Ensemble Learning Strategies: Integration of Polynomial Kernel Logistic Regression, Kernelized k-Nearest Neighbors, and Deep Random Forests

Jen-Lung Hsu
Institute of Data Science, National
Cheng Kung University
Tainan, Taiwan
RE6121011@gs.ncku.edu.tw

***Abstract*—Ensemble learning techniques play a pivotal role in advancing predictive modeling within the domain of machine learning. This paper explores and compares various ensemble methodologies, namely polynomial kernel logistic regression, kernelized k-nearest neighbors (KNN), and the novel paradigm of Deep Random Forests. The research investigates the efficacy of combining outputs from these distinct models to enhance predictive performance. Experimental evaluations showcase the comparative strengths and weaknesses of these ensemble strategies across different datasets. The findings provide insights into the suitability and potential applications of these ensemble techniques, shedding light on their impact on predictive analytics within the machine learning landscape.**

Keywords—ensemble learning, machine learning, polynomial kernel, logistic regression, k-nearest neighbors, random forest

I. INTRODUCTION

Ensemble learning methods in machine learning have garnered significant attention due to their capacity to enhance predictive performance by combining diverse models. The amalgamation of various models' predictions often results in more robust and accurate outcomes than any individual model in isolation. This approach has proven its significance across numerous applications, contributing to the advancement of predictive analytics.

The primary aim of this research is to delve into and compare distinct ensemble methodologies within the machine learning domain. Specifically, this study focuses on exploring the efficacy and comparative performance of ensemble techniques involving polynomial kernel logistic regression, kernelized k-nearest neighbors (KNN), and the innovative concept of Deep Random Forests. By investigating and contrasting these diverse ensemble strategies, we seek to elucidate their individual strengths and weaknesses in enhancing predictive modeling capabilities.

II. PROBLEM DESCRIPTION

The dataset used for this study is derived from Kaggle's "Car Insurance Claim Prediction" dataset. It comprises comprehensive information about policyholders and the attributes associated with them, offering a rich source of data for exploring and solving real-world problems related to insurance claim prediction.

The dataset encompasses various features, including but not limited to:

- **Policy Tenure:** The duration for which a policyholder has been enrolled.
- **Car Age:** The age of the insured vehicle.
- **Car Owner's Age:** The age of the car owner.
- **Population Density:** The density of the population in the city where the policyholder resides.
- **Car Make and Model:** The specific make and model of the insured vehicle.
- **Power:** A measure of the vehicle's power.
- **Engine Type:** The type of engine in the insured car.
- And many more.

The dataset also includes a crucial target variable, which serves as the focal point of this study. This binary target variable indicates whether a policyholder is likely to file an insurance claim within the next 6 months or not. Predicting this claim occurrence accurately is of paramount importance to insurance companies, as it directly influences their risk assessment, pricing strategies, and operational decisions.

The problem at hand can be framed as a binary classification task, where the goal is to build machine learning models that can effectively predict whether a policyholder will file an insurance claim within the specified timeframe. This has practical implications for the insurance industry in terms of optimizing their claim management, fraud detection, and overall policyholder experience. To tackle this problem, we delve into the implementation and evaluation of supervised learning algorithms, with a specific focus on manual implementation, thereby gaining a deeper understanding of their functioning and performance. The choice of algorithms includes basic linear classifiers, k-nearest neighbors (K-NN) using various distance metrics, and decision trees, both with and without pruning. We also explore feature engineering techniques to enhance the predictive power of the models.

Furthermore, cross-validation is employed to assess the stability and generalization capability of these classifiers, addressing the need for robust and reliable insurance claim prediction models. By addressing these challenges in the context of the "Car Insurance Claim Prediction" dataset, this research aims to provide valuable insights for the insurance industry, ultimately contributing to improved risk assessment and claim management strategies.

III. DATA PREPROCESSING

In this section, we detail the preprocessing steps undertaken to prepare the dataset for our supervised learning classification tasks. The dataset[1] is retrieved from a real-world source, and we employ several operations to clean, enhance, and address class imbalance issues.

A. Handling Continuous Variables[2]

1) Removal of Uninformative ID Column: The first step in data preprocessing is to remove the **`policy_id`** column as it lacks any meaningful information for our prediction task.

*2) Processing **`max_torque`**:* The **`max_torque`** column contains information about torque values and the corresponding revolutions per minute (RPM). We extract the torque and RPM values and convert them to numeric data types. Subsequently, we calculate the 'torque to RPM ratio' for each entry to gain insights into the relationship between torque and RPM. The original **`max_torque`**, **`rpm`**, and **`torque`** columns are removed from the dataset.

*3) Processing **`max_power`**:* Similar to the **`max_torque`** column, the **`max_power`** column is processed to extract power and RPM values, which are then converted to numeric data types. We calculate the 'power to RPM ratio' for each entry. The **`power`**, **`rpm`**, and **`max_power`** columns are removed from the dataset.

B. Handling Categorical Variables[2]

1) Conversion of 'Yes' and 'No' to 1 and 0: In the dataset, categorical variables are represented as 'Yes' and 'No'. To make them suitable for machine learning algorithms, we transform these values into binary format: 'Yes' to 1 and 'No' to 0.

2) Dummy Encoding for Remaining Categorical Variables: To handle the remaining categorical variables, we employ dummy encoding. Notably, we use **`drop_first=True`** to avoid multicollinearity, ensuring one column is omitted to prevent redundancy.

C. Addressing Class Imbalance

The original dataset exhibits class imbalance with significantly more samples in the **`is_claim`** category labeled as '0' compared to '1'. To mitigate this issue, we employ Synthetic Minority Over-sampling Technique (SMOTE).

SMOTE is employed to oversample the minority class (**`is_claim`** = 1) by generating synthetic examples. The number of samples in the minority class is increased to match the number of samples in the majority class (**`is_claim`** = 0). The final preprocessed dataset contains a balanced distribution of classes, with an equal number of '0' and '1' labels for the **`is_claim`** target variable.

In summary, the preprocessing steps described in this section aim to enhance the dataset's suitability for training machine learning models, particularly in the context of classification tasks. The balanced dataset ensures that both classes are equally represented, allowing for more reliable model training and evaluation.

In the following sections, we will delve into the manual implementation of various classification algorithms, feature engineering, and model evaluation using cross-validation techniques.

IV. Approaches

A. Combining Outputs of Polynomial Kernel Logistic Regression and Kernelized KNN Models

The integration of outputs from Polynomial Kernel Logistic Regression and Kernelized KNN models within the ensemble framework involves a systematic approach. Firstly, the predictions generated by each model are obtained for a given dataset. These predictions, representing class labels, are then combined using ensemble strategies such as majority voting. This fusion of predictions aims to leverage the diverse strengths of individual models to achieve improved predictive performance.

B. Construction of Deep Random Forests

The construction of Deep Random Forests involves a novel adaptation of the traditional Random Forest framework. Instead of using decision trees as individual components, each tree is replaced by a 2-layer Multilayer Perceptron (MLP) neural network. These MLPs are trained on random subsets of features and instances, similar to the randomization principle in Random Forests. The training process involves optimizing the weights and biases of the MLPs using techniques like backpropagation.

Once the individual MLPs are trained, their outputs—representing predictions or probabilities—are combined within the ensemble structure. This combination can be achieved through various methods, such as simple averaging of the MLP outputs or employing more sophisticated techniques like weighted averaging, where the weights are determined based on the performance of each MLP on a validation set.

The overall objective of integrating these models—Polynomial Kernel Logistic Regression, Kernelized KNN, and Deep Random Forests—is to exploit their complementary strengths and diversity to enhance the predictive power of the ensemble system. The combination of these outputs aims to mitigate individual model weaknesses and capitalize on their collective strengths for more accurate and robust predictions.

V. Experiment and Results

A. Experimental Design

In this study, we devised an experimental framework that encompassed the creation of ensemble models incorporating Polynomial Kernel Logistic Regression, Kernelized KNN, and Deep Random Forests. The experimental process involved the partitioning of the dataset into training, validation, and testing sets for model training and evaluation.

B. Model Training and Evaluation Methodology

For the ensemble of Polynomial Kernel Logistic Regression and Kernelized KNN models, multiple models were constructed and subsequently combined into an ensemble. Training of these ensembles was carried out using the training dataset (**`X_train`**, **`y_train`**). The performance of the ensembles was assessed on the validation dataset (**`X_val`**, **`y_val`**), evaluating metrics such as accuracy.

For the Deep Random Forest model, an ensemble of 2-layer MLPs replacing traditional decision trees was employed. This ensemble was trained using the training dataset (**`X_tra`**, **`y_tra`**). Performance evaluation was conducted on the validation dataset (**`X_val`**, **`y_val`**), assessing accuracy as a key metric.

C. Experimental Results

The ensemble comprising Polynomial Kernel Logistic Regression and Kernelized KNN models exhibited promising performances, showcasing an accuracy of 0.81464 on the validation set and 0.81429 on the test set in Table I. These

results demonstrate the robustness and consistency of this ensemble approach in accurately predicting outcomes.

Conversely, the performance of the Deep Random Forest model, with accuracies of 0.49857 on the validation set and 0.50027 on the test set in Table II, presented notably lower accuracy metrics compared to the Polynomial Kernel Logistic Regression and Kernelized KNN ensemble. This discrepancy in performance could potentially be attributed to the nature of the dataset, which may not align optimally with the capabilities of the 2-layer MLPs integrated within the Deep Random Forest ensemble.

TABLE I. ACCURACY OF THE ENSEMBLE OF POLYNOMIAL KERNEL LOGISTIC REGRESSION AND KERNELIZED KNN MODELS IN THE VALIDATION SET AND TESTING SET RESPECTIVELY

The Ensemble of Polynomial Kernel Logistic Regression and Kernelized KNN models	
	Accuracy
Validation set	0.81464
Testing set	0.81429

TABLE II. ACCURACY OF THE DEEP RANDOM FOREST MODEL IN THE VALIDATION SET AND TESTING SET RESPECTIVELY

The Deep Random Forest model	
	Accuracy
Validation set	0.49858
Testing set	0.50027

D. Discussion and Potential Implications

The observed disparity in performance between the two ensemble strategies suggests a dependence on the dataset's characteristics and its compatibility with the models employed. The Deep Random Forest, incorporating 2-layer MLPs, might not be as effective on this specific dataset as anticipated. This outcome highlights the importance of understanding the inherent nature of the data and the suitability of model architectures when designing ensemble systems.

Further investigations into the dataset's properties and features may provide insights into why the Deep Random Forest model struggled to achieve higher accuracies. Exploring alternative model configurations or adjusting hyperparameters within the Deep Random Forest architecture could potentially enhance its performance and align it better with the dataset's characteristics.

This discrepancy underscores the need for meticulous consideration of model-data compatibility when constructing ensemble systems, as the effectiveness of ensemble techniques can vary based on the nature and complexity of the dataset. These observations prompt future avenues of research aimed at fine-tuning the Deep Random Forest model or exploring alternative architectures that might better align with the inherent characteristics of the dataset to achieve improved predictive performance.

VI. CONCLUSION

In summary, this research explored and compared ensemble methodologies comprising Polynomial Kernel Logistic Regression, Kernelized KNN, and Deep Random Forests. The experimental evaluation revealed key insights into their performance when applied to a specific dataset.

The ensemble strategy involving Polynomial Kernel Logistic Regression and Kernelized KNN models exhibited notable consistency and robustness, showcasing commendable accuracy levels of 0.81464 on the validation set and 0.81429 on the test set. These results underscore the efficacy and reliability of this ensemble approach in achieving accurate predictions.

Conversely, the Deep Random Forest model, incorporating 2-layer MLPs, yielded lower accuracies of 0.49857 on the validation set and 0.50027 on the test set. This discrepancy highlights the nuanced relationship between model architecture and dataset characteristics, emphasizing the importance of aligning model selection with the dataset's intrinsic properties for optimal performance.

The findings underscore the potential impact of ensemble strategies in enhancing predictive modeling capabilities. While the Deep Random Forest model's performance fell short in this specific context, the study emphasizes the need for continued exploration and refinement of ensemble methodologies tailored to specific datasets.

The contributions of this research lie in shedding light on the strengths and limitations of different ensemble strategies. The study emphasizes the potential of ensemble methodologies, especially Polynomial Kernel Logistic Regression and Kernelized KNN ensembles, in improving model performance across various applications in predictive analytics.

Moving forward, further investigations into model architecture adjustments, hyperparameter tuning, or alternative ensemble strategies could pave the way for enhanced performance and broader applicability of ensemble methods in diverse real-world scenarios. This research underscores the significance of thoughtful model selection and ensemble construction to harness the full potential of ensemble techniques in predictive modeling and machine learning applications.

REFERENCES

[1] Iftesha Najnin (2022). Kaggle Data Set:Car Insurance Claim Prediction. Car Insurance Claim Prediction (kaggle.com)

[2] Zubin Relia (2023). Car Insurance Claim Classifier. Car Insurance Claim Classifier | Kaggle

[3] ChatGPT