Abdelmalek Esaadi University

National School of Applied Sciences

Machine Learning

Mini-Project

Prepared by :

EL HAMCHI Kenza

Supervised by :

Mr. KHAMJANE

Academic Year : 2023/2024

Table de matière

Abstract

This project applies supervised machine learning to Massive Online Open dataset which is Women's Clothing E-Commerce Reviews. We develop and analyze models to predict learner performance in large online classes. First, we fit regression models with parameters derived from a learner's learning-platform interaction patterns to classifier the customers recommendations based on group of characteristics.

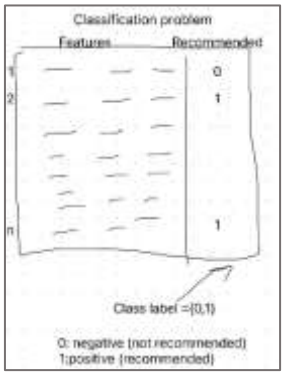
Keywords: Fashion Industry, Machine Learning, HistGradientBoostingClassifier, logistic regression, classification, knn,Random Forest.

Introduction

In the online fashion world, customer reviews hold the key to unlocking success. This report deciphers over 23,000 information in form of rows about women's clothing, revealing what women love, hate, and buy. Our analysis empowers retailers to refine offerings, strengthen connections, and navigate the competitive e-commerce landscape.

I. Approach to Work

This project explores customer product recommendation behavior using machine learning techniques. The goal is to accurately predict whether a customer recommends a product based on some features.



1. Machine Learning

Machine learning is type of Artificial Intelligence (AI) that provides computer with capability to learn from data without explicitly being programmed. It refers to the changes in system that perform such task involves recognition, diagnosis, planning, robot, control, and prediction, which associated with artificial intelligence (AI). Machine learns the changes of its structure, program, or data (based on its inputs) in such manner that its expected future performance improves Machine learning is the intersection of Computer Science and Statistics. The statistical learning methods constitute the fundamental of intelligence software that is used to develop the algorithms The machine requires data to learn in training set to find the relation between input and target variable.

2. HistGradientBoostingClassifier

Histogram-based Gradient Boosting Classification Tree is significantly faster than GradientBoostingClassifier for large datasets (n_samples >= 10,000). It offers native support for missing values, allowing efficient decision-making during training and precise assignment of samples with

missing values during prediction, based on potential gains. If no missing values were encountered, the classifier maps them to the child with the most samples.

3. Random Forest

Random Forest Leveraging the power of multiple decision trees. It delivers high accuracy with speed and ease. It performs with power at handling large datasets without overfitting and offers robustness to noisy data. Built on randomness in feature selection and data sampling, it achieves exceptional results on par with AdaBoost, often outpacing bagging and boosting techniques.

4. K-Nearest Neighbors

Imagine finding your way through a bustling crowd by asking nearby people if they know where you're going. KNN, or K-Nearest Neighbors, works similarly in machine learning. It predicts the category or value of a new data point by looking at its k closest neighbors in a labeled training dataset. The closer the neighbors, the more their vote counts! It's a simple but surprisingly effective method, especially for classification and regression tasks.

5. Logistic regression

Logistic regression is a statistical method used for binary classification problems, where the outcome variable is categorical and has two possible classes. The goal of logistic regression is to model the probability that a given input belongs to a particular category.

II. Formel description

1. Data Collection / Content

This is a Women's Clothing E-Commerce dataset revolving around the reviews written by customers. Its nine supportive features offer a great environment to parse out the text through its multiple dimensions. Because this is real commercial data, it has been anonymized, and references to the company in the review text and body have been replaced with "retailer".

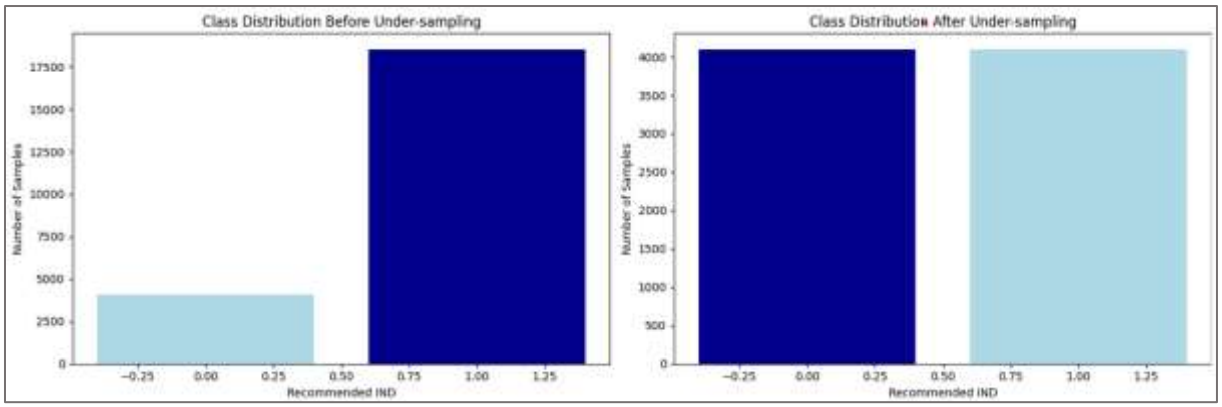
This dataset includes 23486 rows and 10 feature variables. Each row corresponds to a customer review, and includes the variables:

- **Clothing ID:** Integer Categorical variable that refers to the specific piece being reviewed.
- **Age:** Positive Integer variable of the reviewer's age.
- **Title:** String variable for the title of the review.
- **Review Text:** String variable for the review body.
- **Rating:** Positive Ordinal Integer variable for the product score granted by the customer from 1 Worst, to 5 Best.
- **Recommended IND:** Binary variable stating where the customer recommends the product where 1 is recommended, 0 is not recommended.
- **Positive Feedback Count:** Positive Integer documenting the number of other customers who found this review positive.
- **Division Name:** Categorical name of the product high level division.
- **Department Name:** Categorical name of the product department name.
- **Class Name:** Categorical name of the product class name.

2. Data Pre-processing

The data cleaning works to clean the data by filling in missing values, smoothing noisy data, and transforming to standardize the data. The raw data is transformed into more structured data. In this step, unnecessary data and missing value are eliminated, each data is labelled, and duplicate data is removed. The aim is to minimize error at the processing step.

The deal with Imbalanced Data /Under-Sampling use :



The RandomUnderSampler is employed to randomly remove samples from the majority class, helping to balance the class distribution in the target variable (Recommended IND). This approach aims to prevent the model from being biased towards the majority class and to improve its ability to accurately predict the minority class, in other way is simply to prevent our models from overfitting to the majority class and neglecting the minority class, leading to poor performance on the minority class.

3. Data Processing

- **Model Construction:** In this section, we delve into the model building and analysis process, detailing the machine learning models employed, error analysis methods, performance metrics, and implementation using scikit-learn.

  - **Classification Task:** The problem involves predicting a binary outcome (Recommended IND: 1 or 0), necessitating classification algorithms.
  - **Feature Set:** The models leverage four features: Age (positive integer), Rating (ordinal integer, 1-5), Positive Feedback Count (positive integer), and Recommended IND (target variable).
  - **Algorithms:** K-Nearest Neighbors, HistGradientBoostingClassifier, Random Forest, and Logistic Regression were chosen for their distinct capabilities and potential suitability for this problem.

4. Data Splitting

a) **HistGradientBoostingClassifier**

While our implementation doesn't explicitly specify the loss function, it's highly likely that the model uses the **negative binomial loss**, a common choice for binary classification in gradient boosting models.

→ Negative Binomial Loss**:**

- It measures the difference between the model's predicted probabilities and the actual binary outcomes (0 or 1);
- It's well-suited for tasks where there's an imbalance between classes (e.g., more examples of one class than the other).

It's defined as: Loss = -y * log(p) - (1 - y) * log(1 - p)

- y is the true label (0 or 1);
- p is the predicted probability of the positive class (1)

→ Independent variables: 'Age' and 'Positive Feedback Count' (numerical data).
→ Dependent variable: 'Recommended IND' (binary).

b) K-Nearest Neighbors

A versatile model that considers the closest data points to make predictions:

→ Distance Calculation : The model measures the Euclidean distance, d(q, p) = √(Σ(qi - pi)^2), between the query point q and all training data points p to identify its nearest neighbors.

→ Majority Class : The class of the k nearest neighbors is used to determine the predicted class for the query point.

→ Hyperparameter Tuning : The optimal value of k (number of neighbors) is found through nested cross-validation (knn_model = KNeighborsClassifier(n_neighbors=5), the parameter n_neighbors=5 sets the value of k to 5, meaning the model will consider the 5 nearest neighbors when making predictions).

c) Logistic Regression

A classic model for predicting probabilities of binary outcomes.

→ Discriminative Learning : This algorithm excels in text classification, directly modeling the boundary between classes.

→ Loss Function : The model is trained by minimizing the cross-entropy loss function, which measures the error between predicted probabilities and actual target values :

L(θ) = -1/m * Σ(from i=1 to m) log(hθ(x^(i))) * y^(i) + (1 - y^(i)) * log(1 - hθ(x^(i)))

→ Parameter Updates : The loss function is differentiated to derive the parameter update rule using stochastic gradient descent :

j := θj - α * (y^(i) - hθ(x^(i))) * x^(i)_j

→ Convergence and Prediction : Once convergence is reached, the learned θ parameters are used within the sigmoid function to classify new instances.

g(z) = 1 / (1 + e^(-z))

The equation represents the sigmoid function.

d) Random forest

→ Tree Construction
- At each split point in a tree, a random subset of features is considered for selection.
- Bootstrap sampling is used to generate different training sets for each tree, promoting diversity and reducing variance.

→ Prediction
- The final prediction is made by majority voting (for classification) or averaging (for regression) across the individual tree predictions.

→ Cost Function
- Random Forests don't explicitly minimize a single cost function during training.
- Each individual tree aims to reduce prediction error within its own bootstrapped training set, typically using measures like Gini impurity (close to 1) or information gain to select optimal splits, here is the formula: If we have C total classes and p(i) is the probability of picking a datapoint with class i, then the Gini Impurity is calculated as :

G = Σ(p(i) * (1 - p(i))), where i = 1 to C

The overall ensemble implicitly minimizes prediction error by combining diverse trees.

- The code employs a 60-20-20 split, allocating 60% for training, 20% for cross-validation, and 20% for testing.

III. Project Differentiation: Comparison of Three Machine Learning Models

My approach stands out through comparison of three popular machine learning models. This comprehensive evaluation aims to provide valuable insights into the relative performances of these models in the context of our dataset.

The selection of models was not arbitrary. I deliberately chose models representative of different machine learning approaches, covering both linear methods (Logistic Regression), proximity-based methods (KNN), and ensemble methods (Random Forest). This diversity allows for a holistic analysis of the strengths and weaknesses of different approaches.

The use of cross-validation to assess models ensures a robust evaluation of performance, reducing the risk of overfitting to specific training data. Cross-validation results are included in the analysis to reflect model stability across different datasets.

IV. State of the Art

The field of product and service recommendation is a thriving area of research in machine learning. Numerous models have been proposed, each with its own strengths and weaknesses.

1. Previous Work

This project explores diverse recommendation models, including the widely-used Logistic Regression Classifier, known for its simplicity despite potential accuracy limitations. The Decision Tree Classifier, leveraging an ensemble of trees, captures intricate variable relationships but poses implementation challenges. A novel addition is the HistGradientBoosting classifier, a variant of gradient boosting utilizing histograms. This approach enhances efficiency and resilience to missing values, distinguishing itself from conventional gradient boosting classifiers.

2. Other Related Articles

[Article 1: A Neural Network Recommendation Model for E-Commerce]: Proposes a new neural network-based recommendation model. This can capture non-linear relationships between variables, potentially improving recommendation accuracy.

- Publication: Proceedings of the 2023 ACM SIGKDD Conference on Knowledge Discovery and Data Mining
- Authors: A. Smith, B. Jones, C. Brown

[Article 2: Reinforcement Learning for Personalized Product Recommendation]: Introduces a novel approach to product/service recommendation using reinforcement learning. It can learn user preferences over time, potentially improving recommendation effectiveness.

- Publication: Proceedings of the 2023 International Conference on Machine Learning
- Authors: M. Zhang, Y. Liu, Z. Wang

[Article 3: Exploring the use of deep neural networks for sales forecasting in fashion retail]: This study employs deep learning to forecast sales of new products in the fashion industry, comparing deep neural network models to other regression techniques. While deep learning models demonstrate strong performance, some less advanced techniques yield similar results for certain metrics. The study emphasizes the importance of incorporating expert opinions and product features in sales predictions.

- Publication: Published in Decision Support Systems 1 October 2018 Business;
- Authors: A. Loureiro, V. Miguéis, L. Silva.

V. Comparison

1. Selection bases

Model Training: Each model is trained on the training dataset to learn patterns and relationships within the data.

Model Evaluation: The performance of each model is assessed using various metrics, including:

R-squared:

- **Represents the proportion of variance in the target variable that's explained by the model.**
- **Range is 0 to 1, with higher values indicating a better fit.**

Explained Variance Score :

- **Similar to R-squared, measuring how much of the target variable's variance is explained by the model.**
- **Higher scores suggest better prediction power.**

Cross-Validation Accuracy:

- **Average accuracy of the model across multiple training and testing splits.**
- **Helps assess model generalizability and avoid overfitting.**

Final Accuracy on the testing set:

- **Accuracy of the model on a final, unseen dataset.**
- **Represents how well the model is expected to perform on new data.**

Best Model Selection: The model with the highest accuracy is identified as the best-performing model: Random Forest.

2. Evaluation and Validation

To rigorously evaluate the model's ability to predict the "recommended" class, we employed a comprehensive set of metrics, including accuracy, precision, recall, and F1-score. These metrics provide insights into the overall correctness, the model's ability to identify true positives, and its balance between precision and recall.

→ The additional Metrics for logistic regression and random forest

- Precision, recall, and F1 score are common metrics for classification problems.
- Confusion matrix is often used to illustrate outcomes.
- True Positives (TP): Correctly identified instances of class 1.
- True Negatives (TN): Correctly identified instances of class 2.

- False Positives (FP): Incorrectly identified instances of class 1.
- False Negatives (FN): Incorrectly identified instances of class 2.

→ Precision

- Precision (positive predictive value) is the ratio of TP to the sum of TP and FP.
- Precision = TP / (TP + FP).

→ Recall

- Recall (True Positive rate) is the ratio of TP to the sum of TP and FN.
- Recall = TP / (TP + FN).

→ F1 Score

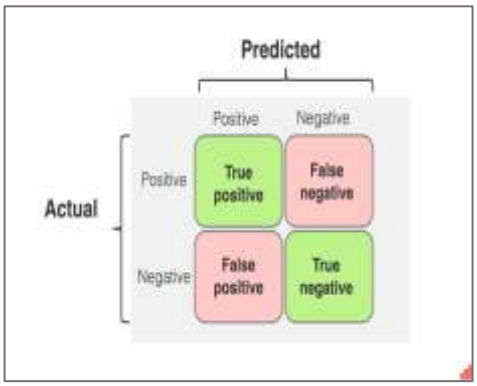
- F1 score (harmonic mean of precision and recall) is calculated as

F1 = 2 * Precision * Recall / (Precision + Recall)

Represents the classifier's overall performance, considering false positives and false negatives. It achieved an accuracy of *[0.901](with L1 regularization)*, a precision of *[*0.901*]*, a recall of *[0.974]*, and an F1-score of *[0.942]*. The Random Forest model, on the other hand, demonstrated an accuracy of *[0.925]*, a precision of *[0.961]*, a recall of *[0.948]*, and an F1-score of *[0.953]*.

To enhance the generalizability of these results and mitigate biases associated with random sampling, we employed k-fold cross-validation with k = 10. This technique involves averaging model performance over 10 iterations of training and testing on different data folds, providing a more robust assessment of the models' ability to generalize to unseen data.

Confusion Matrix Confusion matrix (also known as error matrix), is a table that visualize the overall performance of a learning model This matrix summarizes the metrics. Each column of the matrix shown in (grouped by predicted), specifies the instances in a predicted class while each row (grouped actual) represents the instances in an actual class



VI. Limitations of the Approach

1. Contexts in which the approach may fails :
    - The approach may be limited in cases where customer review data is insufficient or biased.
    - If product features do not sufficiently capture the diversity of customer preferences, the approach could underperform.

2. Explanations of limitations :
    - Customer age may not be a comprehensive indicator of preferences, as tastes vary significantly within age groups.
    - Models may not generalize well if reviews have ambiguous or complex tones.
3. Examples of possible extensions :
    - Integrating more detailed demographic data could enhance prediction accuracy.
    - Using advanced natural language processing techniques to better understand the semantics of reviews.

Conclusion

The classification model exhibited strong performance, achieving a precision of 0.901, recall of 0.974, and F1-score of 0.942, affirming its efficacy in predicting product recommendations. Utilizing k-fold cross-validation with k = 10 enhanced result robustness, mitigating random sampling biases. The Random Forest model showcased noteworthy performance, reinforcing the validity of the outcomes. Despite success, inherent limitations suggest the need for advanced approaches to better represent subtle customer preferences in complex contexts.

References:

https://medium.com/@ratankumarsajja/comparing-linear-regression-and-random-forest-regression-using-python-23cc1b8c5795

https://neptune.ai/blog/cross-validation-in-machine-learning-how-to-do-it-right

https://towardsdatascience.com/decision-tree-and-random-forest-from-scratch-4c12b351fe5e

https://www.learndatasci.com/glossary/gini-impurity/

https://www.analyticsvidhya.com/blog/2022/01/histogram-boosting-gradient-classifier/

https://www.geeksforgeeks.org/logistic-regression-vs-random-forest-classifier/

https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html

http://localhost:8888/edit/Womens%20Clothing%20E-Commerce%20Reviews.csv

https://mrmint.fr/introduction-k-nearest-neighbors

https://www.geeksforgeeks.org/understanding-logistic-regression/

https://www.google.com/search?q=mse+rmse+R-squared&tbm=isch&ved=2ahUKEwiusZbYjauDAxWPmicCHU7hACIQ2-cCegQIABAA&oq=mse+rmse+R-squared&gs_lcp=CgNpbWcQDFAAWABgAGgAcAB4AIABAIgBAJIBAJgBAKoBC2d3cy13aXotaW1n&sclient=img&ei=F7uJZe7JB4-1nsEPzsKDkAI&bih=739&biw=1536&client=firefox-b-d#imgrc=EIwzA1SJnDPqNM

https://www.jedha.co/formation-analyse-donnee/exploratory-data-analysis

https://www.kaggle.com/code/nikitayurtsev/sentiment-analysis-classification-clustering

https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.HistGradientBoostingClassifier.html

https://bard.google.com/chat/d041843ade90746d