CS412 - Machine Learning Project Report

January 12, 2025

Abstract

This report provides a comprehensive analysis and implementation of machine learning techniques for regression and classification tasks. The project focuses on predicting 'like_counts' using LightGBM for regression and categorizing 'accountType' using a Voting Classifier. It details the preprocessing steps, feature engineering strategies, model configurations, and evaluation metrics used to achieve robust performance. The report also discusses challenges faced, strategies to mitigate them, and potential future improvements.

Contents

1 Introduction

Machine learning has become an integral part of data-driven decision-making processes. This project explores two critical machine learning problems:

- Predicting ‘like_count‘ for user-generated content using regression techniques.
- Categorizing ‘accountType‘ based on user profiles and post data through classification.

The primary goals of this project are to:

1. Develop robust preprocessing pipelines for regression and classification tasks.
2. Engineer meaningful features from raw data to improve predictive performance.
3. Implement state-of-the-art machine learning models and evaluate their effectiveness.
4. Analyze results and provide actionable insights for future enhancements.

2 Data Description

2.1 Overview of the Dataset

The datasets used in this project were obtained from user-generated content platforms. They consist of structured data for regression and classification tasks, including numerical, categorical, and text features.

2.2 Regression Task

The regression dataset includes features such as:

- **like_count:** The target variable, representing the number of likes a post received.
- **comments_count:** The number of comments on a post.
- **caption_length:** The length of the caption associated with a post.
- **media_type:** The type of media (e.g., photo, video) in the post.

2.3 Classification Task

The classification dataset consists of user profiles and post metadata. Key features include:

- **follower_count:** The number of followers a user has.
- **is_verified:** A binary indicator for verified accounts.
- **post_captions:** Text data from user-generated captions.
- **accountType:** The target variable, categorized into domains such as *fashion*, *sports*, and *travel*.

3 Methodology

3.1 Preprocessing

Preprocessing ensures the data is clean, consistent, and ready for model training. Each task required specific preprocessing techniques.

3.1.1 Regression Task

- **Handling Missing Values:** Null values in features like 'like_count' were replaced with global medians or means.
- **Feature Scaling:** Numerical features were standardized using `StandardScaler` to improve model convergence.
- **Encoding Categorical Data:** The 'media_type' feature was label-encoded to ensure compatibility with machine learning models.
- **Target Transformation:** Logarithmic transformation was applied to 'like_count' to address skewness.

3.1.2 Classification Task

- **Text Preprocessing:** Captions were cleaned by removing URLs, mentions, and stopwords, and lowercased for consistency.
- **Vectorization:** TF-IDF vectorization was applied to convert text data into numerical features.
- **Scaling:** Profile features like 'follower_count' and 'is_verified' were normalized to maintain uniformity.
- **Handling Imbalance:** SMOTE (Synthetic Minority Oversampling Technique) was used to address class imbalance.

3.2 Feature Engineering

Feature engineering enhances the predictive power of models by extracting meaningful insights from raw data.

- **Regression:** Features such as median, mean, and standard deviation of historical likes were computed to capture user-specific trends.
- **Classification:** Combined features from posts and profiles were engineered to enrich the representation of user behavior.

3.3 Models

3.3.1 Regression

LightGBM was chosen for its ability to handle large datasets and complex relationships efficiently. Key configurations include:

- Objective: ‘regression‘.
- Learning Rate: 0.05.
- Number of Estimators: 500.
- Regularization Parameters: `reg_alpha` and `reg_lambda` to prevent overfitting.

3.3.2 Classification

A Voting Classifier was implemented using the following base models:

- Logistic Regression for its simplicity and interpretability.
- Support Vector Classifier (SVC) with a linear kernel for robust boundary separation.
- XGBoost for its gradient boosting capabilities.

Soft voting was employed to combine predictions, leveraging the strengths of each model.

4 Results and Analysis

4.1 Regression Task

Table 1 presents the key metrics for the regression model. The LightGBM model demonstrated effective performance with minimal error.

Metric	Value
Log MSE	0.078
Min Predicted Likes	0
Max Predicted Likes	1,327,989.58
Zero Predictions	534

Table 1: Regression Performance Metrics

4.2 Classification Task

The classification results are summarized in Table 2. While the training accuracy was high, validation performance highlighted areas for improvement, particularly for imbalanced classes.

Category	Precision	Recall	F1-Score
Art	0.23	0.08	0.12
Fashion	0.53	0.72	0.61
Sports	0.80	0.70	0.74

Table 2: Classification Metrics

4.2.1 Insights

- The LightGBM model effectively captured trends in the regression task, but further tuning of hyperparameters could reduce error margins.
- The Voting Classifier ensemble demonstrated the utility of combining models, although class-specific performance varied significantly.

5 Discussion

5.1 Challenges Faced

- **Data Imbalance:** Addressing class imbalance in the classification task was challenging, necessitating the use of SMOTE.
- **Feature Engineering:** Designing features that generalize well across diverse user behaviors required extensive experimentation.

5.2 Future Work

- Incorporate advanced natural language processing techniques, such as embeddings, for better representation of text data.

- Explore deep learning models for both regression and classification tasks to improve performance further.
- Conduct extensive hyperparameter tuning and cross-validation to enhance generalizability.

6 Conclusion

This project highlights the application of machine learning models for regression and classification. The key takeaways include:

- LightGBM's effectiveness in handling skewed regression targets.
- The utility of ensemble methods like Voting Classifiers for imbalanced classification tasks.

Future efforts should focus on addressing class imbalance more effectively and exploring advanced modeling techniques.

7 References

1. McKinney, W. (2010). *Data Structures for Statistical Computing in Python.*
2. Hunter, J. D. (2007). *Matplotlib: A 2D Graphics Environment.*
3. Pedregosa, F., et al. (2011). *Scikit-learn: Machine Learning in Python.*