[Applied Analytics Project Week 12 Assignment]

Major: Applied Analytics

Name: Heng Bai, Yuang Guo

Document your environment dependencies. Specifically: OS, Python version, Python packages used, and their versions

The model is developed and executed in a Google Colab environment, which runs on a Linux-based operating system. The Python version used is Python 3.10. A number of Python packages are essential for the model's development and deployment: xgboost (1.6.2) for model training and evaluation, scikit-learn (1.1.3) for metrics and data splitting, joblib (1.2.0) for saving and loading models efficiently, pandas (1.3.5) and numpy (1.21.6) for data manipulation and numerical computations, nltk (3.8) for text preprocessing and tokenization, gensim (4.2.0) for building Word2Vec embeddings, and tensorflow (2.11.0) as a fallback for advanced deep learning tasks. These dependencies ensure smooth execution of the codebase, covering all aspects of preprocessing, modeling, and evaluation. It's critical to document these packages with their respective versions to guarantee reproducibility across environments.

Discuss if your model will be deployed in batch mode or using real-time inference and why you chose one over the other

This model is best suited for deployment in batch mode rather than real-time inference. The primary reason is the computational cost associated with generating Word2Vec embeddings and training a classification model like XGBoost. Batch mode allows the processing of large datasets in a scheduled or triggered manner without the need for immediate results, making it ideal for applications such as sentiment trend analysis or periodic reporting. Conversely, real-time inference would require optimized pipelines and faster embedding generation, which may not align with the current implementation's computational demands. However, real-time inference could be considered in the future for applications like live tweet analysis during events, provided the architecture is modified for low latency.

Identify performance metrics (model, business, other) that will be tracked in the monitoring plan. Why are these important to track in production?

The performance of the model is measured using a combination of metrics. From a modeling perspective, metrics like accuracy, precision, recall, and F1-score are employed to evaluate the classification model's effectiveness. Additionally, log loss is monitored to assess the probabilistic confidence of the predictions. From a business perspective, metrics such as the turnaround time for predictions and the cost of incorrect predictions are crucial. For example, misclassifications in sentiment analysis could lead to incorrect strategic decisions. Tracking these metrics ensures the model's reliability, provides transparency into its performance, and aligns the technical outcomes with business goals. Focusing on these metrics helps maintain the balance between operational efficiency and model accuracy in production.

Identify thresholds corresponding to green (nothing is wrong with the model), yellow (errors should be tracked closely), and red (model should be pulled out of production) for each selected metric

Performance thresholds are set to identify when the model is operating effectively or needs intervention. In the "green" zone, the model achieves accuracy above 90% and an F1-score of at least 85%, indicating optimal performance. In the "yellow" zone, accuracy between 80% and 90% or an F1-score below 85% signals that the model's predictions may not be as reliable, warranting close monitoring and potential tuning. The "red" zone is defined by accuracy dropping below 80% or a significant increase in log loss, which signifies the need to pull the model from production and investigate the cause. These thresholds help maintain the model's performance while avoiding adverse effects on downstream tasks.

Discuss risk mitigation strategies for green, yellow and red flags

Risk mitigation strategies are essential for ensuring smooth operation in production. In the “green” zone, routine monitoring and scheduled evaluations should continue without any urgent actions. In the “yellow” zone, a deeper analysis of potential issues such as shifts in input data, changes in user behavior, or outdated feature engineering processes should be conducted. Hyperparameters may need retuning, or the model architecture might require optimization. For “red” zone scenarios, the model should be immediately removed from production to prevent further incorrect predictions. A backup or older stable version of the model can temporarily take over while a root-cause analysis is performed to identify and resolve the issues.

Discuss if and how frequently you would retrain your model and why?

Retraining the model periodically is critical, especially for tasks like sentiment analysis where language usage evolves rapidly. A bi-weekly retraining schedule is recommended to capture new trends, such as emerging slang or new hashtags in tweets. Additionally, retraining should be triggered whenever significant data drift or concept drift is detected. Frequent retraining ensures that the model remains relevant and performs well on the latest data, preventing degradation of its predictive accuracy. By proactively updating the model, businesses can avoid risks associated with stale predictions, thereby maintaining reliability and user trust.

Discuss if the data for your use case can be impacted by data drift, or concept drift and how you’d mitigate these risks

Data drift and concept drift are significant risks for this use case, as the input data (e.g., tweets) and their relationship with target labels can change over time. For example, the frequency of certain words or the sentiment associated with them might evolve, leading to a mismatch between the training data and real-world inputs. To mitigate data drift, regular monitoring of input data distributions should be implemented, along with retraining the model on fresh data. Concept drift can be managed by updating the training set with newly labeled data that reflects the current context. Combining active learning techniques with regular evaluations ensures that the model adapts to changes and continues to provide accurate predictions.