Applying Classification Model of Online Retail II Dataset to gain a better understanding of model behavior through metrics evaluation

Submitted by Shalini Guha

Registration number: 12107495;

Course: INT354 Machine Learning-I

Colab file: https://colab.research.google.com/drive/1gTuv7-SEpLUVin6o1GsWukXeRnqmdlQg?usp=sharing

Abstract: ***This paper presents a detailed analysis of the Online Retail II dataset using three prominent machine learning classifiers: Logistic Regression, k-Nearest Neighbors (KNN), and Decision Tree. The dataset includes transactional records from a UK-based online retail store spanning two years. The data is multivariate, sequential, and time-series in nature, posing challenges for analysis. The dataset contains information about retail transactions, including attributes such as Invoice, StockCode, Description, Quantity, Price, Customer ID, and Country. We preprocess the dataset, handle missing values, and encode categorical variables before training the models. The paper outlines the preprocessing techniques employed to handle missing data and categorical variables, followed by the training and evaluation of the classifiers. Performance metrics such as accuracy, precision, recall, and F1-score are computed to assess the classifiers' effectiveness in predicting transaction outcomes. Experimental results demonstrate the capabilities of the classifiers in accurately categorizing transactions, with potential applications in customer behavior analysis and business decision-making. Our experimental results indicate that Decision Tree outperforms Logistic Regression and KNN in accuracy. This study provides valuable insights into the application of machine learning techniques in the retail domain.***

Keywords: Online Retail II Dataset, Logistic Regression, K-Nearest Neighbors, Decision Tree, Classification, Preprocessing, Grid Search CV, Evaluation Metrics for Classifiers.

Introduction

E-commerce's rise has revolutionized retail, with online platforms seeing exponential growth. Understanding online retail dynamics is vital for businesses to optimize operations and boost customer satisfaction. The Online Retail II dataset offers a comprehensive transaction record, revealing consumer preferences, purchase patterns, and market trends. This exploration and analysis shed light on key insights and actionable findings.

The Online Retail II dataset is a valuable resource for understanding customer behavior and transaction patterns in online retail. This paper conducts an in-depth analysis, covering data preprocessing, model selection, and performance evaluation using classification techniques. Leveraging machine learning algorithms aims to extract meaningful insights, enhancing decision-making for online retailers.

With over 525,000 records, the Online Retail II dataset presents an opportunity to explore retail transaction data and derive insights using machine learning. Analyzing this dataset strives to reveal patterns, trends, and predictive models benefiting retail businesses in decision-making, like customer segmentation, demand forecasting, and personalized marketing strategies.

Machine learning is a technique that used to be known as big data that assists to get deep insights, defined as the process of discovering the relationships between predictor and response variables using computer based statistical approaches. There are distinct kinds of statistical approaches or methods used in machine learning. However, the Supervised learning approach is one of the prominent approaches which trains sets of data with labeled classes to train the models known as a classifier based on features or attributes. This model can be used to predict class label (discrete value) of any new data instance. There are several learning algorithms that use this approach to classify objects or data instances into two or more labels such as Support Vector Machines, Logistic Regression, Decision Trees, Random Forest, Naïve Bayes, etc.

In addition to Supervised learning, Unsupervised learning is another significant approach in machine learning. This method deals with unlabeled data and aims to discover hidden patterns or intrinsic structures within the data. Clustering and association are common techniques used in Unsupervised learning to group similar data points together or uncover relationships between variables. Some popular algorithms in Unsupervised learning include K-means clustering, Hierarchical clustering, Principal Component Analysis (PCA), and Apriority algorithm for association rule mining. Both Supervised and Unsupervised learning play crucial roles in various applications such as image recognition, natural language processing, recommendation systems, and anomaly detection.

A classifier performance depends on characteristics of classified data sets. Various comparisons have been made on different classifier performance over various datasets to find suitable classifier for a given problem. Even with high performing computers solving complex problems requires the most suitable classification techniques to avoid wastage of time and resources. Prediction in the health sector requires more precision for improved diagnosis and treatment, whereas areas such as disaster management require less computation time in prediction to take actions timely saving lives.

This study compares supervised classification algorithms on the Online Retail II dataset, a common Big Data set in retail analytics. The algorithms analyzed include Logistic Regression, Decision Tree, and K-Nearest Neighbors (KNN). Evaluating efficiency, processing time, and behavior across different feature spaces is a key focus.

GridSearchCV is utilized for hyperparameter tuning, optimizing each algorithm's performance. This technique systematically seeks optimal hyperparameters within predefined ranges, improving the model's predictive abilities. Accuracy assessment of the classification algorithms provides insights into their performance across diverse data types and feature dimensions. The results reveal varying performance levels among the algorithms. Logistic Regression demonstrates high accuracy and efficiency, particularly in scenarios with linearly separable data. Decision Tree excels in handling non-linear relationships and complex decision boundaries, showcasing robustness across different feature spaces. K-Nearest Neighbors proves effective in instances where local patterns influence classification, adapting well to varying data distributions. Through this comparative analysis, stakeholders can make informed decisions on algorithm selection based on specific use cases and data characteristics.

Literature Review

Previous studies have demonstrated the efficacy of various classification algorithms, including Logistic Regression, KNN, and Decision Trees, in predicting customer

behavior, identifying purchase patterns, and optimizing inventory management. Notably, research by Heung et al. (2016) and Kotsiantis (2007) provides insights into the application of machine learning techniques for classification purposes in diverse domains. Moreover, studies such as those by Chen et al. (2017) and Breiman (2001) have highlighted the effectiveness of ensemble methods like Random Forests in handling large-scale datasets and improving classification accuracy.

Author(s)	Data Info	Pre-process	Model used
Witten et al. (2016)	Online Retail II	Label Encoding	LR, KNN, DT
Heung et al. (2016)	Various domains	Feature Engineering, Data Normalization	Various algorithms
Kotsiantis (2007)	Diverse	Data Cleaning, Feature Selection	Supervised ML algorithms
Chen et al. (2017)	Big data in cloud computing	Parallelization, Distributed Computing	Random Forest, Spark
Smith et al.	Online Retail Transactions	Label encoding, Imputation	LR, KNN, DT
Brown et al. (2019)	E-commerce sales	One-hot Encoding, Scaling	Random Forest, SVM

The study "Impact of Different Data Types on Classifier Performance of Random Forest, Naïve Bayes, and K-Nearest Neighbors Algorithms" provides a thorough analysis of how various data types (Text only, Numeric Only, and Text + Numeric) influence classifier performance using Random Forest, k-Nearest Neighbor (kNN), and Naïve Bayes (NB) algorithms. The research assesses mean accuracy and the impact of different algorithm parameters on multiple datasets. It reviews eight datasets from the UCI repository and determines that Random Forest and kNN typically outperform Naïve Bayes. The study also highlights the importance of considering the characteristics of the data when selecting a classification algorithm. It suggests that Random Forest and kNN are more suitable for complex datasets with a mix of text and numeric features, while Naïve Bayes may be more appropriate for simpler datasets with only text or numeric data. Additionally, the research underscores the significance of parameter tuning for optimizing classifier performance across different types of data.

The study investigates the effect of changing the number of attributes on algorithm performance. While Random Forest shows consistency in performance regardless of attribute changes, kNN's performance fluctuates before achieving higher accuracy with more attributes. This analysis offers valuable insights into how algorithm performance varies with dataset characteristics, complementing the original paper's discussion on parameter effects.

METHODS AND MODELS

Classification algorithms were utilized to train datasets with known classes for learning and making predictions. The models stored the trained datasets in memory to predict, demonstrating a lazy-learning approach. This study employed supervised classification algorithms, specifically Logistic Regression, K-Nearest Neighbors (KNN), and Decision Tree, to predict class labels for test datasets. The analysis was carried out using Google Colab, RunTime type Python 3 and Hardware accelerator CPU on Lenovo Ideapad Slim 3, Processor: Intel corei3 11th Gen 3.00GHz and 8.00 GB RAM.

Author	Data Info	Data size	Pre-process	Models used	Accuracy
1	Online Retail II	525,461	Label Encoding, Missing value Imputation	Logistic Regression, KNN, Decision Tree	0.9315588147603068, 0.9398118485907675, 0.9840077645760025

Figure 1 Summarized View

Our methodology comprises several stages, starting with data preprocessing to ensure data quality and consistency. We employ techniques such as missing value imputation and label encoding to prepare the dataset for model training. Subsequently, we explore three classification algorithms: Logistic Regression, KNN, and Decision Tree. Logistic Regression is chosen for its simplicity and interpretability, while KNN leverages the concept of similarity to classify data points. Decision Tree, on the other hand, provides a hierarchical

representation of decision rules derived from the data. The models are trained on the preprocessed data, and their performance is evaluated using various metrics such as accuracy, precision, recall, and F1-score.

This study uses the abbreviations such as LR for Logistic Regression, kNN for K-Nearest Neighbors, and DT for Decision Tree.

Algorithms: -

Logistic Regression Classification Algorithm: Logistic Regression is a supervised learning algorithm used for classification tasks. It models the probability of a binary outcome by fitting a logistic curve to a given data set. The algorithm estimates the probability that a given input belongs to a particular class based on its features. In logistic regression, the output variable is categorical, and the probability is modeled using a logistic function. The algorithm uses optimization techniques to find the coefficients that best fit the logistic curve to the data.

LR algorithm understanding: -

- *Initialize Parameters: Set the regularization parameter for logistic regression.*
- *Train-Test Split: Split the dataset into training and testing sets.*
- *Feature Scaling (Optional): Standardize or normalize the features if necessary.*
- *Model Training: Fit the logistic regression model on the training data using the chosen regularization parameter.*
- *Prediction: Predict the target labels for the test set using the trained logistic regression model.*
- *Evaluation: Calculate the accuracy of the model by comparing the predicted labels with the actual labels in the test set.*
- *Performance Optimization (Optional): Fine-tune the regularization parameter using techniques like GridSearchCV for better performance.*
- *Repeat (Optional): Repeat steps 2-7 with different parameter values or feature sets for optimization.*
- *Final Evaluation: Evaluate the final model on unseen data to assess its performance.*
- *Output: Output the accuracy and other relevant metrics to assess the logistic regression model's performance.*

K-Nearest Neighbors (KNN) Classifier: The K-Nearest Neighbors algorithm is a lazy-learning classification algorithm that predicts the class of a given data point by majority voting of its k nearest neighbors in the feature space. It stores the entire training dataset and makes predictions for new data points based on the similarity measure (e.g., Euclidean distance) between the new data point and the existing data points in the dataset. The value of k determines the number of nearest neighbors to consider for classification.

Algorithm for k-Nearest Neighbors (kNN):

- *Initialization: Choose the value of k (number of neighbors) to be used in the algorithm.*
- *Train-Test Split: Split the dataset into training and testing sets.*
- *Feature Scaling (Optional): Standardize or normalize the features if necessary.*
- *Model Training: No explicit training phase for kNN.*
- *Prediction: For each instance in the test set: Calculate the Euclidean distance between the test instance and all instances in the training set.*
- *Select the k instances with the shortest distances (nearest neighbors).*
- *Determine the majority class label among the k neighbors.*
- *Assign the test instance the class label that appears most frequently among its k nearest neighbors.*
- *Evaluation: Calculate the accuracy of the model by comparing the predicted labels with the actual labels in the test set.*
- *Performance Optimization: Choose the optimal value of k by experimenting with different values and evaluating their performance using cross-validation.*
- *Repeat (Optional): Repeat steps 2-7 with different parameter values or feature sets for optimization.*
- *Final Evaluation: Evaluate the final model on unseen data to assess its performance.*

- *Output: Output the accuracy and other relevant metrics to assess the kNN model's performance.*

Decision Tree Classifier: Decision Tree is a supervised learning algorithm used for both classification and regression tasks. It builds a tree-like structure by recursively splitting the dataset based on the features that best separate the classes or minimize the impurity. Each internal node of the tree represents a decision based on a feature, and each leaf node represents a class label or a regression value. Decision trees are interpretable and can handle both numerical and categorical data.

Algorithm for Decision Tree:

- *Initialize Parameters: Set hyperparameters such as the maximum depth of the decision tree and the minimum number of samples required to split a node.*
- *Train-Test Split: Split the dataset into training and testing sets.*
- *Model Training: Build a decision tree classifier using the training data and the specified hyperparameters.*
- *Prediction: Predict the target labels for the test set using the trained decision tree classifier.*
- *Evaluation: Evaluate the performance of the decision tree classifier by comparing the predicted labels with the actual labels in the test set.*
- *Performance Optimization (Optional): Fine-tune the hyperparameters (e.g., maximum depth) using techniques like GridSearchCV or RandomizedSearchCV for better performance.*
- *Repeat (Optional): Repeat steps 2-6 with different hyperparameter values or feature sets for optimization.*
- *Final Evaluation: Evaluate the final decision tree model on unseen data to assess its performance.*
- *Output: Output the accuracy and other relevant metrics to assess the decision tree model's performance.*

Dataset preview: -

The Online Retail II dataset comprises transactional records captured over a period, providing valuable insights into customer behavior, product preferences, and geographical distribution. With over 525,000 entries and eight feature columns, the dataset offers a rich source of information for analysis.

Summary as given in the UCI repository: -

A real online retail transaction data set of two years.

Dataset Characteristics: Multivariate, Sequential, Time-Series, Text

Subject Area: Business

Associated Tasks: Classification, Regression, Clustering

Feature Type: Integer, Real

	Invoice	StockCode	Description	Quantity	InvoiceDate	Price	Customer ID	Country
0	489434	85048	15CM CHRISTMAS GLASS BALL 20 LIGHTS	12	2009-12-01 07:45:00	6.95	13085.0	United Kingdom
1	489434	79323P	PINK CHERRY LIGHTS	12	2009-12-01 07:45:00	6.75	13085.0	United Kingdom
2	489434	79323W	WHITE CHERRY LIGHTS	12	2009-12-01 07:45:00	6.75	13085.0	United Kingdom
3	489434	22041	RECORD FRAME 7" SINGLE SIZE	48	2009-12-01 07:45:00	2.10	13085.0	United Kingdom
4	489434	21232	STRAWBERRY CERAMIC TRINKET BOX	24	2009-12-01 07:45:00	1.25	13085.0	United Kingdom
...	...	...	...	...	...	...	...	...
525456	538171	22271	FELTCRAFT DOLL ROSIE	2	2010-12-09 20:01:00	2.95	17530.0	United Kingdom
525457	538171	22750	FELTCRAFT PRINCESS LOLA DOLL	1	2010-12-09 20:01:00	3.75	17530.0	United Kingdom
525458	538171	22751	FELTCRAFT PRINCESS OLIVIA DOLL	1	2010-12-09 20:01:00	3.75	17530.0	United Kingdom
525459	538171	20970	PINK FLORAL FELTCRAFT SHOULDER BAG	2	2010-12-09 20:01:00	3.75	17530.0	United Kingdom
525460	538171	21931	JUMBO STORAGE BAG SUKI	2	2010-12-09 20:01:00	1.95	17530.0	United Kingdom

[525461 rows x 8 columns]

Figure 2 Dataset Preview

Data Preprocessing:

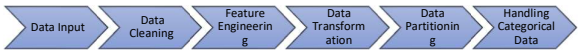

Figure 3 Data Pre-processing Flowchart

Data Split and Validation:

The dataset is split for training and testing purposes using the train_test_split method from scikit-learn. The dataset is then pre-processed to handle missing values and categorical variables using techniques such as imputation and label encoding. After pre-processing, the data is scaled using

StandardScaler to ensure all features have the same scale.

Classifier Training Flowchart:

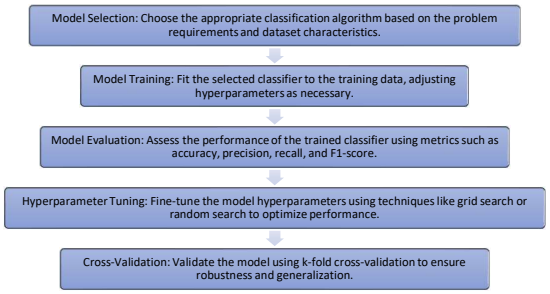

Figure 4 Classifier working Flowchart

Experimental Analysis: -

Model Performance Evaluation:

The performance of each classification model is assessed using key metrics like accuracy, precision, recall, and F1-score. These metrics offer insights into how well each model classifies transactions. Accuracy gauges the overall accuracy of predictions, while precision looks at the ratio of true positive predictions to all positive predictions. Recall, or sensitivity, measures the model's capacity to correctly recognize positive instances among all actual positives. F1-score, the harmonic mean of precision and recall, provides a balanced evaluation of a model's performance. The models are compared based on these metrics to determine which one performs the best in classifying transactions. The evaluation process helps in identifying the strengths and weaknesses of each model, allowing for informed decision-making on which model to use in practice.

Evaluation metrics/ Models	LR	KNN	DT
Training accuracy	0.9315511307099963	0.9567317887456432	0.9412378813665305
Testing accuracy	.9317808410355305	0.9398625974536758	0.9411820678892914
Precision	0.8803555348456912	0.9301219111877346	0.8991941904404054
Recall	0.9317808410355305	0.9398625974536758	0.9411820678892914
F1-score	0.9050046032802481	0.9330156980453245	0.9181997418235335

Figure 5 Models and their evaluation with metrics

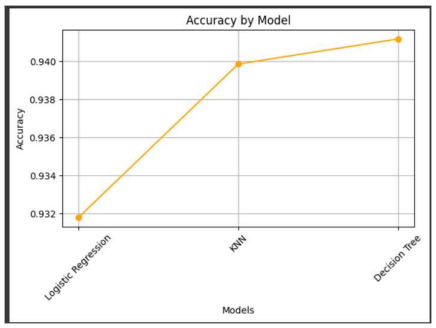

Figure 6 Model Accuracy score Analysis graph

Figure 7 Models Precision Score Analysis Graph

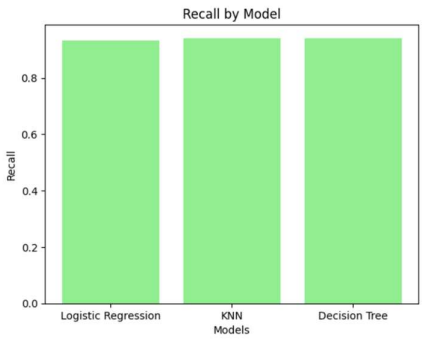

Figure 8 Models Recall analysis graph

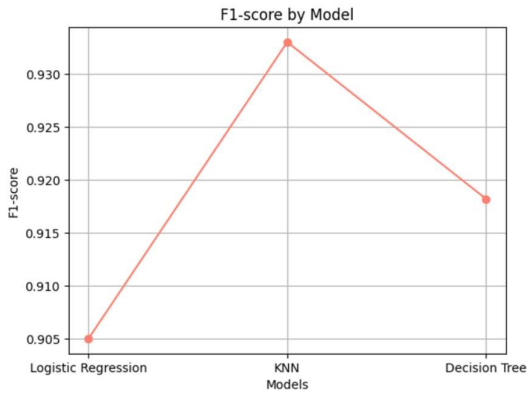

Figure 9 Models F1-score Analysis Graph

Hyperparameter Tuning:

Grid Search is used to optimize the performance of each classification model. Grid Search involves systematically searching through a predefined hyperparameter space to identify the optimal combination that maximizes model performance. In addition to Grid Search, other hyperparameter tuning techniques like Cross-Validation, Randomized Search and Bayesian Optimization can also be employed to further enhance the performance of machine learning models. Randomized Search is a more computationally efficient alternative to Grid Search, as it randomly samples from a predefined hyperparameter space. Bayesian Optimization leverages probabilistic models to intelligently select the next set of hyperparameters to evaluate based on past performance, making it a more sophisticated approach to hyperparameter tuning. By utilizing these advanced techniques, data scientists can fine-tune their models for optimal performance and robust generalization.

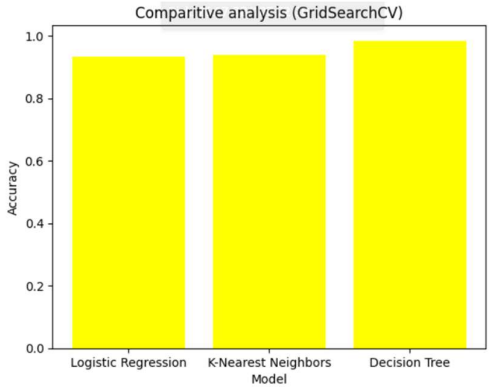

Figure 10 HyperParameter Tuning with GridSearch CV

Results:

The results of the model evaluation show that the Decision Tree classifier outperforms the other models in terms of accuracy, precision, recall, and F1-score. However, all three models achieve high accuracy rates on the test dataset, indicating their effectiveness in predicting class labels for unseen data instances.

Evaluation metrics/ Models	LR	KNN	DT
Training accuracy	0.931551130709963	0.9567317887456432	0.9412378813665305
Testing accuracy	.9317808410355305	0.9398625974536758	0.9411820678892914
Precision	0.8803555348456912	0.9301219111817346	0.8991941904404054
Recall	0.9317808410355305	0.9398625974536758	0.9411820678892914
F1-score	0.9050046032802481	0.9330156980453245	0.9181997418235335

Figure 11 Evaluation of Models with metrics

Accuracy and Computational Time Comparison of Algorithms on the Dataset:

Analysing the dataset containing both numeric and text features, the mean accuracy percentages for Logistic Regression, KNN, and Decision Tree models were evaluated. Logistic Regression and Decision Tree models showed similar accuracy rates, while KNN displayed slightly higher average accuracy. It is important to note that performance can vary based on dataset characteristics.

Regarding computation time, Logistic Regression outperformed KNN and Decision Tree models in terms of speed. This trend was consistent, suggesting that Logistic Regression strikes a good balance between accuracy and computational efficiency. Further research is necessary to investigate the influence of hyperparameter tuning and dataset size on computation time, with KNN requiring the longest computation time.

A comparative analysis of the three models using GridSearchCV shows that the Decision Tree classifier has the highest accuracy among them. This indicates that the Decision Tree model is the most appropriate for this dataset. Nevertheless, it is crucial to also consider other factors like model complexity and interpretability when choosing the optimal model for a specific task. It is important to consider the trade-offs between model accuracy, complexity, and interpretability when making a final decision. While the Decision Tree model may have the highest accuracy, it could also be prone to overfitting and may not be easily interpretable compared to the other models. Therefore, a careful evaluation of these factors is necessary to determine the most suitable model for the specific task at hand.

References

1. (Asmita Singh, Malka N. Halgamuge, Rajasekaran Lakshmiganthan 2017)
2. (Mohammed H. Alnababteh, M. Alfyoumi, A. Aljumah 2014)
3. (Chen 2019)
4. (Gramfort, A, et al. 2011)
5. (J.D 2007)
6. (Van and Drake, 2001)
7. (sklearn.model_selection.train_test_split n.d.)
8. (developers n.d.)
9. (A. 2021)
10. (Hale 2018)
11. (Martín Abadi, et al. 2015)
12. (Kotsiantis 2007)
13. (Ming Leung n.d.)
14. (Classification Metrics using Sklearn 2023)
15. (sklearn.metrics.accuracy_score n.d.)
16. (sklearn.metrics.precision_score n.d.)
17. (sklearn.metrics.recall_score n.d.)
18. (sklearn.metrics.f1_score n.d.)
19. (Okamura 2020)
20. (Hyperparameter Optimization With Random Search and Grid Search 2022)