COVID-19 Symptom Analysis and Prediction System (CSAPS)

by Tao Jin, Yerbol Baizhumanov, Alsiher Aliyev

1.1 SUMMARY DESCRIPTION

The CSAPS project aims to develop a large-scale analysis system capable of detecting and classifying symptoms to predict whether a patient has COVID-19. This system will leverage machine learning techniques, integrated with Apache Spark, to process and analyze metadata associated with various symptoms and improve diagnostic accuracy and efficiency. By utilizing cluster computing platforms: Apache Spark's distributed computing capabilities, we aimed to handle large volumes of medical data in a scalable and reliable manner.

1.2 I/O EXAMPLES

Input Example 1:

- Input: Symptoms: [1,1,1,1,1,0,0,0,1,1,1,1,1,0,1,1,1,1,0,0]
- Output : Does this patient have COVID-19?: Yes

Input Example 2:

- Input: Symptoms: [1,1,1,0,1,0,0,1,1,0,0,0,0,0,0,0,0,0,0]
- Output: Does this patient have COVID-19?: No

1.3 REQUIREMENTS

1.3.1 Definite Requirements

- Implement data cleaning and normalization
- Convert categorical data to numeric
- Develop machine learning models for COVID-19 prediction
- Implement feature importance analysis
- Build scalable data pipeline

1.3.2 Requirements Not Classified Yet

- Provide confidence scores for each prediction
- Multi-language support for international data
- Integration with external health databases

1.3.3 Nice-to-do Requirements

- Support real-time processing of COVID-19 data
- Provide explanatory analysis of its predictions

- Interactive visualization dashboards
- Automated report system

1.4 HOW SUCCESS WILL BE ASSESSED

Success will be measured using the following metrics:

1. Classification Accuracy: Minimum 85% accuracy on a held-out test set of 5500 patients data
2. Sensitivity and Specificity: Minimum 90% sensitivity for COVID-19 detection
3. Computational Performance Metrics:
    1. Training Time: Maximum 4 hours on specified AWS cluster configuration
    2. Inference Time: Maximum 100ms per patient record
    3. Throughput: Minimum 1000 patient records per minute in batch processing
4. ROC-AUC Score: Minimum 0.85
5. F1-Score: Minimum 0.85

1.5 TECHNOLOGY EXPLANATION

Primary Implementation:

- Apache Spark MLlib for distributed machine learning
- Ensemble Learning Approach:
    - Gradient Boosting Decision Trees (using Spark's GBT implementation) for handling complex feature interactions between symptoms
    - Random Forest for capturing non-linear relationships and handling missing data
    - Custom stacking implementation to combine model predictions
- AutoML components for hyperparameter optimization

Alternative Approach:

- Deep Learning approach using Wide & Deep Neural Networks:
    - Wide component: Linear model for memorization of feature interactions
    - Deep component: Neural network for generalization of feature patterns
- Feature importance analysis using SHAP (SHapley Additive exPlanations)
- Online learning approach for continuous model updates

The choice of ensemble methods over deep learning as the primary approach is motivated by:

1. Better interpretability (crucial for medical applications)

2. Robust handling of missing values (common in medical data)
3. Natural feature importance ranking
4. Lower computational requirements for training
5. Better performance on tabular data compared to deep learning approaches

1.6 DATA SOURCES

The project will utilize a dataset containing records of patient symptoms and whether or not they have the COVID-19 virus:

1. Breathing Problem
2. Fever
3. Dry Cough
4. Sore throat
5. Running Nose
6. Asthma
7. Chronic Lung Disease
8. Headache
9. Heart Disease
10. Diabetes
11. Hyper Tension
12. Fatigue
13. Gastrointestinal
14. Abroad travel
15. Contact with COVID Patient
16. Attended Large Gathering
17. Visited Public Exposed Places
18. Family working in Public Exposed Places
19. Wearing Masks
20. Sanitization from Market
21. Patient has COVID-19 or not

The dataset is publicly available and well-documented, with proper annotations and associated metadata. It consists of 5,434 records, which will be divided into two reasonable sized files: one for training and the other for testing. The input data uses a 'yes' and 'no' schema, which will be converted into a binary schema of ‘1’ and ‘0’.

1.8 REFERENCES

[1] Alazab, M., Awajan, A., Mesleh, A., & Alhyari, S. (2020). COVID-19 prediction and detection using deep learning. *International Journal of Computer Information Systems and Industrial Management Applications*, *12*, 14-14.

[2] Solayman, S., Aumi, S. A., Mery, C. S., Mubassir, M., & Khan, R. (2023). Automatic COVID-19 prediction using explainable machine learning techniques. *International Journal of Cognitive Computing in Engineering*, *4*, 36-46.

[3] Ardabili, S. F., Mosavi, A., Ghamisi, P., Ferdinand, F., Varkonyi-Koczy, A. R., Reuter, U., ... & Atkinson, P. M. (2020). Covid-19 outbreak prediction with machine learning. *Algorithms*, *13*(10), 249.

[4] Zoabi, Y., Deri-Rozov, S., & Shomron, N. (2021). Machine learning-based prediction of COVID-19 diagnosis based on symptoms. *npj digital medicine*, *4*(1), 1-5.