Covid – 19 Dataset| Mehrvish and Anastasiya (Team 1 and 2) respectively

Overall Summary of the Contributions from Team 1 (Mehrvish), Team 2 (Anastasiya) and Team 3 (Lauren) respectively

- **Mehrvish** focuses on preprocessing, EDA, feature engineering, and traditional modeling to set the foundation.
- **Anastasiya** focuses on model evaluation, advanced modeling (LSTM), validation, and Kaggle comparisons to build on the foundation.
- **Lauren** focuses on increasing the accuracy of the LSTM model by adding XGBoost added increased feature engineering.

Note: The professor's feedback highlighted that the original dataset's scope was limited to 187 countries, making the task uninteresting due to the lack of "unseen data" for generalization. Following this guidance, the team decided to use the **COVID-19 Clean Complete** dataset instead.

Intro to the Dataset's objective – Team 1 |Mehrvish

The **COVID-19 Clean Complete** dataset, available on Kaggle with a usability rating of **10.00**, provides a comprehensive global perspective on the COVID-19 pandemic. It contains **49,068 records** across **10 columns**, offering critical insights into the progression and impact of the virus. The columns include:

- **Province/State**: Sub-regional location (e.g., states, provinces).
- **Country/Region**: The country or region's name.
- **Latitude (Lat)**: Geographic latitude.
- **Longitude (Long)**: Geographic longitude.
- **Date**: The date of the recorded data point.
- **Confirmed**: Cumulative number of confirmed cases.
- **Deaths**: Total reported deaths attributed to COVID-19.
- **Recovered**: Number of individuals who recovered from COVID-19.
- **Active**: Current active cases (confirmed cases minus deaths and recoveries).
- **WHO Region**: The World Health Organization region classification.

This data set is perfect for descriptive analysis and predictive modeling, allowing researchers to examine temporal and spatial patterns, predict increases in cases, and facilitate proactive decision-making. Having a usability score of 10.00 signifies its preparedness and excellence for machine learning activities, making it a crucial tool for effectively comprehending and controlling pandemic dynamics.

Data Preprocessing, EDA, and Initial Model Evaluation - Team 1 |Mehrvish

Summary of the Approach

In this project, I plan to analyze the provided **covid_19_clean_complete.csv** dataset to predict regions likely to experience a "surge" in COVID-19 cases and working alongside with Team 2. The approach is structured into the following steps:

Data Processing

I was in charge of cleaning and preparing the dataset for modeling in this project. I dealt with null values, converted categorical variables, and standardized numerical features. This is my strategy for tackling each stage:

- **Loading and Inspection:** I will load the **covid_19_clean_complete.csv** dataset, inspect its structure, and identify missing values or anomalies.
- **Handling Missing Values:** Missing numerical data will be addressed using forward fill to preserve temporal consistency.
- **Encoding Categorical Variables:** Categorical features like Country/Region will be label-encoded to ensure compatibility with machine learning models.
- **Feature Scaling:** Numerical features (Confirmed, Deaths, Recovered, Active) will be normalized using Min-Max scaling to ensure equal contribution to the model.

These preprocessing steps addressed inconsistencies in the dataset, ensured numerical features were comparable, and allowed categorical variables to be seamlessly integrated into machine learning models. Additionally, SMOTE was applied to address the class imbalance between 'Surge' and 'No Surge' instances, creating a balanced dataset that improved the model's ability to detect surges and reduced false negatives.

See Google Colab Reference section – Data Processing – Mehrvish

Exploratory Data Analysis (EDA)

Team 1 and 2 analyzed the distributions of key features like confirmed cases, deaths, and recoveries to understand their spread and identify patterns. We retained outliers as they represented real-world surges critical for modeling COVID-19 trends. Using a heatmap, and to our observations we found strong correlations, such as a +0.95 correlation between active and confirmed cases, which guided to do feature engineering in which Team 2 will address. This ensured the model captured temporal dependencies through lag features and rolling averages.

See Google Colab Reference section – Exploratory Data Analysis – Mehrvish and Anastasiya

Model Evaluations and Model Implementation – Team 2| Anastasiya

I evaluated the models using accuracy, confusion matrices, and classification reports. I selected three models: Logistic Regression, Random Forest, and LSTM, each serving a specific purpose to complement the others.

- **Logistic Regression:** I used this as a baseline model to provide a simple and interpretable starting point for comparison.
- **Random Forest:** I implemented this to handle non-linear patterns in the data and improve predictive accuracy over the baseline.
- **LSTM:** I leveraged the sequential nature of the dataset with LSTM to capture long-term temporal dependencies and achieve the best accuracy for predicting surges.

Feature Engineering Team 2| Anastasiya

To improve model performance, I created new features based on the patterns identified during exploratory data analysis:

- **Lag Features:** I introduced rolling averages over the past 7 and 14 days to capture recent trends, which were essential for predicting surges.
- **Week-over-Week Change:** I calculated the percentage change in daily new cases week-over-week, providing a clear indicator of potential surges.
- **Region-Specific Indicators:** I added binary indicators for regions like the Americas to account for geographical variations in case trends.

My Logistic Regression model served as a reliable baseline, demonstrating good initial performance in predicting 'No Surge' cases but it was only able to identify half of the actual “surges”. This indicates that the model was unable to capture the necessary patterns for surge detection. Random Forest showed 100% accuracy as well as the ability to characterize all of the data points correctly. This could be due to overfitting of the model.

My LSTM model, despite leveraging sequential patterns, faced similar challenges to the baseline model and failed to capture surge-specific trends. In Part 3, Lauren plans to focus on refining hyperparameters, such as the dropout rate and batch size for the LSTM model and exploring additional optimizations to enhance the model's ability to detect surges effectively.

See Google Colab Reference section – Data Processing, EDA, and Initial Model Evaluation – Anastaysia

Kaggle’s Analysis Comparison with Kaggle Author’s Work (Milestone 2 Report) – Team 2 Anastasiya

In Mehrvish’s report Milestone 2, she focused on defining the predictive task and conducting exploratory data analysis (EDA) to identify key patterns, such as a +0.95 correlation between active cases and surges. By Milestone 4, Mehrvish and I advanced to predictive modeling using the covid_19_clean_complete.csv dataset. We both created features like "Lag Confirmed" and "WoW Change" to capture temporal trends and applied models like Logistic Regression, Random Forest, and LSTM, with LSTM performing best due to its ability to leverage sequential data.

Mehrvish addressed challenges like missing values through forward-filling and resolved class imbalance using SMOTE. While Kaggle's approach emphasized visual analysis, we both extended the work by focusing on feature engineering and predictive modeling to provide actionable insights, such as early surge detection. With metrics like LSTM’s 58% accuracy, we both demonstrated the dataset’s potential for predictive tasks beyond exploratory analysis and its ability to address real-world problems effectively.

Model Optimization and Enhanced Evaluation Using XGBoost- Team 3 |Lauren

Model Evaluations and Model Implementation – Team 3 | Lauren

I evaluated the models, the same way Team 1 and Team 2 did above so that we can compare the models, using accuracy, confusion matrices, and classification reports to assess their performance in predicting COVID-19 surges. I selected two models, updating Anastasiya’s LSTM and added XGBoost, each was designed to address different aspects of the problem. I used the updated LSTM model and processed input data as sliding windows of temporal sequences so that the model captured long-term dependencies and temporal patterns, making it ideal for detecting surges over time. To complement the LSTM, I implemented XGBoost to model non-sequential patterns in the data.

Cross-validation ensured hyperparameter optimization and made sure overfitting was not occurring. Class imbalance was handled using scale_pos_weight to account for the underrepresented surge class.

Feature Engineering Team 3| Lauren

- I dropped the rows where the lag features that Anastasiya engineered are missing, to prevent errors and to increase the reliability of the model.
- I created a create_sequences function to prepare the data for the LSTM model, this reshaped the data into sequences of past observations to model the temporal dependencies better. The model would now process multiple time steps for each prediction to more easily capture sequential patterns.

Modeling Enhancements Team 3| Lauren

- I implemented XGBoost with cross- validation with early stopping and stratified folds to help with evaluation and hyperparameter tuning and reducing overfitting risk. I also included a visualization of feature importance that shows the features that had the most weight. The inclusion of a feature importance visualization added interpretability to the XGBoost results, highlighting which features contributed the most to the predictions.

- I improved upon Anastasiya's LSTM by adding another layer and adding dropout layers to prevent overfitting and training epochs were increased from 10 to 20 and included more data preprocessing.

Conclusion- Lauren

Model	Accuracy	Precision(surge/no surge)	Recall (surge/ no surge)	F1 score (surge/no surge)
Baseline Log Reg (Anastasiya)	0.788	0.99/0.74	0.49/1.00	0.65/0.85
Random Forest (Anastasiya)	1.0	1.0/1.0	1.0/1.0	1.0/1.0
First LSTM model (Anastasiya)	0.588	0.0/0.59	0.0/1.0	0.0/0.74
Second LSTM model (Lauren)	0.901	0.94/0.88	0.81/0.96	0.87/0.92
XGBoost (Lauren)	0.999	0.99/0.99	0.99/0.99	0.99/0.99

Based on the above results, the models demonstrated distinct strengths and weaknesses in predicting COVID-19 surges. The baseline Logistic Regression provided an accuracy of 0.788 but struggled with identifying surges, as reflected in its low recall of 0.49 for the surge class. The Random Forest model achieved perfect scores (1.0 across all metrics), but this could be a result of overfitting, making it less reliable for real-world applications where generalizability is critical.

The first LSTM model did not perform well, with an accuracy of 0.588 and a recall of 0.0 for the surge class, indicating it failed to capture sequential patterns in the data. However, the second LSTM model addressed these issues and achieved significant improvements, with an accuracy of 0.901 and balanced precision, recall, and F1 scores. This demonstrated its ability to effectively model the temporal relationships necessary for identifying surges, making it a strong candidate for sequential data tasks.

The XGBoost model outperformed all others, achieving near-perfect results with an accuracy of 0.999 and balanced precision, recall, and F1 scores of 0.99 for both classes. These results suggest that XGBoost successfully captured the underlying patterns in the data while avoiding overfitting, because of its robust design, careful tuning, and use of techniques like cross-validation and class balancing.

In conclusion, while the second LSTM model showed strong potential for handling sequential data, XGBoost was the most reliable and effective model for predicting COVID-19 surges. XGBoost balances performance and interpretability which makes it an ideal choice for this task. Moving forward, addressing the overfitting issues observed in Random Forest and further refining the LSTM model through hyperparameter tuning or incorporating additional features could lead to even better results. Exploring ensemble methods that combine the strengths of both XGBoost and LSTM might also enhance predictive accuracy, offering a more comprehensive approach to surge detection.

References for Team 1, 2 and 3

1. **COVID-19 Dataset :** https://www.kaggle.com/datasets/imdevskp/corona-virus-report

2. **Link to the CSV file:** https://drive.google.com/file/d/1qzOuuIMdjoWtHi1WU8QI9a8YrrnfEJ_O/view?usp=sharing (Please download for easy view)

3. **Covid-19 Forecasting Week1 using Random Forest:** https://www.kaggle.com/code/datawarriors/covid19-forecasting-week1-using-random-forest

4. **Milestone 2 Report-** https://drive.google.com/file/d/1vbs2N0OM0xtDwPWYfZJGTO9zuyp6-Ode/view?usp=sharing

5. **See Google Colab** Notebook- https://colab.research.google.com/drive/1j6goPviL65bVbJYrSNWuTSRgDwzLY5JC?usp=sharing
   Team 1 and 2 – Mehrvish and Anastasiya
   1. See Google Colab – Data Processing – Mehrvish
   2. See Google Colab – Exploratory Data Analysis – Mehrvish and Anastasiya
   3. See Google Colab – Data Processing, EDA, and Initial Model Evaluation – Anastasiya

   Team 3 – Lauren
   4. See Google Colab – LSTM Model Optimization and Enhanced Evaluation Using XGBoost – Lauren

6. **PowerPoint presentation**- https://docs.google.com/presentation/d/1DHK_-RXWNBAA_D-N7sFhgo_WS2xdQzAw/edit?usp=sharing&ouid=113915710576067604244&rtpof=true&sd=true

7. PowerPoint video https://drive.google.com/file/d/14m1kW1Z6GG_qN6CHictwtM21OWrO2wlC/view?usp=sharing