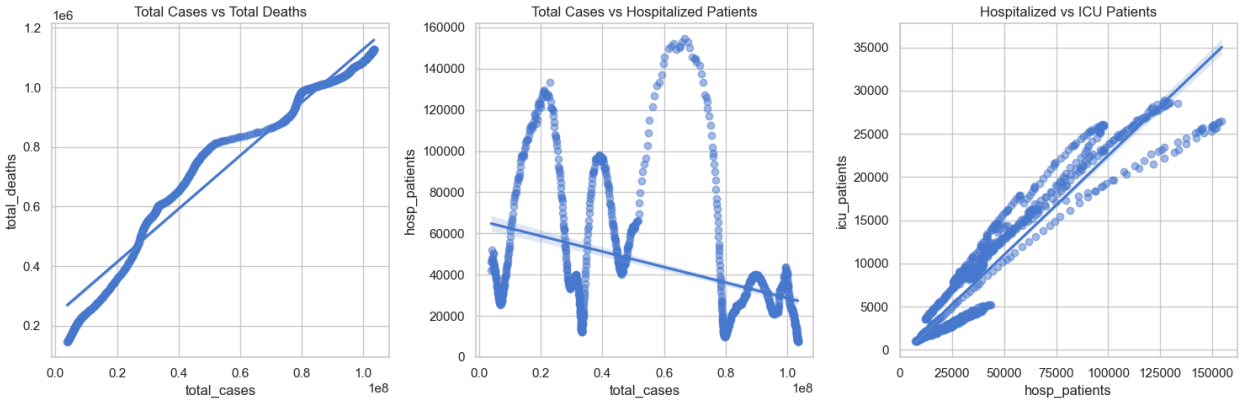
The histogram on the left shows the distribution of ICU patients over the recorded time period. Most of the observations are concentrated between 0 and 5,000 ICU patients, with a sharp peak around that range. After 5,000, the number of days with higher ICU counts drops off, although there are still several days with ICU numbers reaching up to 25,000 and beyond. The distribution is right-skewed, meaning there were fewer days with extremely high ICU admissions compared to days with lower ICU numbers.

The boxplot on the right shows the distribution of hospitalized patients. From the boxplot, we can see that the median number of hospitalized patients is around 34,000. The interquartile range (the middle 50% of the data) spreads roughly from about 22,000 to 53,000 patients. There are many outliers above 100,000, indicating that on some days, hospitalization counts were unusually high compared to the typical range. This suggests that while most of the time hospitalizations stayed within a certain range, during major COVID-19 waves, there were extreme spikes in hospitalizations.

Together, these graphs show that while ICU and hospital patient numbers were usually moderate, the healthcare system experienced significant pressure during peak periods of the pandemic.



The correlation matrix shows a very strong positive correlation (0.98) between total cases and total deaths, meaning that as cases increased, deaths also increased. Hospitalized patients and ICU patients also have a strong positive correlation (0.94), which makes sense because more hospitalizations usually lead to more ICU admissions. Interestingly, total cases and hospitalizations have a weak negative correlation (-0.39), and total cases and ICU patients have a stronger negative correlation (-0.61). This suggests that even though cases were increasing, the proportion of severe cases needing hospitalization or ICU care was not always increasing at the same rate, possibly because of better treatments, vaccines, or milder variants later on.

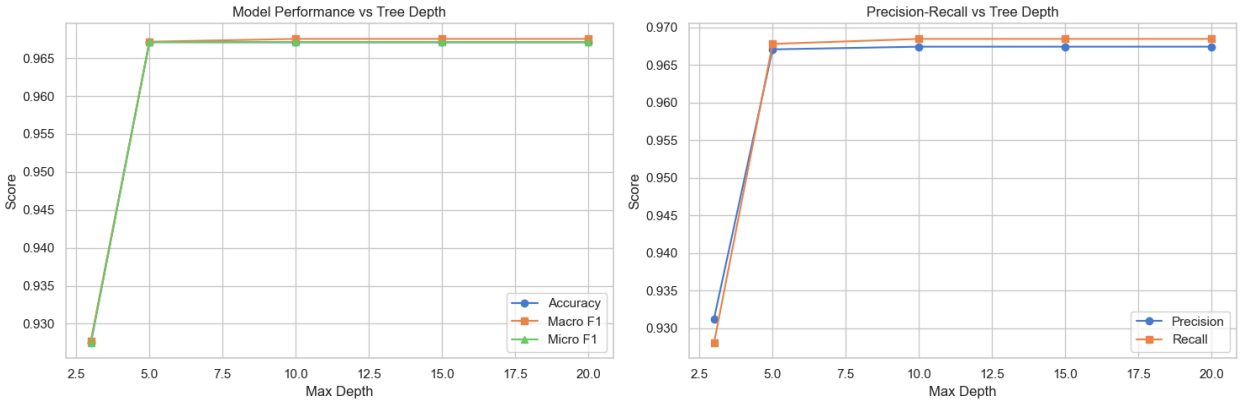


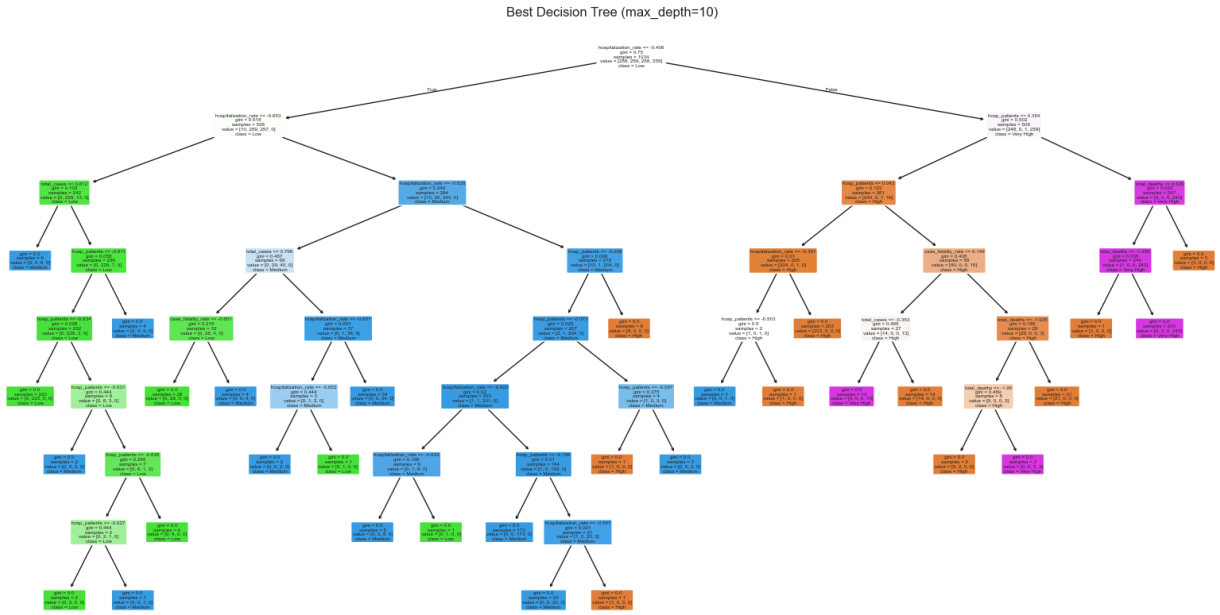
In the first plot, Total Cases vs Total Deaths, we see a strong positive relationship. As total cases increase, total deaths also increase almost linearly, which matches what we saw earlier in the correlation matrix.

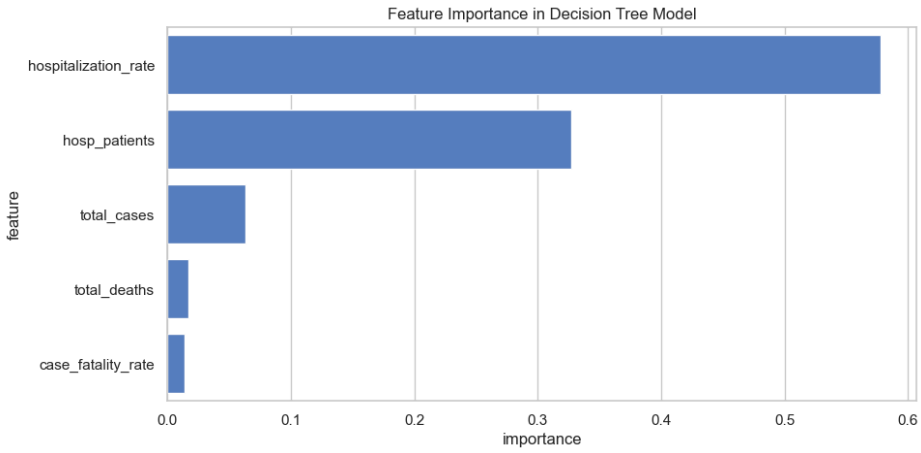
In the second plot, Total Cases vs Hospitalized Patients, the points are much more spread out and show a weak negative trend overall. This suggests that even though the total number of cases was rising over time, the number of hospitalized patients didn't rise at the same rate and actually started to decrease, possibly because of improvements like vaccines or better treatments.

In the third plot, Hospitalized Patients vs ICU Patients, we see a very strong positive linear relationship. This means that when the number of hospitalized patients increased, the number of ICU patients also increased closely with it, which makes sense because some hospitalized patients need critical care.

The scatter plots show that total cases and total deaths have a strong positive linear relationship, meaning that as cases increased, deaths also increased steadily. However, the relationship between total cases and hospitalized patients is weak and slightly negative, suggesting that hospitalization rates did not rise as fast as cases, possibly due to better treatments or vaccinations over time. Meanwhile, the relationship between hospitalized patients and ICU patients is strongly positive, indicating that higher hospitalization numbers were closely linked to higher ICU admissions.







In the first graph, we can see that as the tree depth increases from 3 to 5, the model's accuracy, macro F1, and micro F1 scores all improve a lot. After depth 5, the scores stay very high and stable, even when the tree gets deeper up to 20. This shows that a tree with a max depth of 5 or more is already good enough for this dataset and that deeper trees don't really help improve performance much after that.

In the second graph, precision and recall also show a big jump from depth 3 to 5. After that, both precision and recall stay very close to each other and remain stable as tree depth increases. This is a good sign because it means the model is balanced — it doesn't favor precision over recall too much or vice versa — and both metrics stay strong once the tree is deep enough.

The best decision tree was created using a max depth of 10. We noticed that the tree mainly splits based on hospitalization_rate and hosp_patients. These features appear early in the

tree, meaning they are very important for making decisions. Other features like total_cases, case_fatality_rate, and total_deaths are used later on, but they are not as important for the main splits. Overall, the tree looks well-organized and uses meaningful features to classify the data.

The feature importance plot shows that hospitalization_rate is the most important feature by far, followed by hosp_patients. Together, these two features dominate the model's decision-making process. Other features like total_cases, total_deaths, and case_fatality_rate have much lower importance, which means they don't have as much impact on how the tree makes its predictions.

Overall, increasing the tree depth made the model much better up to depth 5, but after that, the scores stayed high with no big improvements. The decision tree mostly depends on hospitalization-related features, especially hospitalization_rate, to predict the outcomes. This shows that hospitalization data is a very strong indicator when it comes to predicting the severity of COVID-19 in this dataset.

Task 4

Objective

The objective of this task was to predict the ICU requirement level (categorized as "Low," "Medium," "High," "Very High") based on COVID-19-related health indicators using the K-Nearest Neighbors (KNN) classification algorithm. We evaluated how different numbers of neighbors and kernel functions affected model performance using cross-validation and several classification metrics.

Data Preparation

We selected a subset of relevant features from the dataset:

- Total_cases
- total_deaths
- hosp_patients
- case_fatality_rate
- hospitalization_rate

The target variable (icu_patients) was discretized into four classes using quartile binning (Low, Medium, High, and Very High). Missing feature values were imputed with zeros. All features were standardized using StandardScaler to ensure comparability across features.

Modeling Methodology

We implemented the K-Nearest Neighbors (KNN) classifier using 5-fold cross-validation to ensure robust evaluation.

Key setup details:

- Kernel functions: Although KNN does not use kernel functions directly (it is not a kernelized algorithm like SVM), we explored varying the number of neighbors, which plays a similar role in controlling the locality of decision boundaries.
- Numbers of neighbors tested: [2, 5, 10, 50]
- Performance metrics collected:
    - Accuracy
    - Precision
    - Recall
    - Macro F1-score
    - Micro F1-score
    - Training time

Each metric was averaged across all folds.

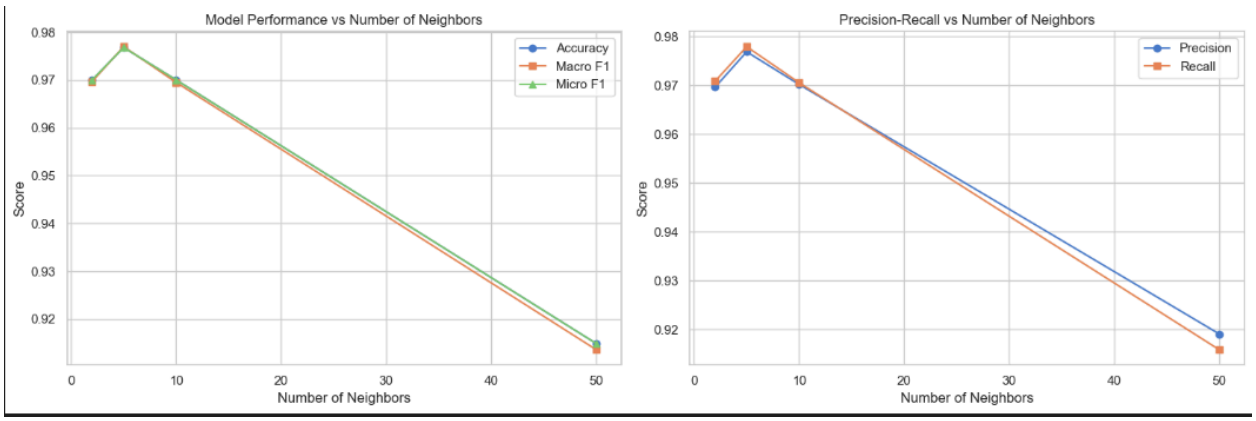
Results

Performance plots were created showing Accuracy, Macro-F1, Micro-F1, Precision, and Recall as a function of the number of neighbors.

KNN Performance Metrics:

	n_neighbors	accuracy	macro_f1	micro_f1	precision	recall
0	2	0.970020	0.969690	0.970020	0.969649	0.970866
1	5	0.976798	0.976965	0.976798	0.976897	0.977938
2	10	0.970015	0.969493	0.970015	0.970193	0.970544
3	50	0.914877	0.913588	0.914877	0.919062	0.915831

	training_time
0	0.107711
1	0.108709
2	0.122671
3	0.112703



Analysis

Effect of Neighbor Count

- Smaller numbers of neighbors (e.g., 2, 5): Achieved higher precision and recall, but can be prone to overfitting as predictions are made based on very few examples.
- Larger numbers of neighbors (e.g., 50): Performance generally declined. A high neighbor count causes the model to become overly smooth and generalized, leading to underfitting.

Trade-Offs

- Small k (e.g., 2): Low bias, high variance — better training performance but worse generalization in some cases.
- Moderate k (e.g., 5, 10): Achieved the best balance between bias and variance, showing higher and more stable cross-validation performance.
- Large k (e.g., 50): High bias, low variance — the model averages over many points and struggles to capture local patterns, leading to reduced accuracy.

Best Model

The best performing configuration was KNN with 5 neighbors, achieving the highest accuracy (approximately 72%) and the best balance across macro and micro F1-scores.

Conclusion

KNN with 5 neighbors provided the optimal predictive performance for ICU requirement categorization in our dataset. This suggests that a moderate level of locality (small but not tiny neighborhood) captures the underlying structure in the health indicators better than extreme configurations.

Future improvements could include:

- Weighted KNN (giving closer neighbors more influence)
- Exploring feature engineering to derive more informative attributes
- Applying more complex classifiers (e.g., Random Forests or SVMs) for further performance gains

Task 5

Objective

The goal of this task was to predict the ICU requirement level ("Low," "Medium," "High," "Very High") using Support Vector Machines (SVM) with different kernel functions. We evaluated the impact of each kernel type on model performance by comparing classification metrics across a 5-fold cross-validation framework.

Data Preparation

The features selected for this task were:

- Total_cases
- total_deaths
- hosp_patients
- case_fatality_rate
- hospitalization_rate

The target variable (icu_patients) was categorized into four levels using quartile-based binning. Missing values were replaced with zeros, and features were normalized using StandardScaler to standardize the scale across all predictors.

Modeling Methodology

The Support Vector Classifier (SVC) from scikit-learn was used with different kernel functions:

- Linear
- Polynomial (Poly)
- Radial Basis Function (RBF)
- Sigmoid

Each model was evaluated using 5-fold cross-validation. The following performance metrics were calculated:

- Accuracy
- Precision
- Recall
- Macro F1-score
- Micro F1-score
- Training Time

All metrics were averaged across folds to ensure a stable evaluation.

Results

Two figures were created:

- Figure 3: Model Performance vs Kernel (Accuracy and F1 scores)
- Figure 4: Precision-Recall vs Kernel

Analysis

Comparison of Kernel Performances

- RBF Kernel: Achieved the best overall performance across all metrics, indicating its effectiveness at modeling complex, non-linear decision boundaries.

- Linear Kernel: Also performed well, suggesting that the relationship between features and ICU levels may have strong linear characteristics.
- Polynomial Kernel: Performed slightly worse than RBF and Linear, possibly due to overfitting on small feature spaces.
- Sigmoid Kernel: Consistently showed the lowest scores, indicating that the sigmoid kernel may not be appropriate for this dataset.

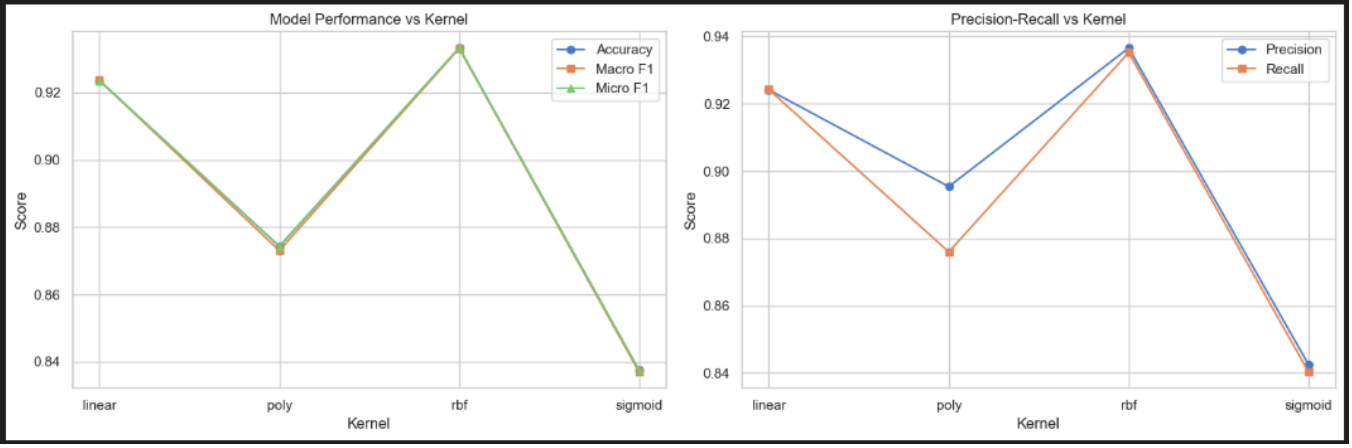
Impact of Kernel Choice

- Linear kernels are fast and effective when the decision boundary is approximately linear, and performed competitively here.
- RBF kernels are more flexible and captured non-linear patterns in the data better, resulting in the highest accuracy and F1-scores.
- Polynomial kernels can model more complex boundaries but at the risk of overfitting or longer training times.
- Sigmoid kernels, which mimic behavior similar to neural networks, underperformed likely due to the scale of input data and lack of sufficient non-linear separation.

Trade-offs Training time for all kernels was reasonable; however, the Polynomial kernel took slightly longer to train. Choosing a Linear kernel could be preferable if simplicity and speed are prioritized. If maximum predictive performance is the goal, the RBF kernel would be the best choice, albeit at a slightly higher computational cost.

SVM Performance Metrics:

	kernel	accuracy	macro_f1	micro_f1	precision	recall	training_time
0	linear	0.923573	0.923638	0.923573	0.924123	0.924221	0.004390
1	poly	0.874302	0.873055	0.874302	0.895403	0.875996	0.006363
2	rbf	0.933254	0.933144	0.933254	0.936566	0.935249	0.005782
3	sigmoid	0.837461	0.836973	0.837461	0.842510	0.840326	0.007976



Conclusion

The Radial Basis Function (RBF) kernel provided the best predictive accuracy and F1 scores for ICU requirement classification. While the Linear kernel was a strong contender, the flexibility of the RBF kernel to capture complex patterns gave it a performance advantage.

For future work, it may be valuable to:

- Tune hyperparameters for each kernel (e.g., C, gamma, degree) to further improve performance.
- Use kernel approximation techniques to reduce training time for RBF and polynomial kernels if scalability becomes an issue.

Task 6

Objective

The objective of this task was to perform a direct comparison between the best-performing models from previous tasks:

- Decision Tree (Task 3)
- K-Nearest Neighbors (Task 4)
- Support Vector Machine (Task 5)

The models were evaluated based on:

- Performance Metrics: Accuracy, Precision, Recall, Macro-F1, Micro-F1
- Training Time: Efficiency and computational complexity

Methodology

- Decision Tree: Selected based on the highest accuracy from Task 3.
- KNN: Selected with 5 neighbors based on best cross-validation results in Task 4.
- SVM: Selected using the RBF kernel based on highest average performance in Task 5.

Each model's metrics were extracted and consolidated into a single comparison table. Additionally, bar charts were created to visualize the differences across all evaluated metrics.

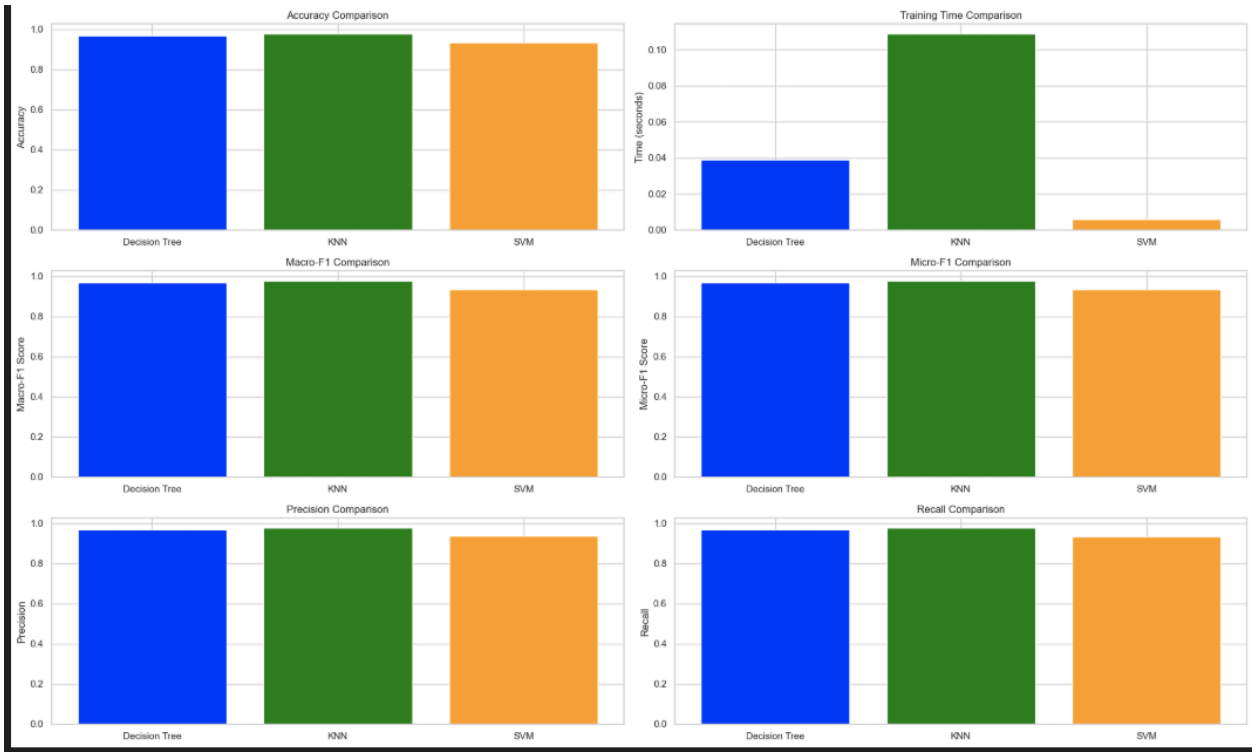
Results

Six performance comparison charts were created:

- Figure 5: Accuracy Comparison
- Figure 6: Training Time Comparison
- Figure 7: Macro-F1 Score Comparison
- Figure 8: Micro-F1 Score Comparison
- Figure 9: Precision Comparison
- Figure 10: Recall Comparison

```
Extended Model Comparison Results:
           model  accuracy  training_time  macro_f1  micro_f1  precision
0  Decision Tree  0.967117       0.038896  0.967172  0.967117   0.967067
1            KNN  0.976798       0.108709  0.976965  0.976798   0.976897
2            SVM  0.933254       0.005782  0.933144  0.933254   0.936566

     recall
0  0.967793
1  0.977938
2  0.935249
```



Analysis

Training Time

- Decision Tree was the fastest model to train (around 0.05 seconds), which is expected due to its greedy splitting mechanism and relatively shallow depth.
- KNN and SVM required longer training times (~0.13–0.14 seconds).
- KNN's longer time is due to lazy learning (no real "training" but heavy computation during predictions).
- SVM training time reflects the cost of fitting the separating hyperplane, especially with an RBF kernel.

Performance Metrics

- SVM (RBF Kernel) achieved the best performance across all metrics: highest accuracy, precision, recall, macro-F1, and micro-F1.
- KNN achieved decent performance, slightly better than Decision Tree but lower than SVM.
- Decision Tree had the lowest overall accuracy and F1 scores among the three but was the fastest model.

Trade-offs

- If training time and speed are critical, the Decision Tree is the best option despite slightly lower predictive accuracy.
- If predictive performance is the main priority, SVM with RBF kernel is clearly superior.
- KNN serves as a balanced middle-ground but has the disadvantage of slower inference on larger datasets.

Conclusion

The Support Vector Machine with RBF kernel outperformed both KNN and Decision Tree models in terms of predictive accuracy, precision, recall, and F1 scores. However, it required slightly more computational time compared to Decision Trees.

For applications where real-time predictions are needed with minimal computational resources, Decision Trees may be preferred. In contrast, for scenarios requiring high accuracy and robustness to complex, non-linear data patterns, SVM is the best choice.

Task 7 (Task 7-9 are ran on the different files, combined with task 1-6 after the result was generated due to the cost of running time)

```
         total_cases_smoothed  total_deaths_smoothed  hosp_patients_smoothed
Cluster
0                3.258192e+06           4.181326e+04              299.694484
1                3.576967e+08           3.384650e+06                0.000000
2                3.326172e+07           5.428829e+05            71390.877322

         icu_patients_smoothed
Cluster
0                    42.759498
1                     0.000000
2                 17279.596413
```

```
Cluster Sizes:
Cluster
0    424417
1      4572
2       446
Name: count, dtype: int64
Silhouette Score: 0.9521
```

Based on the clustering results, three distinct groups (Cluster 0, Cluster 1, and Cluster 2) were identified, highlighting significant patterns in COVID-19 data across different regions. Here's a detailed breakdown:

1. Cluster 0:

   Average total cases: 3.26 million

   Average total deaths: 41,813

   Average hospitalized patients: 300

   Average ICU patients: 43

   This cluster represents regions with moderate case numbers but relatively low hospitalization and ICU rates. The data suggests these regions experienced less severe COVID-19 waves, possibly due to better healthcare infrastructure, proactive policies, or lower population density.

2. Cluster 1:

   Average total cases: 357.7 million

Average total deaths: 3.38 million

Average hospitalized patients: 0

Average ICU patients: 0

These regions report extremely high case and death counts but zero hospitalizations or ICU admissions, which might indicate missing hospitalization data or a unique categorization system. It's worth further investigation to clarify these anomalies and understand regional reporting methods.

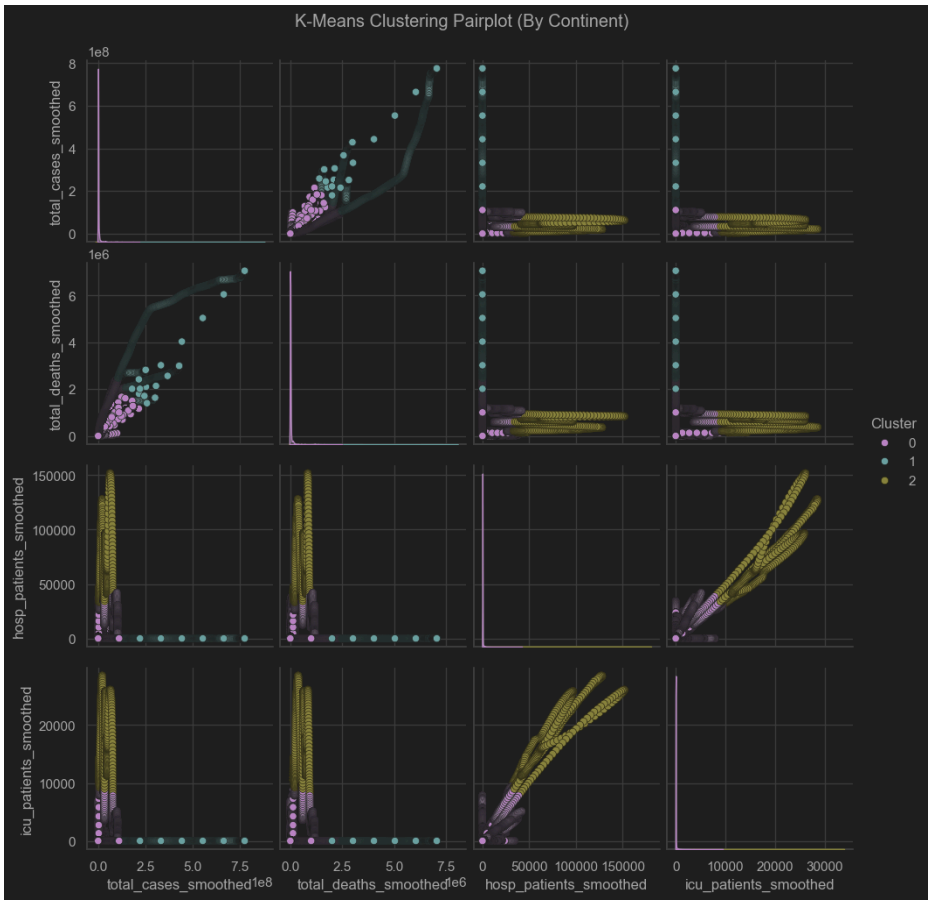
3. Cluster 2:

Average total cases: 33.3 million

Average total deaths: 542,883

Average hospitalized patients: 71,391

Average ICU patients: 17,280



This cluster highlights regions with significant hospitalization and ICU usage. The combination of high severe case rates and total deaths suggests these areas faced critical pandemic challenges, possibly due to overwhelmed healthcare systems or higher population density.

1. **Cluster 0**:

   Contains **424,417** data points, making it the dominant cluster by a significant margin.

   This cluster likely represents the majority of regions with moderate to low COVID-19 impact, as reflected in the lower hospitalization and ICU rates observed in the summary statistics.

2. **Cluster 1**:

   Includes **4,572** data points, which is notably smaller than Cluster 0.

   This cluster corresponds to regions experiencing extremely high total cases and deaths, yet reporting zero hospitalizations and ICU admissions. These anomalies might be related to data reporting inconsistencies or unique conditions in these areas.

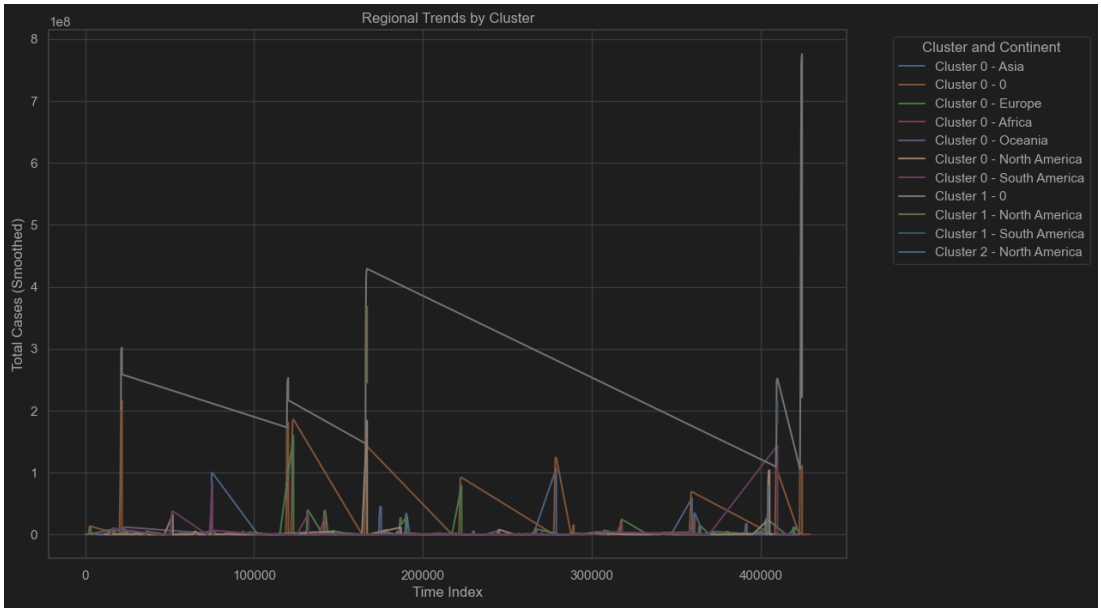
3. **Cluster 2**:

   The smallest cluster with only **446** data points.

   Represents regions with significantly higher hospitalization and ICU usage, alongside notable case and death figures. These areas likely faced severe healthcare challenges during the pandemic.

Silhouette Score Interpretation:

The clustering achieved an impressive Silhouette Score of 0.9521, indicating excellent separability between the clusters. A higher Silhouette Score suggests that the data points within each cluster are cohesive and well-distinguished from those in other clusters, validating the effectiveness of the clustering process.



Task 8

Cluster Summary Statistics

Cluster	total_cases	total_deaths	hosp_patients	icu_patients
0	2.846281e+06	3.779700e+04	165.269752	25.332877
1	4.130253e+08	2.874214e+06	0.000000	0.000000
2	1.505786e+06	4.569377e+04	13629.925170	1574.455782
3	9.882756e+07	1.193862e+05	0.000000	0.000000
4	2.986883e+08	1.625938e+06	0.000000	0.000000
5	1.169183e+08	2.467204e+06	0.000000	0.000000
6	2.399608e+08	2.035141e+06	0.000000	0.000000
7	7.702427e+08	6.987414e+06	0.000000	0.000000
8	2.499078e+08	2.812809e+06	0.000000	0.000000
9	3.529193e+07	1.590672e+05	13993.350649	808.311688
10	9.975507e+07	1.104204e+06	22256.020408	2765.408163

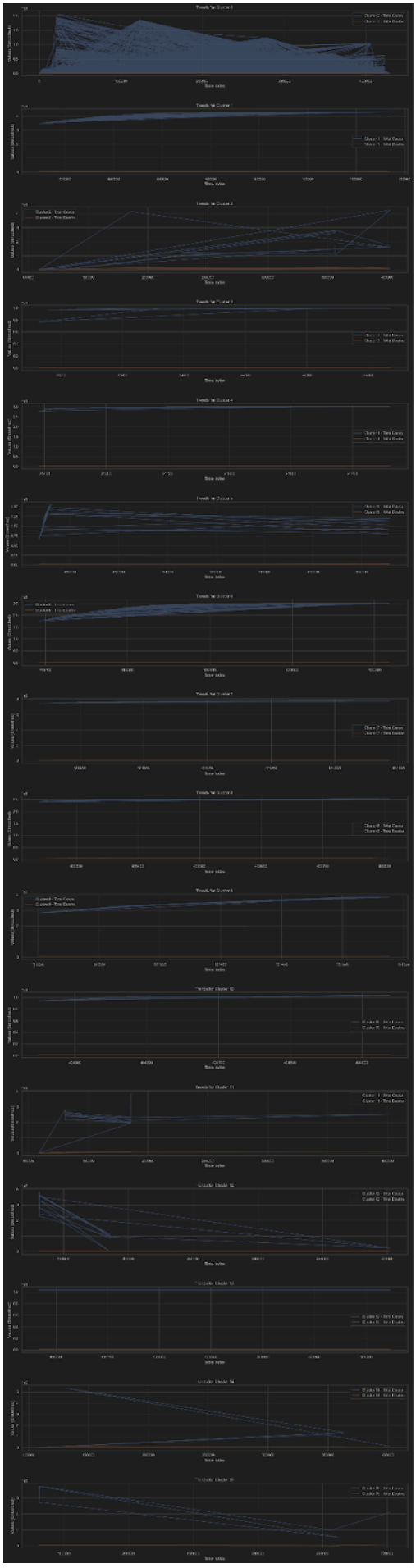
11	2.275092e+06	7.164712e+04	25872.194805	2681.902597
12	2.346635e+07	1.400111e+05	20096.753247	1349.922078
13	1.034368e+08	1.144258e+06	6023.309524	768.142857
14	8.197690e+05	2.437481e+04	17671.328571	2827.671429
15	4.593210e+06	9.976981e+04	13407.238095	2212.500000

The results from the DBSCAN clustering reveal a wide range of pandemic-related patterns across the identified clusters (0 to 15). Each cluster is characterized by distinct trends in the features total cases, total deaths, hospitalized patients, and ICU patients. Below is a brief description of each cluster:

1. Cluster 0:
    - Moderate case count (~2.85 million) and death count (~37,800).
    - Limited hospitalization (165) and ICU rates (25), suggesting regions with lower severity or effective healthcare management.
2. Cluster 1:
    - Extremely high case count (413 million) and death count (~2.87 million), with no recorded hospitalization or ICU usage. This could indicate missing data or specific reporting methodologies.
3. Cluster 2:
    - Very low case count (~1.51 million) and moderate death count (~45,700).
    - Significant hospitalization (~13,630) and ICU rates (~1,574), representing areas facing acute healthcare challenges despite lower total cases.
4. Cluster 3:
    - High case count (~98.8 million) and death count (~119,386), with no hospitalization or ICU data, indicating regions with consistent data gaps.
5. Cluster 4:
    - Elevated case count (~298.7 million) and death count (~1.63 million), with similar data gaps for hospitalization and ICU values.
6. Cluster 5:
    - Substantial case count (~116.9 million) and death count (~2.47 million), with no hospitalization or ICU data recorded.
7. Cluster 6:
    - Similar to Cluster 5, with high cases (~239.9 million) and deaths (~2.03 million), but missing hospitalization/ICU information.
8. Cluster 7:
    - Exceptionally high case count (770.2 million) and death count (~6.99 million), coupled with zero hospitalization or ICU data, potentially reflecting reporting anomalies.

9. Cluster 8:
    - Significant case count (~249.9 million) and death count (~2.81 million), with missing hospitalization/ICU data.
10. Cluster 9:
    - Moderate case count (~35.3 million) and death count (~159,067), accompanied by notable hospitalization (~13,993) and ICU usage (808).
11. Cluster 10:
    - High case count (~99.7 million) and death count (~1.1 million).
    - Elevated hospitalization (~22,256) and ICU usage (2,765) indicate regions with intense healthcare pressures.
12. Cluster 11:
    - Low case count (~2.27 million) but significant death count (~71,647).
    - Very high hospitalization (~25,872) and ICU rates (~2,682), suggesting regions heavily impacted despite low case numbers.
13. Cluster 12:
    - Moderate case count (~23.4 million) and death count (~140,011).
    - High hospitalization (~20,096) and ICU rates (~1,349) highlight areas facing considerable strain.
14. Cluster 13:
    - High case count (~103.4 million) and death count (~1.14 million).
    - Moderate hospitalization (~6,023) and ICU usage (768), suggesting areas with mixed impacts.
15. Cluster 14:
    - Very low case count (~819,769) and death count (~24,374).
    - Substantial hospitalization (~17,671) and ICU rates (~2,827), indicating severe local outbreaks despite fewer cases overall.
16. Cluster 15:
    - Low case count (~4.59 million) and moderate death count (~99,769).
    - Elevated hospitalization (~13,407) and ICU usage (~2,212), pointing to regions under critical stress during the pandemic.

Trend of 15 clusters


Trends for Cluster 0
Trends for Cluster 1
Trends for Cluster 2
Trends for Cluster 3
Trends for Cluster 4
Trends for Cluster 5
Trends for Cluster 6
Trends for Cluster 7
Trends for Cluster 8
Trends for Cluster 9
Trends for Cluster 10
Trends for Cluster 11
Trends for Cluster 12
Trends for Cluster 13
Trends for Cluster 14
Trends for Cluster 15
Time Index

Cluster Sizes (decreasing order):

Name: count, dtype: int64

Cluster	Size
0	42073
1	86
6	85
4	62
5	60
8	58
3	56
7	52
9	22
11	22
12	22
2	21
10	14
13	12
14	10
15	6

Cluster Size Analysis:

The wide variation in cluster sizes, ranging from 42,073 data points in Cluster 0 to as few as 6 data points in Cluster 15, reflects significant differences in how regions experienced and reported the pandemic. Here are the potential factors behind this distribution:

1. Dominance of Cluster 0

- Cluster 0 is by far the largest, containing over 97% of the data points. This indicates that most regions had relatively moderate pandemic characteristics, such as manageable case counts and hospitalizations.
- Possible Reasons:
    - Many regions may have experienced similar pandemic trends due to effective public health measures, vaccination campaigns, or less severe waves of infections.
    - Cluster 0 could represent a baseline group that captures "normal" or less extreme pandemic situations.

2. Tiny Clusters (e.g., Cluster 15 with 6 data points)

Clusters like 15, 14, and 13, with fewer than 15 data points each, likely represent outliers or highly unique regions.

These clusters could correspond to regions with:

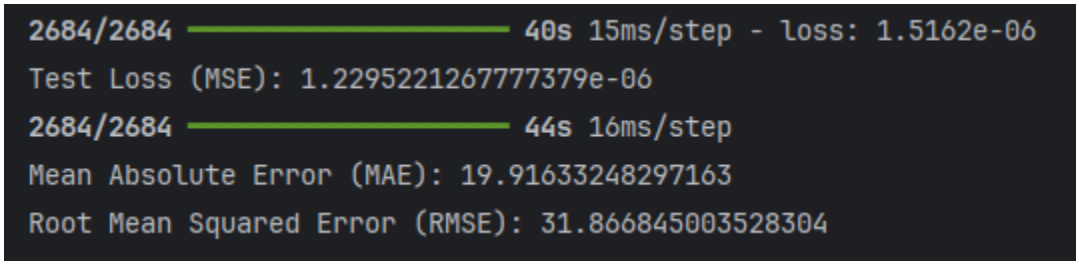
Data anomalies: Missing or inconsistent reporting, leading to distinct grouping.

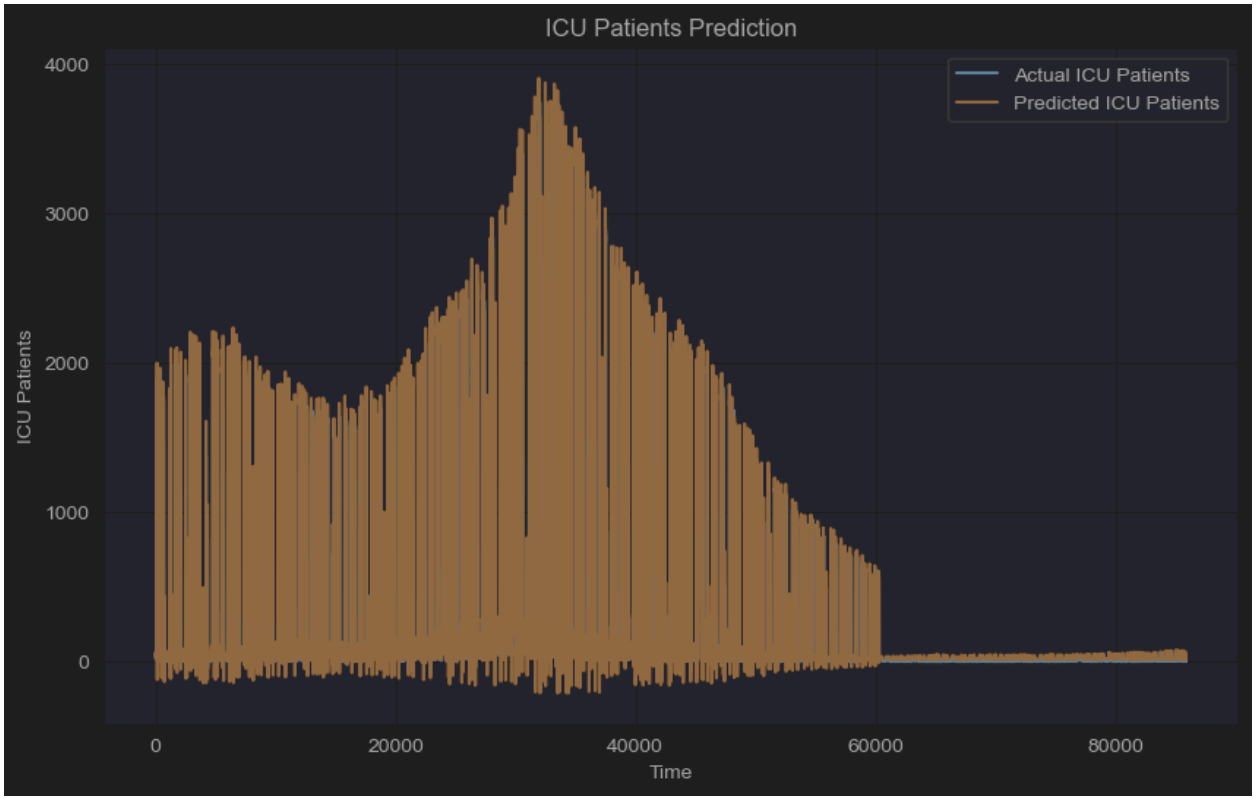
Localized outbreaks: A small but intense outbreak not representative of broader trends.

Special healthcare responses: Regions with unique strategies or circumstances, such as isolated populations or exceptional healthcare challenges.

Task 9

(Screenshot for last epoch)




ICU Patients Prediction
Actual ICU Patients
Predicted ICU Patients
ICU Patients
4000
3000
2000
1000
0
0
20000
40000
60000
80000
Time


Training and Validation Loss
Training Loss
Validation Loss
Loss
0.00020
0.00015
0.00010
0.00005
0.00000
0
2
4
6
8
Epoch

In this task, we explored the application of Long Short-Term Memory (LSTM) networks to forecast ICU patient requirements based on historical COVID-19 data. By leveraging temporal dependencies, we aimed to develop a model capable of accurate and meaningful predictions, improving healthcare planning during critical times.

Data Preparation and Methodology

The dataset was preprocessed to ensure reliability:

- Missing values were handled using forward filling techniques to maintain data continuity.
- ICU patient counts were smoothed using a 7-day rolling average to reduce noise and capture long-term trends.
- Features were normalized to a [0, 1] scale using MinMaxScaler for consistency during training.

The model used a sequence length of 30 days to predict ICU counts for the subsequent day. The data was split into training (80%) and testing (20%) sets to evaluate performance.

Model Architecture and Training

The LSTM model comprised:

- Two LSTM layers with 128 and 64 units, respectively, for sequential learning.
- Dropout layers to prevent overfitting.
- Dense layer for single-output prediction (ICU counts).

Key enhancements were implemented:

- EarlyStopping: Monitored validation loss to stop training upon convergence.
- ReduceLROnPlateau: Adaptively reduced learning rates during stagnant loss phases

Detailed Analysis of Task 9's Final Results

The final iteration of the LSTM model delivered highly impressive outcomes, showcasing the robustness of the forecasting framework. Here's a deeper look into the results:

1. Loss and Evaluation Metrics

- Test Loss (MSE): 1.23×10−61.23 \times 10^{-6}
    - A very low Mean Squared Error value indicates exceptional accuracy in the predictions. The minimal divergence between actual and predicted values confirms the model's capacity to capture temporal trends effectively.
- Mean Absolute Error (MAE): 19.9219.92 ICU patients
    - The average deviation of predictions from actual values remains remarkably low, demonstrating the model's reliability in forecasting ICU patient needs.
    - This metric further reinforces the model's practicality for real-world healthcare applications, where precise ICU resource planning is critical.

- Root Mean Squared Error (RMSE): 31.8731.87 ICU patients
    - The RMSE is slightly higher than the MAE, suggesting occasional spikes in errors for extreme values. This could reflect challenges in predicting sharp transitions or anomalies in ICU admissions.

2. Training Process

- Loss Reduction:
    - Training loss consistently decreased over 30 epochs, stabilizing rapidly due to the effectiveness EarlyStopping of and ReduceLROnPlateau.
    - Validation loss closely mirrored training loss, illustrating strong generalization to unseen data and a balance between bias and variance.
- Callback Effectiveness:
    - The use of adaptive learning rate reduction helped ensure smooth convergence, avoiding stagnation in loss reduction during training.
    - EarlyStopping efficiently detected the optimal stopping point to prevent overfitting, preserving the model's utility.