Practical Report 1.2

Group 8: Shiyao Gu, Fulin Zhang, Ruicong Wang

Introduction

This report outlines the process and results of predicting user sentiments from Australia users' tweets related to COVID-19. The task involves two scenarios: predicting sentiments using known features and predicting sentiments using features generated by regression models. The following sections provide a detailed overview of the methodologies, results, and analyses for each scenario.

Code Overview and Data Processing

Data Preparation

- **Data Loading**: Loads multiple CSV files from a specified directory, ensuring each CSV corresponds to sentiment data collected on different days. This method is beneficial for managing and analyzing time-series data where each observation is tied to a specific date.
- **Date Parsing**: Extracts the date from the filenames, which is essential for analyzing trends over time and ensuring the data aligned correctly with the study's timeline.
- **Data Concatenation**: Combines all individual data frames into a single comprehensive data frame, making it easier to apply transformations and models to the complete dataset. It's also beneficial to split the training data and validation data base on date.

Create new features: Moving Average Calculation

- **Moving Average Calculation**: Computes the rolling mean for various sentiment intensities ('valence_intensity', 'fear_intensity', 'anger_intensity', 'happiness_intensity', and 'sadness_intensity') over a 7-day window for each user. The moving averages smooth out short-term fluctuations and highlight longer-term trends in sentiment, which can be crucial for understanding user behavior over time. These new features (rolling means) can be very useful for machine learning models, as they provide additional context that isn't available from just the raw sentiment scores. This can potentially improve model accuracy, especially in predicting future sentiments.

Data Splitting and Model Training

Splitting the Data: Ensures that data is sorted by date, which is essential for time-series data to maintain chronological order, critical for rolling calculations and splitting the dataset. Defines the validation set as the last 5 days of data and the remaining as the training set. This separation method is typical in time-series forecasting, where the model's predictive performance is tested on the latest data, which it hasn't seen during training.

Scenario 1: Using Known Features
Model Setup and Evaluation

- **Feature Selection**: Includes a set of features, both raw intensities and their corresponding rolling means, capturing both immediate sentiment and its trend over time.
- **Model Training**: Utilizes a **DecisionTreeClassifier** for training, which can handle non-linear relationships.

Performance Metrics

The model excels in identifying negative and positive sentiments, as evidenced by high precision and recall values in these classes. The overall accuracy and weighted average F1-score are quite high, indicating robust performance across the dataset. The model struggles relatively more with neutral sentiments. This is indicated by the lower recall and F1-score for class 0, suggesting some neutral sentiments are either missed or misclassified as positive or negative. This could be due to the inherent challenge of distinguishing neutral sentiment, which might not exhibit strong indicators like positive or negative sentiments.

	precision	recall	f1-score	support
-1	0.94	0.98	0.96	18349
0	0.85	0.75	0.80	5212
1	0.96	0.95	0.95	4931
accuracy			0.93	28492
macro avg	0.92	0.89	0.90	28492
weighted avg	0.93	0.93	0.93	28492

The RandomForest model has shown a robust performance across all classes, with particularly impressive improvements in the precision for neutral sentiments. The recall for negative sentiments at 1.00 is a highlight, indicating that the model effectively captures all negative instances without any false negatives. While there's significant improvement in precision for neutral sentiments, the recall remains relatively lower at 0.70. This suggests that while most of the predictions for neutral are correct, the model still misses a number of neutral instances.

	precision	recall	f1-score	support
-1	0.93	1.00	0.96	18349
0	0.97	0.70	0.81	5212
1	0.95	0.99	0.97	4931
accuracy			0.94	28492
macro avg	0.95	0.89	0.91	28492
weighted avg	0.94	0.94	0.94	28492

Scenario 2: Using Generated Features
Model Setup for Feature Generation

- **Feature Standardization**: Standardizes features using StandardScaler to ensure equal contribution to the model, essential for models sensitive to input data scales like RandomForest.
- **Feature Prediction**: Trains a **RandomForestRegressor** for each sentiment-related feature, leveraging its ability to handle non-linear relationships and provide high accuracy.
- **Time as a Feature**: Uses the Unix timestamp derived from the date as the sole feature for predicting sentiment-related features, treating the problem as a time-series forecasting issue.

Implementation Steps

1. **Date and User ID Preparation**: Creates combinations of user IDs and future dates for which sentiment-related features will be predicted.
2. **Feature Generation Using Regressors**: Uses previously trained **RandomForest regressors** for each feature to predict values based on future dates.
3. **Scaling Back to Original**: Scales the predicted features back to their original scale using the inverse of the initial standardization.

Comparative Analysis of Classifier Performance: Scenario 1 vs. Scenario 2
Scenario 1: Using Known Features

	precision	recall	f1-score	support
-1	0.94	0.98	0.96	18349
0	0.85	0.75	0.80	5212
1	0.96	0.95	0.95	4931
accuracy			0.93	28492
macro avg	0.92	0.89	0.90	28492
weighted avg	0.93	0.93	0.93	28492

Scenario 2: Using Generated Features

	precision	recall	f1-score	support
-1	0.94	0.98	0.96	18349
0	0.85	0.75	0.80	5212
1	0.96	0.95	0.95	4931
accuracy			0.93	28492
macro avg	0.92	0.89	0.90	28492
weighted avg	0.93	0.93	0.93	28492

Insights and Analysis

1. **Overall Accuracy**:
    - The overall accuracy is identical (0.93) for both scenarios, indicating that the generated features are as effective as the known features in predicting sentiments.
2. **Class-specific Performance**:
    - **Negative Sentiment (-1)**: Both scenarios show identical high performance, indicating robust performance in identifying negative sentiments.
    - **Neutral Sentiment (0)**: Both scenarios show identical performance metrics, suggesting that the model's ability to capture neutral sentiments remains consistent.
    - **Positive Sentiment (1)**: Both scenarios also show identical performance metrics, reinforcing the classifier's effectiveness in predicting positive sentiments using generated features.

The generated features are effective, as evidenced by the identical performance metrics across both scenarios. This suggests that the regression models for feature generation are well-tuned and capable of capturing essential characteristics needed for sentiment prediction. The classifier model is robust, demonstrating high and consistent performance across both scenarios. This is a positive indicator for practical applications where real-time or future data might not have all features readily available.