COVID-19 Symptom Analysis and Prediction System (CSAPS)

- BY Tao Jin, Yerbol Baizhumanov, Alsiher Aliyev

Overview

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. It uses three machine learning models: Random Forest, Gradient Boosting Trees, and Logistic Regression. The system is built to scale efficiently for large datasets using Apache Spark. 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.

Features

1. Data Processing:

- Reads a CSV dataset containing symptom data.
- Converts categorical columns (`Yes/No`) into numeric values (`1/0`).
- Extracts symptoms into a feature vector for model training.

2. Modeling:

- Utilizes three machine learning algorithms:
  - Random Forest Classifier
  - Gradient Boosting Classifier
  - Logistic Regression
- Pipelines streamline the process of feature transformation, target indexing, and model fitting.

3. Evaluation:

- Evaluates models using:
  - Binary Classification Metrics: Area Under ROC (AUC)
  - Multiclass Metrics: Accuracy
- Prints feature importances for Random Forest and coefficients for Logistic Regression.

4. Prediction:

- Provides predictions along with confidence scores for all three models.
- Accepts symptom inputs for real-time predictions.

Dependencies

- Use AWS to create cluster or use Google Cloud, use PySpark and Python

Usage

Step 1: Data Preparation

The dataset should be in CSV format with columns representing symptoms (e.g., `Breathing Problem`, `Fever`, etc.) and the target column (`COVID-19`).

Example column headers:

Breathing Problem, Fever, Dry Cough, Sore throat, Running Nose, Asthma, ..., COVID-19

Step 2: Running the Script

Run the script from the command line, passing the path to your dataset as an argument:

```
python covid_prediction_system.py <path_to_dataset>
```

Step 3: Example Predictions

The script includes examples for predicting COVID-19 status based on user-provided symptom data. Example input arrays should match the order of symptoms in the dataset.

Code Structure

Class: `COVIDPredictionSystemSpark`

This class implements the end-to-end pipeline for the prediction system.

1. __init__:

- Reads and preprocesses the dataset.
- Converts categorical symptom data to numeric.

2. prepare_data:

- Splits data into training and testing sets.
- Creates a feature vector and indexes the target variable.

3. create_model_pipeline:

- Defines pipelines for Random Forest, Gradient Boosting Trees, and Logistic Regression.

4. train_and_evaluate_models:

- Trains each model.
- Evaluates performance metrics (AUC, accuracy).
- Outputs feature importances and coefficients.

5. predict:

- Provides predictions and confidence scores for new symptom data.

Example Output

Sample output from training and predictions:

```
Random Forest Metrics:

AUC: 0.9922121117446432

Accuracy: 0.9665071770334929



Feature Importances for Random Forest:

Breathing Problem: 0.1944

Fever: 0.0749

Dry Cough: 0.1710

Sore throat: 0.1654

Running Nose: 0.0037

Asthma: 0.0031

Chronic Lung Disease: 0.0022

Headache: 0.0012

Heart Disease: 0.0024

Diabetes: 0.0018

Hyper Tension: 0.0034

Fatigue : 0.0017

Gastrointestinal : 0.0028

Abroad travel: 0.1879

Contact with COVID Patient: 0.0672
```

```
Attended Large Gathering: 0.1076
Visited Public Exposed Places: 0.0015
Family working in Public Exposed Places: 0.0080
Wearing Masks: 0.0000
Sanitization from Market: 0.0000

Gradient Boosting Metrics:
AUC: 0.9990594652295035
Accuracy: 0.9866028708133971

Logistic Regression Metrics:
AUC: 0.9899339998437865
Accuracy: 0.9760765550239234

Logistic Regression Coefficients:
Breathing Problem: -1.7212
Intercept: 5.0992
Fever: -1.6238
Intercept: 5.0992
Dry Cough: -1.8908
Intercept: 5.0992
Sore throat: -1.6755
Intercept: 5.0992
Running Nose: 0.3420
Intercept: 5.0992
Asthma: -0.0846
```

Intercept: 5.0992

Chronic Lung Disease: 0.0455

Intercept: 5.0992

Headache: 0.0320

Intercept: 5.0992

Heart Disease: -0.0446

Intercept: 5.0992

Diabetes: -0.1612

Intercept: 5.0992

Hyper Tension: -0.1730

Intercept: 5.0992

Fatigue : 0.0497

Intercept: 5.0992

Gastrointestinal : -0.0041

Intercept: 5.0992

Abroad travel: -2.7918

Intercept: 5.0992

Contact with COVID Patient: -1.4489

Intercept: 5.0992

Attended Large Gathering: -2.5702

Intercept: 5.0992

Visited Public Exposed Places: -0.1258

Intercept: 5.0992

Family working in Public Exposed Places: -0.6342

Intercept: 5.0992

Wearing Masks: 0.0000

Intercept: 5.0992

Sanitization from Market: 0.0000

Intercept: 5.0992

Example Predictions:

Example 1 Prediction: {'random_forest_prediction': 'No', 'random_forest_confidence': 0.03318306460384017, 'gradient_boosting_prediction': 'No', 'gradient_boosting_confidence': 0.013263930924715228, 'logistic_regression_prediction': 'No', 'logistic_regression_confidence': 0.0013825968442934267}

Example 2 Prediction: {'random_forest_prediction': 'No', 'random_forest_confidence': 0.40689667640422583, 'gradient_boosting_prediction': 'Yes', 'gradient_boosting_confidence': 0.6871890561092804, 'logistic_regression_prediction': 'Yes', 'logistic_regression_confidence': 0.5480369880970453}

Key Considerations

- The script assumes clean data with consistent column names.
- Model hyperparameters (e.g., tree depth, regularization) can be adjusted for performance tuning.
- Could use neural network implementation to increase efficiency.