COMP-8740-Project1. Date of submission October 20, 2021, date of current version October 19, 2021.

Performance Assessment of Classification Methods for In-Vehicle Coupon Recommendation Systems (October 2021)

R.Khalil, MSc Student: (104991048), L.Khalil, MSc Student:(105144173)
University of Windsor, Windsor, Ontario, Canada
Corresponding authors: First R.Khalil (e-mail: khalilr@uwindsor.ca). Second L.Khalil (e-mail: khali121@uwindsor.ca)

ABSTRACT Recommender systems are one of the most popular and successful applications of modern data machine learning technologies. In particular, coupon recommendation systems, such as Groupon and Honey, can enhance customers' engagement with businesses and increase the likelihood of the customer's return. The customer is often looking for convenience and ease of use of any system. Because of that, the focus of this project is in-vehicle coupon recommendation systems. In-vehicle recommendation systems aim to conveniently supply the right coupons based on preferences of the user and by taking available context information into account. This information consists of the user's current driving situation, such as the destination, current time, weather, passengers, etc. However, the challenge is getting to know the customer and deciding which coupons to recommend. This task can be complex because a single brand can have thousands of customers, each of whom responds to coupon offers differently depending on the categorical attributes presented in their customer profile. To solve this complex task, we will use various machine learning classification techniques to predict a user's response to a coupon recommendation based on their user profile or similarities to other user's profile using an in-vehicle coupon recommendation system dataset that consists of 12684 instances. This dataset was downloaded from the UCI Machine Learning Repository (Bache and Lichman, 2013). Feature selection methods and supervised learning classifier models with different hyperparameters were used to train the data. The best accuracy achieved among five different classifier models was a 73 percent, while the best precision was a 72 percent. These results suggest that we have "High Bias" in our models. In other words, our model is underfitting our example dataset and it is not presenting an accurate representation of the relationship between input and predicted output. For better accuracy and precision, a boosting algorithm called CatBoost was used for gradient boosting on an ensemble of decision tree classifiers to overcome the underfitting problem. The accuracy result was 95.7 percent, while the precision was 94 percent, with 96 percent recall and 99 percent Area Under the ROC Curve (AUC).

KEY WORDS Coupon purchase, Sales recommendation System, Machine Learning, Categorical data mining, Gradient boosting technique, classification.

1. INTRODUCTION

Coupon recommendation systems work on the basis of locating and recommending coupons that are most likely to be purchased by a user. The marketing of systems like the in-vehicle coupon mobile recommender can benefit from a robust recommendation technique. Machine learning encompasses a diverse set of algorithms that learn from data in order to make more accurate predictions. Although, the number of machine learning prediction algorithms is continually increasing, most machine learning tools perform effectively with numerical single-sourced datasets. They don't function as well with categorical non-numerical data that includes multi-sourced variables. Occupation, marital status, and frequency of restaurant visits are examples of such features from our dataset. These categories will have to be transformed to numbers to use typical machine learning algorithms on this type of data, which can be a nontrivial task that may not result in accurate predictions. In this study we applied various machine learning classification algorithms on the in-vehicle Coupon Recommendation dataset, which is available online at the UCI Machine Learning Repository (Bache and Lichman, 2013). We began our performance comparison analysis by examining the dataset and transforming the categorical variables into numerical forms according to their types. The prediction problem presented in this dataset is of the following form:

if a customer (goes to coffee houses ≥ once per month ∧ destination = no urgent place ∧ passenger ≠ kids) ∨ (goes to coffee houses ≥ once per month ∧ the time until coupon expires = one day) ⇒ predict Y = 1

This means that the customer will accept the coupon for a coffee house (Wang et. al. 2017). Keeping this in mind, we used content-based classifiers like the K Nearest Neighbor algorithm. This algorithm makes predictions based on similarities between the user's features and previous coupon buying decisions. Next, we analyzed the results of a classification-based method like Logistic Regression, which predicts the value of unseen data based on its association with predictor variables. Additionally, we used several-input variables classifiers such as

Decision Tree and Random Forest. After numerous trials with hyper-parameter tuning, Random Forest has achieved the best accuracy of all classifiers at roughly 73 percent. As a result, we adopted the CatBoost classifier from the Pycaret library, which is based on the gradient boosting decision tree algorithm (Dorogush et. al. 2018). Our model achieved a 95.7 percent accuracy with the CatBoost classifier.

2. LITERATURE REVIEW

We started by reviewing a paper written by Tong Wang et. al. and published in 2017 (Wang et. al. 2017). This paper discussed the interpretability issue of recommendation systems that utilize traditional machine learning algorithms. The focus of this study was on their proposed Bayesian Rule Sets (BRS) to predict user's coupon purchase behavior while driving their vehicle. The model is provided with a customized set of rules that are conjunction of conditions. Based on these rules the model predicts and provides characteristics of the users and the scenarios that lead to accepting a coupon. Their approach involved comparing the performance of their model against that of a number of classifiers when applied to the in-vehicle coupon recommendation dataset, which is available at the UCI Machine Learning Repository (Bache and Lichman, 2013). We discovered that when we used a variety of classifiers on the same dataset, the results were comparable to theirs. Their BRS technique had a 2 percent higher AUC than their decision tree classifier, with BRS scoring 77 percent and the decision tree scoring 75 percent. Consequently, we reviewed a paper written by Dorogush et. al. and published in 2018 (Dorogush et. al. 2018). According to this paper, the gradient boosting decision tree achieves state-of-the-art results in recommendation systems. This algorithm's powerful prediction technique is based on iteratively combining the results of weaker models, such as multi decision tress, to leverage their performance. This greedy procedure corresponds to gradient descent in a function. It is worth noting that using the CatBoost classifier from the Pycaret library on the in-vehicle coupon recommendation dataset yielded the greatest results, with a 99 percent AUC score.

3. PROBLEM STATEMENT

With the massive number of new recommendation systems based on multi-sourced categorical data, researchers must determine the most efficient way to provide users with the most accurate predictions in order to maximize their benefits. The issue is that most machine learning algorithms only work with numerical single-source datasets as opposed to categorical non-numerical data that has multiple sources of variables. Dealing with categorical data necessitates encoding, which may be a tedious process that can have a significant impact on prediction results if done incorrectly. Motivated by our literature review, we conducted this comparative study, finding that the gradient boosting decision tree method achieved the best prediction on the in-vehicle coupon purchase multi-sourced categorical dataset with a 99 percent ACU score. To anticipate a target label, these algorithms employ an ensemble of decision trees. Avoid underfitting, high bias, and improving accuracy is accomplished by attempting to minimize the loss function by merging the findings at each level of the decision tree. This suggests that the use of gradient boosting decision tree method can be a possible solution for this issue.

4. GENERAL CONCEPTS

4.1 Dataset

The in-vehicle coupon recommendation Data Set was obtained from the UCI Machine Learning Repository (Bache and Lichman, 2013). It contains records of 12684 users and 26 attributes. This non-linear dataset studies whether a person will accept a coupon recommended to them based on different driving scenarios from which the user profiles are built. The driving scenarios can include the destination, current time, weather, passenger, etc. The data was collected via a survey on Amazon Mechanical Turk. (Bache and Lichman, 2013).

4.2 Classification

Classification is a predictive modelling problem in machine learning, where a class label is predicted for a certain input (Bishop, 2007). In classification problems, the model approximates a function that maps the input variable to the predicated output variable i.e. $y = f(x)$, where y is a representation of x (Bishop, 2007). In this model, data $= \{ x1, x2, \dots, xn \}$, where $x \in \mathrm{R}^{\mathrm{d}}$, and x represents the user's attributes, coupon attributes and contextual attributes. The target y is represented by binary values, with 1 indicating accepted coupons and 0 indicating rejected coupons.

4.3 Supervised Learning Classifiers

A supervised learning classifier uses a predefined dataset for training purposes. The algorithm then predicts how an unknown input variable will be categorized according to the training dataset labels (Bishop, 2007). In this case, known user profiles were used as training data. If the supervised learning classifier is accurately trained, it will be able to predict the likelihood of a user accepting a coupon from a new, unlabeled user profile sample. In the following sections we describe the classifiers used in this project.

K Nearest Neighbor (KNN): This is a content-based algorithm that makes predictions based on similarities between the user's attributes and previous coupon buying decision. KNN works well on non-linear datasets. It predicts the class of the new observations based on the company they keep. Samples are labeled based on the identity of the 'k' nearest (most similar) neighbors (where k is a natural number). Distance can be measured in several ways, such as the Euclidean distance.

Support Vector Machine (SVM): This classifier mainly aims at detecting the hyperplane which generates the largest boundary between samples in the dataset. SVM is a powerful classifier, as it can be used with different kernels (linear, polynomial, sigmoid, and RBF). This, combined with the "kernel trick," allows for efficient

identification of relationships in high dimension space (Bishop, 2007).

Logistic Regression (LR): Is a classifier that predicts the value of unseen data based on its association with predictor variables. LR uses a sigmoid function to predict the target variable. Sigmoid function is defined as:

$$\sigma(z) = \frac{1}{1 + \exp(-z)}$$

Here the Z is the input features multiplied by a random number denoted by θ. In the case of our multi-feature datasets, the value of Z is calculated by the following function where X denotes the features.

$$Z = \theta_0 + \theta_1 X_1 + \theta_2 X_2 + \theta_3 X_3 + \ldots\ldots$$

Sigmoid function returns a value of 0 or 1 based on a threshold of 0.5. If the result is more than 0.5 then the output is 1 and 0 otherwise. The θ value gets updated in each iteration using the gradian decent formula. To measure how far the prediction is from the actual output, the following cost function is used:

$$J = -\frac{1}{m} = [\Sigma y^i log h^i + (1 - y)\log(1 - h^i)]$$

Where m is the training example, y is the actual output and h is the prediction output (Bishop, 2007).

Decision Tree: This classifier uses nodes to represent attributes and leaves to represent class labels. This model works recursively to find the best split to the fullest purity based on conditions. (Bishop, 2007).

Random Forest: This classifier extends the decision tree model. It works perfectly on multiclass classification problems where it selects the best result made by a variety of decision trees. Random forests usually produce highly accurate and interpretable results (Bishop, 2007).

Categorical Boosting Decision Tree (Catboost): This is a strong machine-learning technique that produces cutting-edge outcomes in a range of categorical real-world applications (Prokhorenkova et. al. 2018). It provides a framework that, in contrast to the standard technique, aims to solve for categorical features using the collective effort of decision tress ensemble to minimize the loss function iteratively using the greedy gradient boosting function that builds iteratively a sequence of approximations: $F^t: R^m \rightarrow R, t = 0,1, \ldots$, were F^t is obtained from the previous approximation F^{t-1} in an additive manner as follows:

$F^t = F^{t-1} + \alpha\, h^t$, where α is the step size and h^t is the base predictor (decision tress in this case). Then it used the following function to minimize the loss (Prokhorenkova et. al. 2018):

$$h^t = \underset{h \in H}{argmin}\, L(F^{t-1} + h) = \underset{h \in H}{argmin}\, EL(y, F^{t-1}(x) + h(x))$$

Categorical Feature: Is a discrete set of nonnumerical unique values called categories.

Categorical data encoding: A method for converting category features to numerical features, depending on the feature type. If the categorical data is ordinal, for example, a manual encoding method suggests preserving the data's underlying order. Otherwise, if the data is nominal and the order is irrelevant, OneHot, Label, or Dummy encoders can be used.

Hyperparameters: is a parameter whose value is used to determine the behaviour of a classifier. For instance, we can use the Support Vector Machine with different Kernels. Also, we can define the number of neighbours that the nearest neighbour classifier should consider and so on. The classifiers can be used with the default parameters or with a modified version of them. Hyperparameter tweaking, using methods such as Grid Search, can improve the effectiveness of classification models (Bishop, 2007).

4.4 Classifier Evaluation

Classifier evaluation techniques are used to ensure that a classification model is producing accurate and reproducible results, and that the classifier is not overfitting or underfitting the data. The methods used in this work are described below.

K-fold cross validation: Evaluates the anticipated performance of the model. First, split the dataset into K equal subsets. One subset is selected to validate the model, while the remaining subsets are used to train the model. This process continues until all the subsets are used to validate the model.

Confusion Matrix: A matrix that is used to compare a model's predictions to the actual identities of the training/test population. Common metrics used to evaluate model performance such as accuracy, precision, and recall can be calculated using the matrix.

		Actual	
		+	-
Predicted	+	TPs	FPs
	-	FNs	TNs

- **True Positive:** the user's decision to purchase a coupon is correctly classified as a yes.
- **False Positive:** the user's decision to purchase a coupon is incorrectly classified as a yes.
- **True Negative:** the user's decision to purchase a coupon is correctly classified as a no.
- **False Negative:** the user's decision to purchase a coupon is incorrectly classified as a no.

Predictive Value (PPV) or Precision:
A measure of the trustworthiness of correct positive predictions made. Precision is calculated as

$$PPV = \frac{TP}{TP + FP}$$

Recall: A measure to check the percentage of how often the model generates positive results for users who accepted the coupon.

$$\text{Recall} = \frac{TP}{TP + FN}$$

Accuracy: A measure of the model's ability to distinguish between labels. It is calculated as

$$\text{Accuracy} = \frac{TP + TN}{numbr\ of\ samples}$$

Area under the Receiver Operator Characteristic (AUC-ROC): a probability curve that compares the True

Positive Rate (TPR) to the False Positive Rate (FPR) at different thresholds. The ROC curve is summarised using the Area Under the Curve (AUC). The better the classifier's performance, the higher the AUC. If the AUC is closer to 1, the classifier can tell the difference between positive and negative values in the different classes. The classifier predicts false negatives and false positives if the ACU value is close to 0 (Bishop, 2007).

4.5 Interpretability

The interpretability of a Machine Learning system is defined by the degree to which a cause and effect can be identified within the system. In other words, it is the ability to comprehend what the model has learned to predict what will happen when the algorithm's input or parameters are modified. Interpretability is less important when the goal is strictly to make a classification based on a set of input features. If we wish to understand the relationship between the features and the dependant variable, however, interpretability is critical.

4.6 Feature Selection

Reducing the number of input variables is known as the feature selection process. It is a statistical method to select the input variables in data = $\{ x1, x2, \dots, xn\}$ that contribute the most to the model's training process. There are several feature selection methods that can be applied to the model. The model can be used with a variety of feature selection methods. Filter methods like chi-Square, Wrapper methods like forward and backward selections and Embedding methods like random forest feature selection are just a few examples. There are various methods that can be used for feature selection. Here, we will define the one that we used in our study.

Chi-square: Also known as, Chi-2, is a statistical measure that is used for the determination of dependency and similarity among two events. The aim of feature selection in machine learning is to select the input variables that depend greatly on the prediction. This means that the selected variables would have a greater effect on the prediction result. Chi-square is therefore used to select features in machine learning models (Bishop, 2007).

MinMax Scaler (Normalization): It is a method that is used in machine learning algorithms to scale the numerical values into a standard range. The normalizer scales each value of the input variables into a range between 0 and 1 (Bishop, 2007).

Dimensionality Reduction: It reduces the feature space by using linear combinations of the features in place of the original features. This method can improve both accuracy and runtime dramatically. Because the original features are lost, however, it often removes or reduces our ability to link individual features to the target variable (Bishop, 2007).

Linear Discriminant Analysis (LDA): Is a tool used for both dimensionality reduction and classification. LDA generates linear combinations of features that best preserve the characteristics of the original features (Bishop, 2007).

5. Approach

In this order, our project went through four stages: data examination, data preprocessing, feature selection including trying dimensionality reduction, testing the classifiers with a variety of hyperparameters and finally, achieving 95.7 percent accuracy.

5.1 Dataset Examination

To get started, we had to save the data as a comma-separated values (CSV) file. We used the python Pandas library to take a peek into the data and explore the data and its variables. The dataset is a supervised learning dataset with over 12000 instances and 26 attributes; this means there is an input variable X and an out variable y. The data was first converted into a data frame to allow for flexible analyzation. The data has a mix of numerical datatypes and string object datatypes. However, we want our data to be categorical with a mix of binary, ordinal and nominal types. There were some missing values represented as NaN. This data can be dropped or replaced by the most frequent label because missing values are insufficient and can cause sparse in the dataset and produce inaccuracies in the classification model.

5.2 Data Preprocessing

There are missing values in several columns. Primarily, the attribute "car" has 108 non-null values from a total of 12684 instances, more than 90 percent of the values are missing. This is a significant number of missing values and therefore this attribute is insufficient and can be removed from the dataset. The number of NaN values in the rest of the column wasn't as significant as the "car" attribute, so we will iterate through the data frame table and get all the columns with empty or NaN values, and then for each column the code is going to find the largest variable count and fill the empty values with the corresponding variable with maximum count. We also looked at columns with non-unique values. The attribute 'toCoupon_GEQ5min' has only one unique variable which won't help much in the encoding of the categorical variables. Therefore, that attribute was dropped. Lastly, we converted all our data to categorical data types in order to encode it as numerical values. When the order of the values matter, the categories are ordinal. After encoding, the data sequence should be retained. Thus, we used a manual encoding method. In manual label encoding, each category is turned into an integer number that indicates its order. In this dataset we have 7 ordinal categorical columns: Income, age, RestaurantLessThan20, Restaurant20To50, Bar, CoffeHouse, and CarryAway. We mapped each category with a numerical value to represent it and its order. We also noticed that 10 attributes have nominal categorical data, meaning that the categories do not have an inherent order. These Nominal attributes are: destination, passenger, weather, time, coupon, expiration, gender, maritalStatus, occupation, and education. Since we have 10 Nominal categorical columns, the one-hot encoding method is not recommended. It will increase the sparsity of the data. Thus, we will use the Label Encoding method via Sklearn,

as order does not matter. At the end, we converted all the data to a numerical datatype.

5.3 Feature Selection Methods

As mentioned previously, various methods can be used for feature selection. Knowing which methods will result with the best outcome was a challenge for us. For this, we followed the trial-and-error approach to find the best method for our dataset. Initially, we used a manual method based on a graph relation between each feature and the target output (Appendix B, Figure 6). This process only eliminated two features and made no impact on the accuracy of the classifier (Appendix C). Then, we applied several feature selection methods including, Chi-square (Appendix B, Figure 4), MinMax Scaler (Normalization), and Linear Discriminant Analysis (LDA). Chi-square reduced the features to only 7, showing a significant difference in their score comparing to the other features. MinMax normalization and LDA did not have any significant impact on the performance of the tradational classifiers. Thus, we decided not to apply them to out data.

5.4 Classifiers and Hyperparameters

To achieve the highest accuracy, we made an educated choice of the classifiers on which we tested our model. The dataset is not linear. Therefore, we did not use linear classifiers with it. Additionally, even though Neural Networks produce extremely accurate results, we decided not to use them. The reason for this was that they lack interpretability. Knowing the precise chain of reasoning of the user's profile behind the prediction of coupon purchase is an important part of this model, and it cannot be compromised (Wang et. al. 2017). So, based on the theory and our previous experiments with the classifiers, we narrowed our choice down to the following classifiers with modified hyperparameters for some of them. We list these classifiers according to their performance from lowest to highest:

K Nearest Neighbor: We used KNN with the default Euclidean Distance function, however we modified the default number of neighbours by using Gridsearchcv for finding best K value.

Support Vector Machine (SVM): We used SVM with the Radial Basis Function (RFB) kernel, as the dataset is not linearly separable. This was evident by the poor performance of the SVM with linear kernel that achieved zero percent AUC. With gemma equals 0.1, we maximized how far the influence of a single training example reaches. We have also set the cash size to 500MB since we have enough RAM of 13GB in Google Collaboratory. Additionally, since we have a lot of samples, we decided to give a per-class scores for each sample by setting the probability hyper-parameter to True.

Decision Tree: With this classifier, we fixed the random state to an integer to allow obtaining a deterministic behaviour during fitting. This approach seemed to be the most fit for the dataset, as interpretability is important here.

Logistic Regression: It is widely used in recommendation systems. Since our dataset have multi-features, we set the optimization algorithm to newton-cg solver. After OneHot encoding trial on the dataset, we set the multi-class hyper-parameter to 'ovr', so a binary problem is fit for each label.

Random Forest: The accuracy and interpretability of the predictions are crucial in this experiment. Thus, we chose this classifier as it suits the main objective of the project. With n_job = -1 to activate parallel processing, and max depth of 10 for each tree in the forest, the model achieved the second highest AUC at 81 percent. We used a trial-and-error method to determine the best max depth value for RF. Our evaluation was based on the classifier's performance with each max depth.

6. EXPERIMENTAL SETUP

Our experiments were performed using the in-vehicle coupon recommendation dataset obtained through the UCI Machine Learning Repository (Bache and Lichman, 2013). The columns, which each represent a user's attribute, a coupon attribute or a contextual attribute, serve as the features (independent variables) for our model. The last binary column labelled 'Y' of this dataset was used as the target/dependant variable.

We designed our classifier using Python 3.7 and the following libraries:

- Numpy (Numpy arrays)
- Pandas (Dataframes)
- Scikit Learn 0.24.1 (Machine Learning Models)
- PyCaret Python library for machine learning

We used Google Collaboratory notebook for our development environments. Our code is publicly available through GitHub at this link. We used two notebooks. The first one (*Traditional_classify*) to perform the classification with the traditional methods, and (*Catboost_classify*) to perform the classification using Catboost. Wrappers functions for the major components of our classification pipeline (loading the data, feature selection, dimensionality reduction and classification) can be found in the *Traditional_classify* notebook. In that notebook we started by preprocessing the data and encoding it. Please refer to section 5 for more details. Then we used the train_test_split function from sklearn. In our first trial, we started with the standard 80:20 for training and testing, then we changed it to 40:60. We did not notice any difference in the classifiers' performance, so we reverted to 80:20. In *Catboost_classify* we used the PyCaret Python library for data preprocessing, baseline model comparisons, hyperparameter tuning, and classification using Catboost. The classifiers accuracy was determined using an 8-fold cross validation, Confusion Matrix, and AUC-ROC in both notebooks.

7. RESULTS AND DISCUSSION

7.1 Results

Comparison with classical classifiers: We compare Catboost performance to the most suitable well-known machine learning classifiers. The details are provided in in section 5 and section 6. For all traditional classifiers, we encoded the categorical features using manual

encoding for ordinal data, and label encoding for nominal data. As for the Catboost classifier, the data pre-processing and encoding was done automatically using the PyCaret Python library. The parameter tuning and training were performed on 80 percent of the data and the testing was performed on the remaining 20 percent. The results were evaluated by accuracy, AUC, recall, and precision. Our Catboost classification model can determine the user's coupon purchase decision based on sample user's profiles taken from the dataset with 95.7 percent accuracy and 99 percent AUC, without dimensionality reduction (Appendix B, Figure 5). This method will reserve the interpretability of the model. Refer to Appendix A, Table 1 for the full results of this classifier. On the other hand, the traditional classifiers mentioned in section 4.3, were not able to achieve accuracy higher than 77 percent, even with the best hyper-parameter tuning and dimensionality reduction. After dimensionality reduction, the highest classification accuracy of 77 percent was achieved by Random Forest (RF). The rest of the classifiers achieved between 65-70 percent accuracy. We also measured accuracy on using data that had not undergone dimensionality reduction – since this process can distort the effect of individual features. The best accuracy achieved without dimensionality reduction was 74.7 percent, which was achieved using an RF model. The rest of the classifiers achieved between 65-68 percent accuracy (Appendix A, Table 1 & 2). The results were multiplied by 100 and rounded up to two decimal places.

7.2 Discussion

Our goal was to determine the likelihood of a user accepting a coupon based on user profiles in the dataset. In addition to deciding on the best method for encoding categorical data. Understanding the nature of the features, the link between them and the target output could lead to a classification model that can accurately identify the likelihood of coupon purchase in unlabeled samples. When first presented with the dataset, we struggled to achieve classification accuracies exceeding 74.7 percent using the traditional classifiers. We started by encoding the ordinal features manually to reserve the order and used label encoder on the nominal features. Then, we attempted to perform several layers of feature selection, including passing the data through a Chi2 filter, and scaling the data using MinMax scaler. Unfortunately, after Chi2 selected 7 out of 22 features, the performance of the classifiers did not improve. Next, we decided to examine the relation between each feature and the target output. We did this by plotting each feature against the target output (Appendix B, Figure 6). A full list of our observations is provided in Appendix C. This analysis suggested that only a few features did not have a lot of impact on the target variable. Thus, we decided to go back to using all the 22 features that remained after data preprocessing. Next, we studied the distribution of the training features and the target output. As shown in Appendix B, Figures 1 & 2, we noticed that the number of observations for each feature is balanced and all features varies across the population. Additionally, we have seen that target frequency is well balanced. Thus, the accuracy can be used as a good metric for model performance. Following that, we decide to OneHot encode the entire dataset after it has been assigned numerical values by various encoding methods, in the hopes that the classifiers will benefit from the unique binary vector value of each entry. Similarly, this strategy had no effect on the outcomes. To address the traditional classifiers performance issue, we decided to examine the hyper-paraments. The details of our hyper-parameters tuning are available under section 5.4. This series of steps did not enhance the performance of the five tradition classifiers. Our literature review revealed that the issue of classifying categorical data with multi-sourced features is well known (Wang et. al. 2017) and (Dorogush et. al. 2018). The main problem is that the traditional classifiers were basically designed to handle numerical data. Our study shows that these classifiers were not able to achieve high accuracy on this dataset even after trying several techniques such as feature encoding, feature selection, hyper-parameter tuning and dimensionality reduction. Both Decision Tree and Random Forest classifiers did not perform well, as they lack the ability of gradient boosting available in Catboost classifier. Finally, our last step was to start fresh by using the Pycaret library to pre-process the data. Then, compare Catboost with the five models using the Pycaret *best-model* function. Surprisingly, Catboost outperformed all of the classifiers by achieving a 95.7 percent accuracy score and 99 percent AUC score. This is because Catboost natively handles multivariate numerical and categorical data from large and diverse sources. It uses the collective effort of decision tress ensemble to minimize the loss function iteratively using the greedy gradient boosting function that builds a sequence of approximations. The details about this function are provided in section 4.3.

In addition, we compared our findings to those reported in (Wang et. al. 2017). The performance of our RF classifiers was comparable to that in (Wang et. al. 2017). Via 5-fold testing on the in-vehicle Recommendation dataset, they found that RF achieved an AUC of 80 percent. In 8-fold testing on the same dataset, our RF achieved an AUC of 81 percent. They also put SVM to the test on the Connect-4 categorical dataset from the UCI machine learning library, which had 67557 instances and 42 characteristics with no missing values. On Connect-4, SVM achieved 82 percent accuracy using 5-fold crossvalidation. Furthermore, we applied SVM on the UCI machine learning repository's in-vehicle Recommendation dataset, which had 12000 instances and 26 characteristics with missing values. On this dataset, SVM obtained 67 percent accuracy using 8-fold cross crossvalidation. As a result of the variations between the two datasets, comparing SVM performance may not be appropriate in this case. However, we should point out that SVM has a hard time dealing with missing values in a dataset.

8. CONCLUSION

Our results show that the Catboost classifier can be used to accurately predict the likelihood of coupon purchase

using categorical input features. Machine Learning algorithms are usually correlated with numerical datasets. However, this is not how most of the data in the world is represented. Numerous encoding methods of categorical data have been used on this dataset. However, this did not aid the performance of the traditional classifiers. This study shows the need for classification techniques that can successfully handle categorical features such as the gradient boosting algorithms. Further work is needed to ascertain the generalizability of these results.

9. FUTURE WORKS

Even though this model produces good results by using the gradient boosting decision tree algorithm, we cannot know, at this point, whether they will be generalizable.
We would like to better understand why the model performed well when classifying the data by Catboost but failed to produce similar results when we attempted classification using other classifiers. A study examining the encoding techniques of the categorical data and imputation of missing data may help to explain the differences in classifier performance. Furthermore, since KNN is a nearest neighbour model and RBF in SVM acts as a weighted nearest neighbour model, the classification of the new values in these two models are heavily influenced by the nearest neighbours. Thus, applying K-mean clustering might have an impact on the performance of these classifiers.

10. ACKNOWLEDGEMENT

We would like to thank scikit-learn and Pycaret for assisting us with the implementation of all the techniques used in this project. We would also like to thank Dr. Robin Gras for answering our questions about the project and providing the necessary datasets.

11. REFERENCES

[1] Bache and M. Lichman. UCI machine learning repository, 2013. URL http://archive.ics.uci.edu/ml

[2] Wang, T., Rudin, C., Doshi-Velez, F., Liu, Y., Klampfl, E., & MacNeille, P. (2017). A bayesian framework for learning rule sets for interpretable classification. The Journal of Machine Learning Research, 18(1), 2357–2393.,URL https://www.jmlr.org/papers/volume18/16-003/16-003.pdf

[3] Dorogush, A., Ershov, V., & Gulin, A. (2018). CatBoost: gradient boosting with categorical features support. arXiv preprint arXiv:1810.11363, URL https://arxiv.org/pdf/1810.11363.pdf

[4] "Pattern Recognition and Machine Learning“, Chris Bishop, Springer, 2007. URL: http://users.isr.ist.utl.pt/~wurmd/Livros/school/Bishop

[5] Prokhorenkova, L., Gusev, G., Vorobev, A., Dorogush, A., & Gulin, A. (2018). CatBoost: unbiased boosting with categorical features. In *Advances in Neural Information Processing Systems*. Curran Associates, Inc. URL https://papers.nips.cc/paper/2018/file/14491b756b3a51daac41c24863285549-Paper.pdf

[6] Author, L.Khalil, C.Dikie (2021). Prostate Cancer Detection Using Machine Learning Techniques. Unpublished paper.

12. APPENDIX

12.1 Appendix A – Results of various classifiers

	Classifier	Accuracy	AUC	Recall	Precision
KNN	K Neighbors	0.6550	0.6978	0.7221	0.6858
SVM-L	SVM -Linear Kernel	0.6718	0.0000	0.7623	0.6933
SVM-RBF	SVM – RBF Kernel	0.6715	0.7714	0.8025	0.6838
DT	Decision Tree	0.6799	0.6745	0.7164	0.7182
LR	Logistic Regression	0.6828	0.7363	0.7672	0.7014
RF	**Random Forest**	**0.7473**	**0.8181**	**0.8178**	**0.7563**
CatBoost	**CatBoost**	**0.9572**	**0.9903**	**0.9685**	**0.9414**

Table 1 – Model Evaluation Metrics

	Classifier	Accuracy	AUC	Recall	Precision
SVM-RBF	SVM – RBF Kernel	0.6569	0.7579	0.8234	0.6585
KNN	K Neighbors	0.6671	0.7225	0.7879	0.6783
DT	Decision Tree	0.6846	0.6968	0.7157	0.7257
LR	Logistic Regression	0.7081	0.7435	0.7802	0.7331
RF	Random Forest	0.7740	0.8311	0.8515	0.7272

Table 2 – Accuracies with Dimensionality Reduction

12.1 Appendix B – Figures

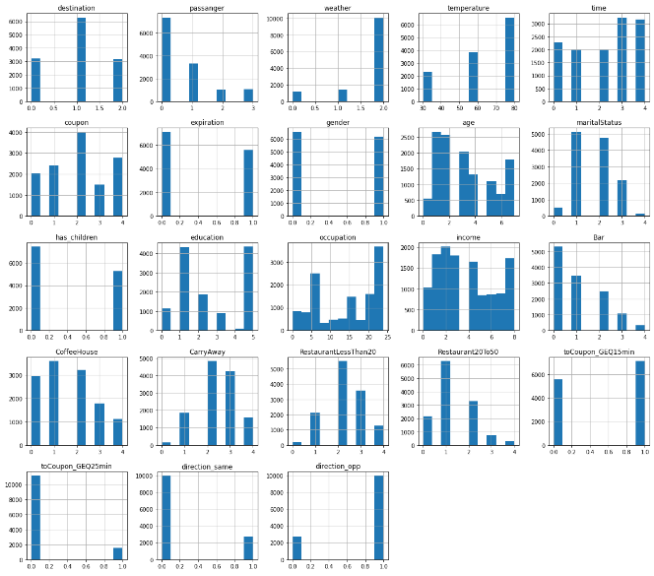

Figure 1 – Data distribution in Training set

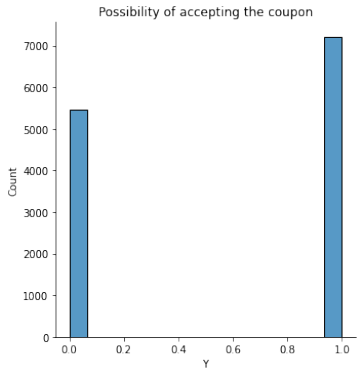

Figure 2 – Data distribution in Target class

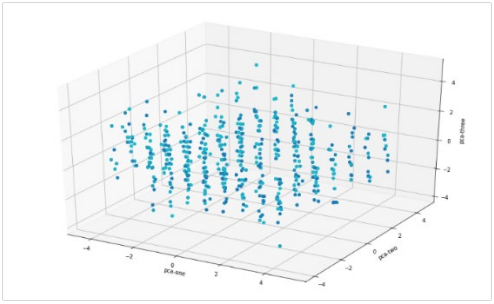

Figure 3 – Data visualization in 3D

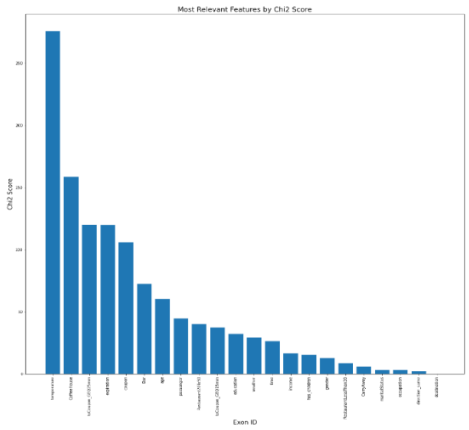

Figure 4 – Feature scores using Chi2

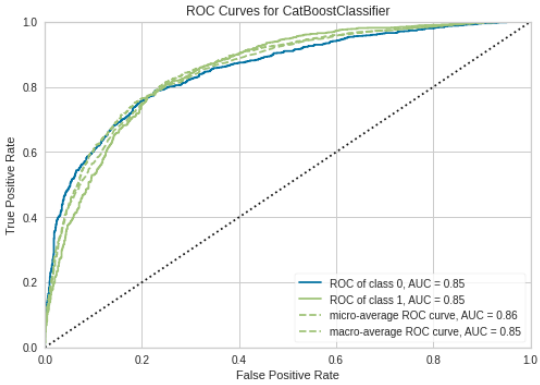

Figure 5 – ROC-AUC for Catboost classifier

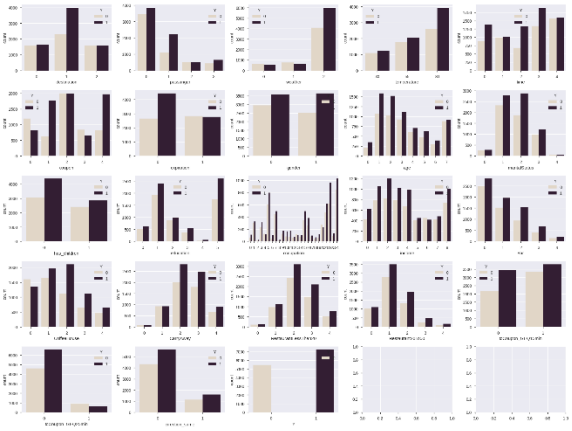

Figure 6 – Relation between feature and target

12.2 Appendix C – Observation of Features relations to the target

1. User attributes

- gender: Both genders have the same rates.
- age: drivers between 21 to 36 years old accept more coupons.
- MaritalStatus: Single drivers accept the coupons more.
- has_children: Driver's who do not have children are more likely to accept coupons.
- education: Some college, Bachelor or high school graduate are more likely to accept the coupon.
- occupation: drivers who are unemployed, students, work in sales, or computers have the highest tendency to accept the coupon.
- income: middle class drivers have a higher tendency to accept the coupon.
- Bar: the lower the frequency of visits the higher possibility of accepting the coupon.
- CoffeeHouse: moderate frequency of visits has a higher possibility of accepting the coupon.
- CarryAway: moderate frequency of visits has a higher possibility of accepting the coupon.
- RestaurantLessThan20: moderate frequency of visits has a higher possibility of accepting the coupon.
- Restaurant20To50: if the driver went less than once, they have a higher tendency of accepting the coupon.

2. Contextual attributes

- passenger: The possibility of accepting the coupon is higher when the driver is alone.
- destination: The possibility of accepting the coupon is higher when the drivers have no urgent place to go.
- weather: The possibility of accepting the coupon is higher when it is sunny.
- temperature: The possibility of accepting the coupon is higher when it is 80 degrees.
- time: The probability of accepting the coupon is higher, if the time is 6pm or 7pm, and not during the day.
- toCoupon_GEQ15min: not much difference
- toCoupon_GEQ25min: if the distance is less than 25mins to the coupon location it will be more likely to be accepted.
- direction_same: if the direction is not the same, the coupons get purchased more.

3. Coupon attributes

- coupon: 1) The possibility of accepting coupons for restaurants that cost less than 20 dollars, carry out and take away is high. 2) The acceptance and rejection of coffee house coupons is equal. 3) coupons for bars and restaurants that cost between 20-50 dollars have a higher rejection rate.
- expiration: Coupon that expire in one day have higher acceptance rate than the ones expiring in two hours.