BDM 3035: BIG DATA ANALYTICS CAPSTONE PROJECT

Submitted to: MEYSAM EFFATI

Submitted on: 2024-07-29

SMS SPAM CLASSIFIER MILESTONE REPORT 4

GROUP F

ALWIN KANNYAKONIL SCARIA

ANISHA SUSAN MATHEW

ASHNA VIJI ALEX

JOBIN PHILIP

MOHAMED AFTHAB

INTRODUCTION

This milestone report focuses on the evaluation and validation of machine learning models developed for the SMS Spam Classifier project. After successful model selection and training, we assessed the models' performance using various metrics to determine their effectiveness in classifying SMS messages as spam or non-spam.

PROGRESS REPORT

SUMMARY OF TASKS COMPLETED:

1. Model Selection

Selected machine learning models for evaluation: Logistic Regression, Stochastic Gradient Descent, Naive Bayes, Support Vector Machines (SVM), Random Forest, and Gradient Boosting Machines (GBM).

2. Model Training

Trained each model on a balanced dataset using the Synthetic Minority Over-Sampling Technique (SMOTE).

3. Model Evaluation and Validation

Evaluated models based on accuracy, precision, recall, F1-score, and confusion matrix.

Conducted cross-validation to ensure robustness of model performance.

KEY ACHEIVEMENTS AND MILESTONES

1. Evaluation and Validation of the Model:

    - Metrics Calculated: Accuracy, Precision, Recall, and F1-score. To provide a complete view of performance, these metrics were calculated for each model.
    - The area under the curve, or AUC, was calculated to assess how well the models could differentiate between emails that were spam and those that weren't.

2. Cross – validation:

- Ten-fold cross-validation was used to evaluate the models' generalizability and robustness.
- Ensured that each model performs uniformly across all data subsets.

3. Model Comparison:

    - The outcomes of Stochastic Gradient Descent, Random Forest, Naive Bayes, Support Vector Machines (SVM), Gradient Boosting, and Logistic Regression were compared and assessed.
    - Based on cross-validation results and assessment metrics, ascertained which model performed optimally.

CHALLENGES FACED

Complexity of Model Assessment:

Obstacle: A major challenge was comparing multiple models with different metrics and ensuring fair comparisons due to variations in data distribution and feature relevance. To address this, we sped up the evaluation process by putting in place automated scripts that computed several metrics, including accuracy, precision, recall, F1-score, and AUC-ROC. Furthermore, we used advanced visualization tools to generate comparative charts that made the data easier to understand. In addition, techniques like stratified sampling and feature importance analysis were employed to maintain consistency, ensuring impartial and precise model comparisons.

Resolution: To address this, we developed automated scripts that could efficiently oversee the assessment of several models, thereby optimizing the review process. These scripts were designed to calculate a wide range of metrics, such as accuracy, precision, recall, F1-score, and AUC-ROC, to guarantee a thorough assessment of each model's performance. Furthermore, we used advanced visualization tools to generate comparative plots and charts that made it easier to understand the data and identify the best performing models. This approach increased the accuracy and fairness of the comparisons while significantly reducing the time and effort required for model evaluation.

Implementing Real-Time Systems

Obstacle: Ensuring that the real-time system could handle high traffic levels and respond rapidly presented a significant difficulty during installation. This work was made more challenging by the need to manage latency at peak loads, maintain consistency and integrity of the data, cope with variations in user interactions and data input rates, and ensure fault tolerance and overall system stability.

Resolution: Numerous solutions were implemented to address these issues. Frontend and backend integration was optimized, and load testing was conducted to ensure scalability and efficiency. Asynchronous processing and caching strategies were employed to cut down on latency, and robust data validation and transaction management mechanisms were put in place to protect data integrity. To increase system resilience, redundancy and failover techniques were incorporated. Additionally, dynamic scaling capabilities were developed to adjust system resources in response to actual demand.

LESSONS LEARNED

- **Integration Challenges:** Effective real-time system development requires thorough planning and extensive testing to ensure smooth integration and scalability, highlighting the significance of carefully considering each system component.
- **Managing Data Variability:** It's imperative to consider variations in the distribution of data and the relevance of features across models. Techniques like stratified sampling and feature analysis help maintain consistency and accuracy in model evaluation.
- **Comprehensive evaluation:** It gives a more complete picture of a model's performance by highlighting both its strengths and faults that a single statistic could overlook.
- **Scalability Considerations:** Developing systems that can endure high traffic volumes requires more than just code optimization. Strategic planning is required to ensure consistent performance in the face of changing conditions. This covers both dynamic resource allocation and load balancing.
- **Robust recording:** A thorough documentation of strategies, challenges, and solutions is required. It helps the current project succeed while also serving as a helpful resource for stakeholders and upcoming initiatives.

TIMELINE TABLE

Task	Original Timeline	Revised Timeline	Status
Data Collection	Week 1	Week 1	Completed
Data Preprocessing	Week 2	Week 2	Completed
Feature Extraction	Week 3	Week 3	Completed
Model Selection	Week 4	Week 4	Completed
Model Training	Week 5	Week 5	Completed
Model Evaluation	Week 6	Week 6	Completed
Real-time System	Week 7	Week 7	Upcoming
Final Report and Demo	Week 8	Week 8	Upcoming

NEXT STEPS

1. Advanced Model Development:

Working on advanced model building using BERT (Bidirectional Encoder Representations from Transformers) which is a pre-trained NLP algorithm developed by google AI. It can have a deeper sense of language context and flow compared to the single-direction language models. BERT model instead of predicting the next word in a sequence makes use of a novel technique called Masked LM (MLM).

2. Model Deployment
3. Documentation and Reporting
4. Final Presentation and Submission

CONCLUSION

This achievement marks a significant advancement in our quest to detect email spam, as it includes the development of a real-time detection system and a successful model evaluation. To ensure a complete examination and identify the most successful model, we thoroughly evaluated multiple machine learning models using a variety of metrics and visualization tools.

We have extensively tested the scalability and backend and frontend efficiency of our real-time detection system. It is designed to react fast and endure high traffic. As we approach the finish line, our focus will be on finalizing our report, putting together a comprehensive presentation, and taking meticulous notes on everything. We are confident that our technology will increase email communication reliability by offering a robust and practical remedy for email spam identification.

REFERENCES

1. Almeida, T. A., Hidalgo, J. M. G., & Yamakami, A. (2011). Contributions to the Study of SMS Spam Filtering: New Collection and Results.
2. Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). SMOTE: Synthetic Minority Over-sampling Technique. Journal of Artificial Intelligence Research, 16, 321-357.