Industry Project Report

Classification and Analysis of User Behaviour on Restaurant Discount Offers

Tejas Ajay Parse

20-04-2024

Industry Mentor

Vaibhav Kesharwani

DatStek

Contents

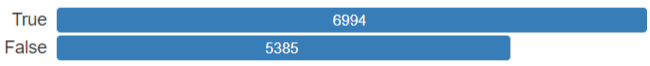
Abstract

This project investigates the behavior of users and drivers to ascertain their likelihood of accepting restaurant offers while driving. The study employs a rigorous methodology involving comprehensive data preprocessing to ensure data quality and relevance. An exploratory data analysis (EDA) is conducted to gain insights into the dataset's features. Subsequently, several classifiers are trained using varied feature sets ranging from 4 to 27, and their performances are compared. The report presents the top 7 models identified in the study. The primary objective of this project is to gain practical experience in handling raw real-world data and to explore the application of different classifiers in predictive modeling. Through this endeavor, valuable insights are obtained into user and driver behavior, contributing to a deeper understanding of the dynamics involved in accepting restaurant offers while driving.

Dataset Description

Feature	Type	Meaning
Offer expiration	categorical	When Does the office expire
income_range	categorical	Range of Income of User
no_visited_Cold drinks	categorical	Number of times visited for cold drinks
travelled_more_than_15mins_for_offer	categorical	Traveled more than 15 mins for offer
Restaur_spend_less_than20	categorical	Number of times spend less than 20 dollar in restaurant
Marital Status	categorical	Married or Single
restaurant type	categorical	Type like Cold Drink Shop or Take away restaurant
age	categorical	Age
Prefer western over chinese	categorical	Prefer western over chinese
travelled_more_than_25mins_for_offer	categorical	Traveled more than 25 mins for offer
no_visited_bars	categorical	Number of times visited bars
gender	categorical	Gender
car	categorical	Car model name
restuarant_opposite_direction_house	categorical	Restaurant is in opposite direction of travel

restuarant_same_direction_house	categorical	Restaurant in same direction of house?
Cooks regularly	categorical	Cook regularly or not
Customer type	categorical	Whom do you prefer to go with
Qualification	categorical	Qualification
is foodie	categorical	Are you a foodie
no_Take-aways	categorical	Number of times opted for a take-away
Job/Job Industry	categorical	Type of industry you work with
has Children	categorical	Has children or not
visit restaurant with rating (avg)	categorical	Rating of restaurant user visits on avg
temperature	integer	Current Temperature
Restaur_spend_greater_than20	categorical	No of times spent more than 20 dollars
Travel Time	Integer	Travel time to restaurant
Climate	categorical	Current Climate
Prefer home food	categorical	Yes or No to prefer home food
drop location	categorical	Heading to which direction



Label: Offer Accepted

Dataset Preparation

Now we prepare the dataset for training classifiers

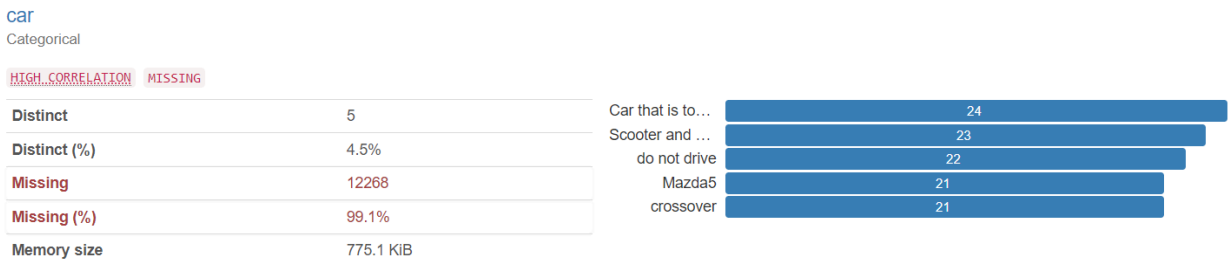
Dataset Preprocessing

Dataset processing involves preparing raw data for analysis through steps like cleaning, transforming, and organizing it. Cleaning addresses missing values and errors, while transformation involves converting types and scaling features. Organizing ensures data is structured for analysis.

Exploratory Data Analysis

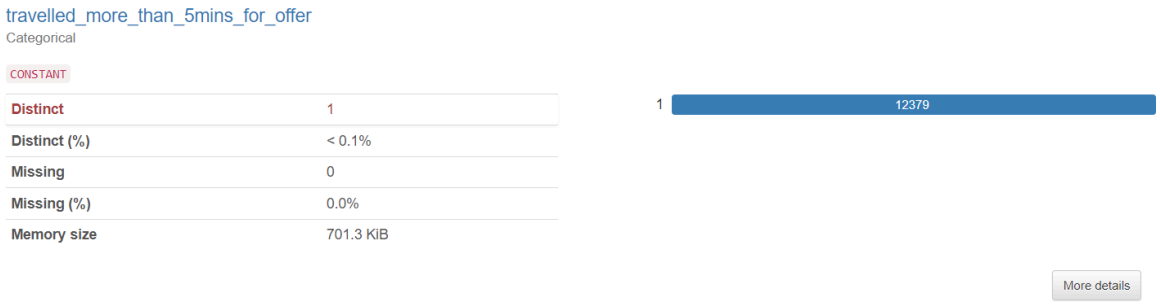
Exploratory Data Analysis (EDA) follows, summarizing data with statistics and visualizations to uncover patterns and relationships. EDA guides further analysis and modeling decisions.

1. Removing “cars” Feature



We removed the 'car' feature because it was missing in 99.1% of the columns, making it largely irrelevant for the classification problem.

2. Removing “travelled_more_than_5mins_for_offer”



We decided to remove the feature 'travelled_more_than_5mins_for_offer' from the dataset because upon analysis, it was found that all users had traveled more than 5 minutes to avail the offer. As a result, this feature became a constant feature with identical values across all rows, providing no useful variation or predictive power for our analysis.

3. Removing “restuarant_opposite_direction_house”

Correlation is a statistical measure that gauges the relationship between two variables. It provides insights into how changes in one variable correspond to changes in another. A negative correlation indicates that as one variable increases, the other tends to decrease, and vice versa. In the dataset, the variables "**restaurant_opposite_direction_house**" and "**restaurant_same_direction_hous**e" exhibit a strong negative correlation, with a coefficient of approximately **-0.809**. This implies that as the distance to a restaurant in the opposite direction of the house increases, the distance to a restaurant in the same direction of the house tends to decrease, and vice versa.

Given the high negative correlation between these variables, it suggests redundancy in the information they provide. Removing one of these variables simplifies the analysis and modeling process by reducing multicollinearity and potentially enhancing model interpretability and performance.

4. Removing Rows with missing values

`no_visited_Cold drinks`	has 198 (1.6%) missing values
`no_Take-aways`	has 144 (1.2%) missing values
`Restaur_spend_greater_than20`	has 160 (1.3%) missing values

Removed rows with missing values in the columns "**no_visited_cold_drinks**," "**no-takeaways**," and "**restaurant_spend_greater_than20**" due to their relatively low proportion of missing data. Specifically, these columns contained 1.6%, 1.2%, and 1.3% missing values, respectively. Since the percentage of missing values is relatively small, removing these rows has minimal impact on the overall dataset size while improving the reliability of subsequent analyses

Training Models

We have a total of 27 Features. After trying several classification models including training a Neural Network, here are top 7 models. Additionally, the results are attached to each model with First Row denoting the top K features taken to train the model and second row being the accuracy obtained. **Chi-squared test** is being used to pick top k features.

DecisionTreeClassifier

The DecisionTreeClassifier is a simple yet powerful tool for classification tasks in machine learning. It creates a tree-like structure where data is split based on features, helping to categorize data into different classes. It's commonly used for tasks like predicting customer churn or classifying spam emails

4	6	8	10	12	14	16	18	20	22	24	26	27
0.6150	0.6103	0.5972	0.5925	0.5913	0.5913	0.5887	0.5714	0.5633	0.5464	0.5451	0.5409	0.5455

KNeighborsClassifier

K Nearest Neighbors (KNN) is a straightforward algorithm used for classification and regression tasks in machine learning. It predicts the class or value of a new data point based on the majority class or average value of its nearest neighbors.

4	6	8	10	12	14	16	18	20	22	24	26	27
0.5172	0.5417	0.5193	0.5460	0.5392	0.5404	0.5532	0.5650	0.5693	0.5620	0.5616	0.5748	0.5680

RandomForestClassifier

Random Forest Classifier builds multiple decision trees during training and combines their predictions to improve accuracy and robustness. It's known for its high accuracy, resistance to overfitting, and ability to identify important features in the data.

4	6	8	10	12	14	16	18	20	22	24	26	27
0.6150	0.6103	0.6010	0.5909	0.5930	0.5917	0.5934	0.5853	0.5798	0.5688	0.5735	0.5824	0.5718

XGBClassifier

The XGB Classifier, or Extreme Gradient Boosting Classifier, is a machine learning algorithm renowned for its speed, performance, and accuracy. It operates by iteratively building a series of decision trees, each one learning from the errors of its predecessors through gradient descent optimization. By incorporating regularization techniques and supporting parallel processing, XGB Classifier effectively controls overfitting and achieves scalability for large datasets. It is widely used across various classification tasks where high predictive accuracy and efficiency are paramount.

4	6	8	10	12	14	16	18	20	22	24	26	27
0.6188	0.6197	0.6091	0.6158	0.6129	0.6125	0.6002	0.6002	0.6074	0.6069	0.6031	0.6069	0.6057

LogisticRegression

Logistic Regression is a statistical method used for binary classification tasks in machine learning. It models the probability of a binary outcome based on predictor variables, assuming a linear relationship between them and the log-odds of the outcome. Logistic Regression is valued for its simplicity, interpretability, and effectiveness in various applications, although it performs optimally when the relationship between predictors and the outcome is approximately linear.

4	6	8	10	12	14	16	18	20	22	24	26	27
0.6129	0.6141	0.6023	0.6069	0.6069	0.6069	0.6137	0.6141	0.6129	0.6133	0.6112	0.6137	0.6129

AdaBoostClassifier

AdaBoostClassifier, short for Adaptive Boosting Classifier, is an ensemble learning algorithm used for classification tasks in machine learning. It combines multiple weak learners, typically decision trees with limited depth, to create a strong learner. During training, AdaBoost assigns weights to each training example, initially setting them equally. It then iteratively trains weak learners on the data, adjusting the weights of misclassified examples to focus more on difficult instances. In the end, AdaBoost combines the predictions of all weak learners through a weighted majority vote to make the final prediction. This process creates a robust classifier that tends to perform well even on complex datasets.

4	6	8	10	12	14	16	18	20	22	24	26	27
0.6129	0.6141	0.6031	0.6069	0.6069	0.6069	0.6133	0.6129	0.6120	0.6120	0.6103	0.6137	0.6146

SVC

SVC, or Support Vector Classifier, is a supervised learning algorithm used for classification tasks in machine learning. It works by finding the hyperplane that best separates different classes in the feature space. SVC aims to maximize the margin between the classes while minimizing classification errors.

4	6	8	10	12	14	16	18	20	22	24	26	27
0.6150	0.6108	0.6082	0.6082	0.6213	0.6222	0.6201	0.6222	0.6230	0.6222	0.6260	0.6273	0.6302

Results

Method	Best Accuracy
SVC	0.6302
XGBClassifier	0.6196
DecisionTreeClassifier	0.6149
RandomForestClassifier	0.6149
AdaBoostClassifier	0.6145
LogisticRegression	0.6141
KNeighborsClassifier	0.5747

Conclusion

Best Classifier (**SVC**) Analysis that gave 0.6302 Accuracy.

		Actual	
		Positive	Negative
Predicted	Positive	1090	602
	Negative	271	398

	Precision	Recall	F1-Score	Support
No	0.59	0.40	0.48	1000
Yes	0.64	0.80	0.71	1361
Accuracy			0.63	2361
Macro Avg	0.62	0.60	0.60	2361
Weighted Avg	0.62	0.63	0.61	2361

In conclusion, this project aimed to predict whether users would pick restaurant offers using various classification models. The process involved data cleaning, exploratory data analysis (EDA) to identify redundant columns, and training multiple classification models. The top 7 models, trained on various subsets of features ranging from 4 to 27, were evaluated. Despite exploring several models, the Support Vector Classifier (SVC) emerged as the most effective, achieving an accuracy of 0.63, as shown by the confusion matrix. The confusion matrix allows you to visualize the performance of the model. In this case, it shows that the model is correct 63% of the time. The precision for the "Yes" class is 0.64, which means that out of all the times the model predicted "Yes", 64% of those predictions were actually correct. The recall for the "Yes" class is 0.80, which means that out of all the actual "Yes" cases, the model predicted "Yes" 80% of the time. Overall, this project provided valuable hands-on experience in handling raw data, exploring different models, and evaluating their performance, ultimately enhancing understanding and proficiency in the field of machine learning.