CS412 “Machine Learning” Project Report

Team Members:

Denizhan Altan 29158

Emre Çavuş 32326

İlhan İskurt 31112

Serhat Cemal Öztürk 29317

Introduction

This report outlines the methodology, implementation, and results of a machine learning project focused on predicting and classifying outcomes using Instagram post data. The project is structured into three main phases: training and prediction using a regression model, training a classification model, and utilizing the trained classification model for output prediction. Each phase is conducted using dedicated Python scripts. By applying feature engineering techniques and different machine learning algorithms, the project aims to identify significant patterns and enhance the accuracy and reliability of predictions and classifications.

Classification Approach and Analysis

The classification task in our project evolved through several phases, with each step building on lessons learned from earlier approaches. Initially, we adopted a baseline model provided by our instructor, which utilized the TF-IDF (Term Frequency-Inverse Document Frequency) method with a Naive Bayes classifier. The instructor's notebook removed emojis during preprocessing, but we chose to retain and extract them, hypothesizing that they might carry meaningful contextual information relevant to the classification task.

Improvements on the Baseline

To improve the baseline model's performance, we experimented with the maximum features parameter of the TF-IDF vectorizer, allowing us to focus on the most informative features in the dataset. Additionally, we tuned the alpha smoothing parameter of the Naive Bayes classifier, which helps control regularization and mitigates overfitting in sparse datasets.

From analyzing our approach, we identified that the gaming category, in particular, suffered from class imbalance. To address this issue, we experimented with methods such as Synthetic

Minority Oversampling Technique (SMOTE) and weighted class approaches. However, these methods did not yield significant improvements in local validation results. The highly specific nature of the gaming category data likely limited the effectiveness of these techniques. While this experimentation is not captured in the classification_training_model.ipynb or classification_prediction_model.ipynb, it formed a significant part of our earlier attempts and provided insights into the challenges of class imbalance.

Incorporating Neural Networks

Building on the baseline, we attempted to improve classification further by developing a custom neural network. This model focused on specific features such as category_enum and category_name, which appeared to provide clear and direct distinctions for the classification task. While the details of this neural network are not documented in the provided notebooks, its design was influenced by observations from earlier data exploration and feature analysis. We integrated this neural network into an ensemble model with the TF-IDF and Naive Bayes classifier, assigning a higher weight to the latter due to its consistent performance during local testing.

Round 1 Performance and Challenges

In local testing, the ensemble model achieved an accuracy of approximately 72%. However, when applied to the Round 1 test dataset, the performance plummeted to 28%. Upon reflection, we believe the primary reason for this catastrophic drop was our overconfidence in developing a custom neural network without sufficient expertise in deep learning techniques. While our intentions were to create a more sophisticated model, the lack of deeper understanding in this area led to suboptimal results. Additionally, the weighted ensemble approach may have amplified errors from the neural network, further degrading the overall accuracy.

During Round 2, efforts to analyze this failure included creating additional test datasets and experimenting with various preprocessing techniques. Despite these attempts, no significant improvement was achieved. Consequently, we refrained from submitting a classification model for Round 2.

Pivot to BERTurk

For Round 3, we shifted our focus entirely and adopted BERTurk, a Turkish pre-trained language model developed by researchers, including professors who had previously worked at our university. The classification_training_model.ipynb notebook captures the steps of preprocessing captions and biographies from Instagram profiles to create meaningful inputs for BERTurk. Fine-tuning was conducted over five epochs, adapting the model to the classification task. Special attention was given to preprocessing consistency, and insights from earlier experiments with TF-IDF and class balancing were integrated into this phase.

The analysis in the classification_prediction_model.ipynb notebook shows how the fine-tuned BERTurk model effectively captured contextual relationships within captions and biographies. This approach yielded significantly better results in categories that were previously challenging, such as gaming.

The transition to BERTurk marked a critical turning point in our strategy, highlighting the importance of leveraging advanced, pre-trained models rather than relying on custom-built neural networks. While earlier phases underscored the pitfalls of overconfidence and inexperience, they also provided valuable lessons that shaped our eventual success in the classification task. By integrating robust preprocessing pipelines, addressing class imbalances, and adopting state-of-the-art tools like BERTurk, we achieved a reliable submission for Round 3, demonstrating the value of a thoughtful and adaptable approach to machine learning challenges.

Regression Approach and Analysis

The regression task in our project was an iterative process, with each stage refining our understanding and methodology. This section details our journey from initial experiments to the final ensemble model, enriched by insights gained from extensive data preprocessing and analysis.

Initial Phase: Exploring Custom Metrics and Correlations

We began by creating several custom metrics to better understand the relationships within the dataset. The initial preprocessing steps included handling missing values, normalizing specific features, and performing exploratory data analysis (EDA) to evaluate correlations. By using visualizations such as scatter plots, heatmaps, and pair plots, we identified features that appeared to have strong relationships with the target variable. These visualizations and the correlation analysis can be observed in the regression_model.ipynb notebook.

Our initial feature set focused on metrics we believed to be directly relevant, aiming to reduce noise and improve model performance. This step provided valuable insights but also highlighted the limitations of our simplistic feature selection approach, as later iterations revealed the need for a more comprehensive feature set.

First Attempt: Handcrafted Neural Network

Based on our analysis, we developed a custom neural network model for the regression task. The network consisted of multiple dense layers with ReLU activation and dropout regularization to prevent overfitting. The model was trained using the Mean Absolute Error (MAE) as the loss function, as it directly aligned with our evaluation metric.

In our local evaluation set, the model achieved a MAE of approximately 2000–2500, which we considered satisfactory. However, when tested on the instructor-provided dataset, the performance dropped significantly, yielding a MAE of around 3800. This discrepancy, similar to what we observed in the classification task, highlighted the challenges posed by overconfidence in custom-built models and insufficient generalization.

Second Phase: Cross-Validation and Feature Refinement

In the second round, we retained the neural network structure but incorporated k-fold cross-validation to better assess the model's generalizability. Additionally, we focused on training the model using a subset of 5–6 features that we identified as the most correlated during the initial phase.

While cross-validation improved the model's robustness in local testing, the results on the instructor's dataset remained suboptimal. In retrospect, we realized that limiting the feature set to only a few variables constrained the model's ability to capture complex relationships within the data. This phase emphasized the importance of balancing feature selection with data richness for more comprehensive modeling.

Final Phase: Outlier Removal, Advanced Models, and Ensemble Approach

For the final round, we made significant changes to our approach. We began by identifying and removing outliers from the dataset, which had previously skewed the results. This was followed by more in-depth preprocessing, including feature scaling, normalization, and the engineering of additional features based on domain insights. These steps are meticulously documented in the regression_model.ipynb notebook.

We then moved away from relying solely on neural networks and experimented with advanced regression algorithms, including **Random Forest Regressor**, **LightGBM (LGBM)**, and other tree-based models. Each algorithm was tuned using hyperparameter optimization techniques such as grid search and random search to maximize performance. These experiments demonstrated the strengths and weaknesses of various algorithms, particularly their ability to handle non-linear relationships and feature interactions.

The final solution was an ensemble model that combined the predictions of multiple algorithms. This approach leveraged the strengths of each model, reducing the impact of individual weaknesses. To ensure sufficient training data while maintaining a robust evaluation set, we allocated 10% of the data as a test set. This allowed us to evaluate the ensemble's performance consistently while preserving enough data for training.

Results and Insights

The ensemble model outperformed all previous attempts, demonstrating improved generalizability and stability. By incorporating a diverse set of features, addressing outliers, and leveraging multiple regression techniques, we achieved a significant reduction in MAE compared to earlier phases. This success underscores the importance of iterative refinement, data preprocessing, and the use of ensemble methods in complex regression tasks.

Reflections

Throughout the regression task, our journey mirrored the challenges and lessons learned in the classification task. Early overconfidence in handcrafted neural networks and restrictive feature selection limited our initial results. However, through a data-driven approach, the adoption of advanced algorithms, and ensemble learning, we were able to overcome these challenges and deliver a reliable solution. This process highlights the value of adaptability, thorough analysis, and embracing state-of-the-art techniques in machine learning projects.