Starbucks Capstone Project

Data Science Nanodegree

Ilya Kalagin
June 30, 2023

Definition

Project Overview

Welcome to the Starbucks Capstone Challenge project, which is the final project for the Udacity Data Scientists Nanodegree Program. Throughout this project, I have delved into the datasets and discovered valuable insights. However, I must acknowledge that one challenge was the need to generate and imagine tasks without a deep understanding of the business context. Effective communication with stakeholders and close collaboration with the business is essential for success in a real-world project scenario. It is crucial to actively engage in discussions, ask pertinent questions, and work closely with stakeholders to gain a deeper understanding of the datasets and formulate a clear problem statement that aligns with the organization's goals and objectives. This lack of context may have impacted the depth of understanding and resulted in potential gaps in the analysis.

Nonetheless, my aim throughout this project remains to showcase the power of data analysis in driving informed decision-making and providing actionable recommendations for businesses. Despite the challenge of generating tasks without a comprehensive understanding of the business, I have utilized the available data to uncover valuable insights into customer behavior, identify patterns, and develop strategies that align with customer preferences. In addition, I have developed a model that predicts whether a particular user is likely to utilize a given offer. This model enhances the understanding of user behavior and can further inform decision-making processes.

Join me on this journey as we explore the Starbucks Capstone Challenge datasets, navigate data complexities, and recognize the importance of stakeholder collaboration. I am committed to providing valuable insights that contribute to Starbucks' success and enhance the customer experience through the utilization of the predictive model.

Problem Statement and Metrics

The project aims to achieve the following objectives:

1) Establish an ETL pipeline.
2) Analyze and transform the original data.
3) Conduct exploration analysis and visualization.
4) Develop a data model capable of predicting whether the customer will accept the offer or not.

In this study, we are addressing a classification problem where our goal is to develop a data model capable of accurately predicting whether the customer will accept the offer or not. To evaluate the performance of our model, we will employ several commonly used metrics:

Accuracy:
Accuracy = (TP + TN) / (TP + TN + FP + FN)

Recall:
Recall = TP / (TP + FN)

Precision:
Precision = TP / (TP + FP)

Where:
TP - True Positives
TN - True Negatives
FP - False Positives
FN - False Negatives

F1-score:
F1-score = 2 * (Precision * Recall) / (Precision + Recall)

When evaluating the performance of a classification model, metrics such as accuracy score, F-score, recall, and precision are commonly used. Each of these metrics provides a different aspect of the model's performance:

The accuracy score measures the proportion of correctly classified instances out of the total number of instances. It provides a general overview of the model's predictive accuracy.

Precision measures the ability of the model to identify positive instances correctly. It calculates the ratio of true positives to the sum of true positives and

false positives. Precision indicates the model's ability to avoid false positives and is useful when minimizing false positives is crucial.

Recall, also known as sensitivity or true positive rate, measures the ability of the model to correctly identify all positive instances. It calculates the ratio of true positives to the sum of true positives and false negatives. The recall is necessary when it is essential to identify all positive instances, and false negatives need to be minimized.

The F-score, or F1-score, is a metric that combines precision and recall into a single value. It represents the harmonic mean of precision and recall and provides a balanced evaluation of the model's performance. The F-score is beneficial when there is a trade-off between precision and recall and a balanced evaluation is required.

In summary, the accuracy score provides an overall measure of model performance, precision focuses on minimizing false positives, recall emphasizes minimizing false negatives, and the F-score combines precision and r into a single metric. Choosing the appropriate metrics depends on the specific requirements and priorities of the classification problem at hand. It is recommended to consider multiple metrics together to evaluate the model's performance comprehensively.

Analysis

Analyze and transform the original data.

Data Sets:
The data is contained in three files:
* portfolio.json - containing offer ids and meta data about each offer (duration, type, etc.)
* profile.json - demographic data for each customer
* transcript.json - records for transactions, offers received, offers viewed, and offers completed

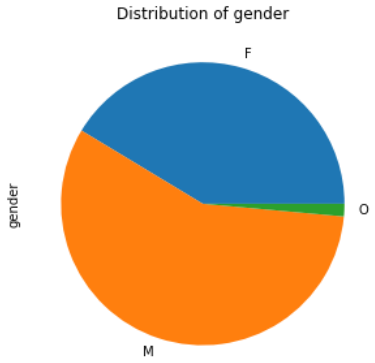
Here is the schema and explanation of each variable in the files:
portfolio.json
* id (string) - offer id
* offer_type (string) - type of offer ie BOGO, discount, informational
* difficulty (int) - minimum required spend to complete an offer
* reward (int) - reward given for completing an offer
* duration (int) - time for offer to be open, in days
* channels (list of strings)

	reward	channels	difficulty	duration	offer_type	id
0	10	[email, mobile, social]	10	7	bogo	ae264e3637204a6fb9bb56bc8210ddfd
1	10	[web, email, mobile, social]	10	5	bogo	4d5c57ea9a6940dd891ad53e9dbe8da0
2	0	[web, email, mobile]	0	4	informational	3f207df678b143eea3cee63160fa8bed
3	5	[web, email, mobile]	5	7	bogo	9b98b8c7a33c4b65b9aebfe6a799e6d9
4	5	[web, email]	20	10	discount	0b1e1539f2cc45b7b9fa7c272da2e1d7
5	3	[web, email, mobile, social]	7	7	discount	2298d6c36e964ae4a3e7e9706d1fb8c2
6	2	[web, email, mobile, social]	10	10	discount	fafdcd668e3743c1bb461111dcafc2a4
7	0	[email, mobile, social]	0	3	informational	5a8bc65990b245e5a138643cd4eb9837
8	5	[web, email, mobile, social]	5	5	bogo	f19421c1d4aa40978ebb69ca19b0e20d
9	2	[web, email, mobile]	10	7	discount	2906b810c7d4411798c6938adc9daaa5

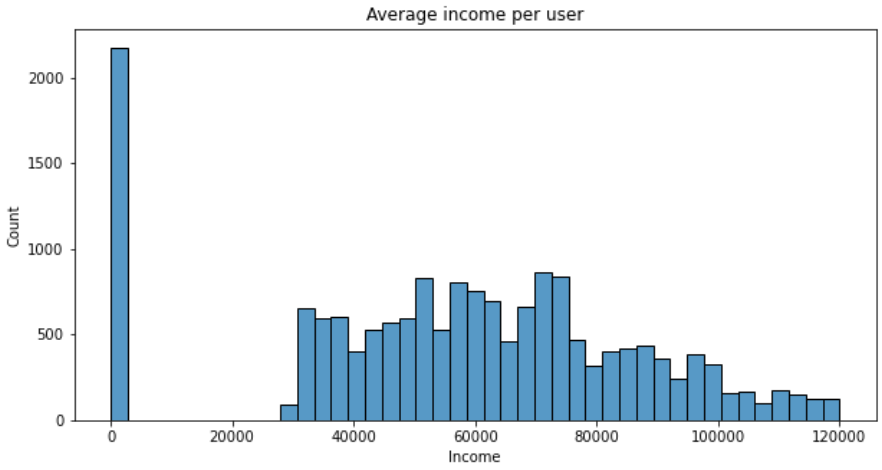
The 'portfolio' dataset consists of 10 entries and includes two categorical variables: 'channels' and 'offer_type'. It is worth noting that the 'channels' variable involves the value 'email' for all records, which makes it unimportant for future analyses.

profile.json
* age (int) - age of the customer
* became_member_on (int) - date when customer created an app account
* gender (str) - gender of the customer (note some entries contain 'O' for other rather than M or F)
* id (str) - customer id
* income (float) - customer's income

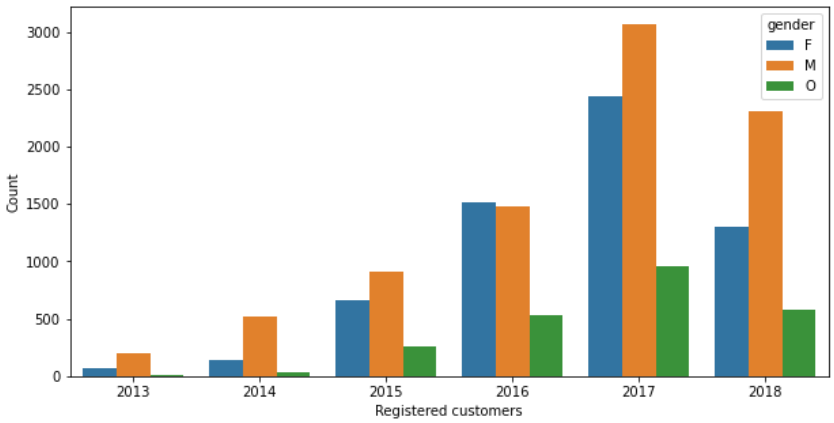
	gender	age	id	became_member_on	income
0	None	118	68be06ca386d4c31939f3a4f0e3dd783	20170212	NaN
1	F	55	0610b486422d4921ae7d2bf64640c50b	20170715	112000.0
2	None	118	38fe809add3b4fcf9315a9694bb96ff5	20180712	NaN
3	F	75	78afa995795e4d85b5d9ceeca43f5fef	20170509	100000.0
4	None	118	a03223e636434f42ac4c3df47e8bac43	20170804	NaN
5	M	68	e2127556f4f64592b11af22de27a7932	20180426	70000.0
6	None	118	8ec6ce2a7e7949b1bf142def7d0e0586	20170925	NaN
7	None	118	68617ca6246f4fbc85e91a2a49552598	20171002	NaN
8	M	65	389bc3fa690240e798340f5a15918d5c	20180209	53000.0
9	None	118	8974fc5686fe429db53ddde067b88302	20161122	NaN



A pie chart was generated to visualize the gender distribution in the 'profile' Data Frame. By grouping the Data Frame based on 'gender' and calculating the count of each gender category, the pie chart accurately represents the proportion of each gender category. This graphical representation offers a clear overview of the gender distribution within the dataset.



A histogram plot was created to visualize the distribution of the 'income' column in the 'profile' Data Frame. The code utilized seaborn to generate the plot and set the appropriate labels for the x-axis and y-axis. The resulting plot provided a clear representation of the income distribution of the users in the dataset. Zero values are unknown data.



A bar plot was generated to visualize the count of registered customers based on the 'year' and 'gender' in the 'profile' DataFrame. The code grouped the Data Frame by these two variables and calculated the count for each combination. Using seaborn's barplot(), the data was plotted with 'year' on the x-axis, 'count' on the y-axis, and 'gender' as the hue. The resulting plot effectively represents the distribution of registered customers across different years and genders.

transcript.json

* event (str) - record description (ie transaction, offer received, offer viewed, etc.)

* person (str) - customer id

* time (int) - time in hours since start of test. The data begins at time t=0

* value - (dict of strings) - either an offer id or transaction amount depending on the record.

	person	event	value	time
0	78afa995795e4d85b5d9ceeca43f5fef	offer received	{'offer id': '9b98b8c7a33c4b65b9aebfe6a799e6d9'}	0
1	a03223e636434f42ac4c3df47e8bac43	offer received	{'offer id': '0b1e1539f2cc45b7b9fa7c272da2e1d7'}	0
2	e2127556f4f64592b11af22de27a7932	offer received	{'offer id': '2906b810c7d4411798c6938adc9daaa5'}	0
3	8ec6ce2a7e7949b1bf142def7d0e0586	offer received	{'offer id': 'fafdcd668e3743c1bb461111dcafc2a4'}	0
4	68617ca6246f4fbc85e91a2a49552598	offer received	{'offer id': '4d5c57ea9a6940dd891ad53e9dbe8da0'}	0
5	389bc3fa690240e798340f5a15918d5c	offer received	{'offer id': 'f19421c1d4aa40978ebb69ca19b0e20d'}	0
6	c4863c7985cf408faee930f111475da3	offer received	{'offer id': '2298d6c36e964ae4a3e7e9706d1fb8c2'}	0
7	2eeac8d8feae4a8cad5a6af0499a211d	offer received	{'offer id': '3f207df678b143eea3cee63160fa8bed'}	0
8	aa4862eba776480b8bb9c68455b8c2e1	offer received	{'offer id': '0b1e1539f2cc45b7b9fa7c272da2e1d7'}	0
9	31dda685af34476cad5bc968bdb01c53	offer received	{'offer id': '0b1e1539f2cc45b7b9fa7c272da2e1d7'}	0

Besides, data transformations were performed to enhance the data quality and extract meaningful insights. These transformations involved various steps, such as cleaning the data to remove errors and inconsistencies, filtering out irrelevant information, splitting or merging fields to restructure the data, standardizing formats for consistency, and performing calculations to derive new variables.

These transformations were crucial in preparing the data for analysis and ensuring that it was suitable for the intended purposes of the study.

Conduct exploration analysis and visualization

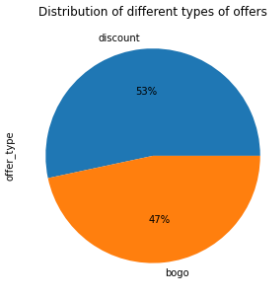
Analyzing the data, we have grouped the records based on whether individuals received an offer, read the recommendations, and subsequently made purchases or completed the offers. This allows us to gain insights into the utilization of the offers by different individuals.

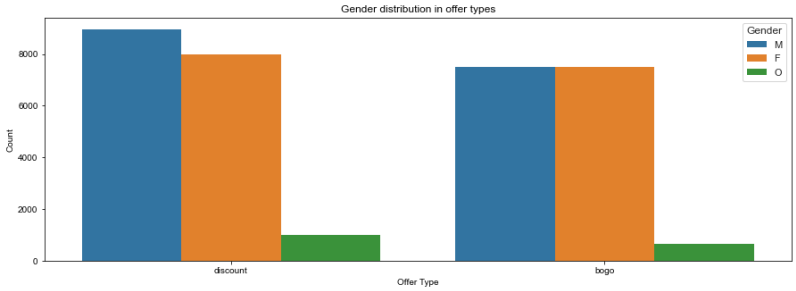
event	user_id	value	time
offer completed	33579	33579	33579
offer received	76277	76277	76277
offer viewed	57725	57725	57725
transaction	138953	138953	138953

A notable finding is that there are cases where individuals fulfilled the conditions of the promotion without being aware of the special offer or receiving any information about it. Out of the total records, it is evident that people who actively engaged with the information and took advantage of the offer account for 28,724 records. However, there are also 4,855 records where individuals fulfilled the offer's conditions unconsciously.

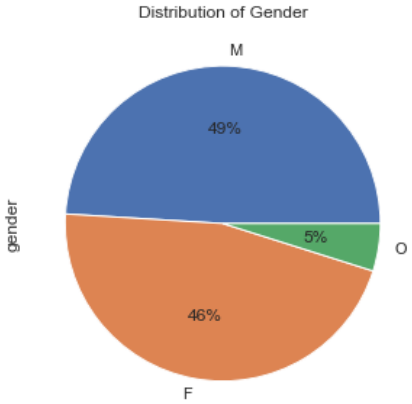
Furthermore, the analysis reveals that over 25% of the received offers were not even viewed by the recipients. This means that a significant portion of people remained unaware of the proposals and did not have the opportunity to consider or take advantage of them.

These findings emphasize the importance of understanding which individuals have actively utilized the offers, as it highlights instances where individuals fulfilled the promotion's requirements unknowingly or were completely unaware of the offer's existence. This information can provide valuable insights for improving the effectiveness of future promotional campaigns and ensuring that targeted individuals are properly informed and engaged with the offers.

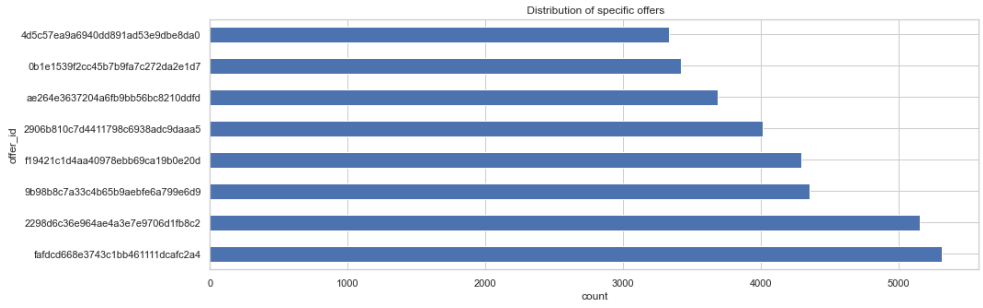




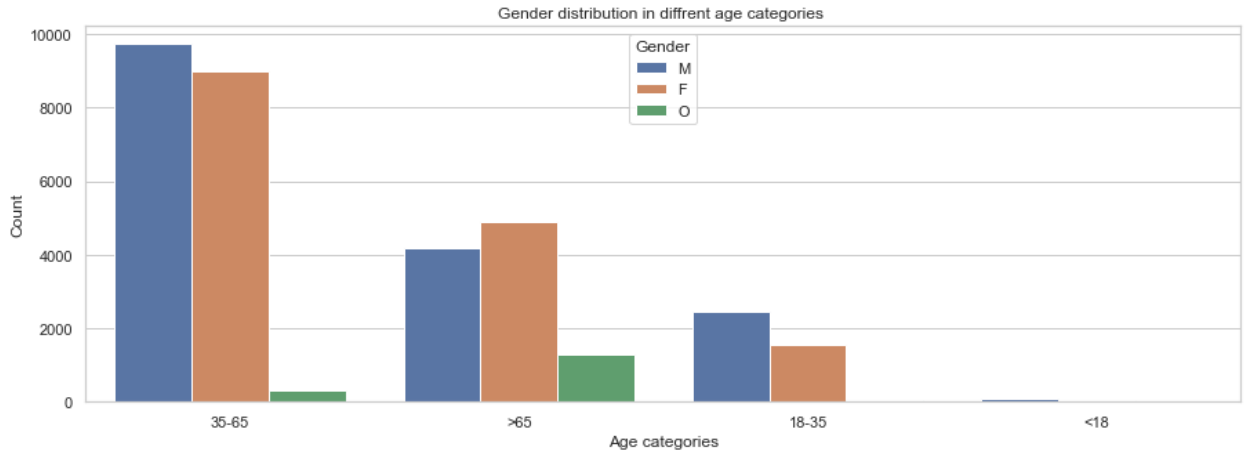
In the next step, we utilized a pie chart to showcase the distribution of various offer types in the dataset and a count plot to visualize the gender distribution within each offer type. These visualizations provide a clear representation of the proportion of each offer type and allow for a better comprehension of the overall distribution. It is worth noting that in this dataset, only two types of offers were observed, and the focus was on examining the data related to completed offers.



A exploration of the gender distribution in the dataset was conducted. We generated a pie chart that represents the distribution of gender. The chart provides a clear visualization of the proportion of each gender category, enabling a better understanding of the overall gender distribution in the dataset.



In the study, an analysis of the distribution of specific offers in the dataset was performed. A horizontal bar chart was created to display the count of each individual offer. This visualization offers valuable insights into the distribution patterns of offers and enhances our understanding of the popularity and utilization of different offer types.



Next, the subsequent stage involves augmenting the dataset by introducing a new column that denotes distinct age groups. The dataset is then scrutinized according to age, with the age categories comprising "<18," "18-35," "35-65," and ">65." The distribution of gender within these age groups is meticulously examined utilizing a count plot. M- male, F- female, O – others. This visual representation furnishes valuable observations regarding gender distribution within various age brackets.

Data Modeling

For modeling, our goal is to solve the classification problem related to the effectiveness of special offers on influencing people to make a purchase. The target variable we will focus on is called "offer_works." To train our classifier and simulate the situation where people have read information about offers, we aim to determine which offers would lead to a purchase and identify the individuals who might be affected by our special offer.

To train our model, we split the dataset into training and testing sets using the train_test_split function. The training set (X_train, y_train) will be used to train our decision tree classifier, represented by the 'dtree' object. We fit the classifier to the training data and make predictions on the testing set (X_test) using the predict method. The classification report and confusion matrix are printed to evaluate the performance of the decision tree model.

```
  precision    recall  f1-score   support

           0       0.45      0.45      0.45      1477
           1       0.91      0.90      0.90      8597

    accuracy                           0.84     10074
   macro avg       0.68      0.68      0.68     10074
weighted avg       0.84      0.84      0.84     10074


Confusion matrix
[[ 670  807]
 [ 830 7767]]
```

However, based on the results, it is evident that the model's performance is not satisfactory. One of the reasons for this outcome is the imbalanced nature of the dataset. The majority class (labels=1) has a significantly higher count (28,724) compared to the minority class (labels=0) with only 4,855 instances.

To address this issue, we decide to add data from 'df_not_viewed,' which represents offers that were not completed by the users. These instances can be labeled as '0' since they did not lead to a purchase. We concatenate 'df' and 'df_not_viewed' using the concat function, resulting in an expanded dataset with a shape of (49,571, 25).

With the updated dataset, we proceed to redefine our training and testing sets, X and Y, respectively. We then repeat the train-test split process and utilize the RandomForestClassifier for our model. The random forest algorithm is known for its ability to handle imbalanced datasets effectively.

After fitting the random forest model on the training data, we make predictions on the testing set and evaluate the performance using classification_report and confusion_matrix. The results demonstrate improved accuracy, precision, recall, and f1-score compared to the decision tree model.

```
precision    recall  f1-score   support

           0       0.83      0.89      0.86      4145
           1       0.92      0.87      0.89      5770

    accuracy                           0.88      9915
   macro avg       0.88      0.88      0.88      9915
weighted avg       0.88      0.88      0.88      9915

Confusion matrix
[[3699  446]
 [ 740 5030]]
```

The result is better!

To further optimize our model, we employ GridSearchCV, which allows us to search for the best combination of hyperparameters. In this case, we focus on parameters such as the number of trees (n_estimators), minimum samples required to split a node (min_samples_split), minimum samples required at a leaf node (min_samples_leaf), class weights for handling class imbalance (class_weight), and the minimum weighted fraction of samples in a leaf node (min_weight_fraction_leaf).

By creating a pipeline with the classifier and specifying the parameter grid, we apply GridSearchCV to our dataset. The best parameters obtained from the grid search are {'clf__class_weight': 'balanced_subsample', 'clf__min_samples_leaf': 1, 'clf__min_samples_split': 2, 'clf__min_weight_fraction_leaf': 0.1, 'clf__n_estimators': 10}. We use these optimal parameters to make predictions on the testing set and evaluate the model's performance once again.

Surprisingly, the results show a decrease in accuracy, precision, recall, and f1-score compared to the random forest model without hyperparameter tuning. There could be several reasons for this outcome, including the default parameter settings.

Tabular form of metrics

DecisionTreeClassifier

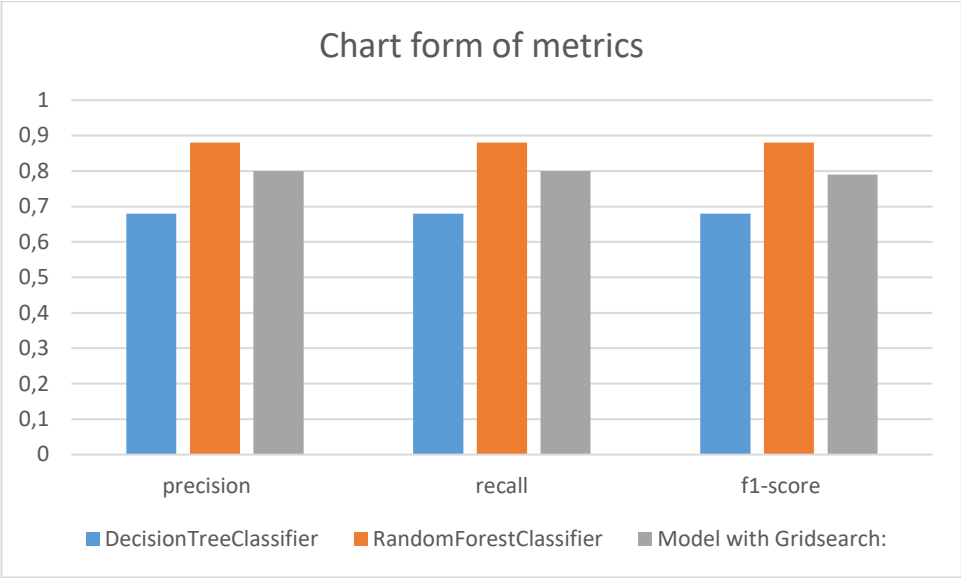
	precision	recall	f1-score	support
0	0.45	0.45	0.45	1477
1	0.91	0.90	0.90	8597
accuracy			0.84	10074
macro avg	0.68	0.68	0.68	10074
weighted avg	0.84	0.84	0.84	10074

RandomForestClassifier

	precision	recall	f1-score	support
0	0.83	0.89	0.86	4145
1	0.92	0.87	0.89	5770
accuracy			0.88	9915
macro avg	0.88	0.88	0.88	9915
weighted avg	0.88	0.88	0.88	9915

Model with Gridsearch:

	precision	recall	f1-score	support
0	0.71	0.87	0.78	4145
1	0.89	0.74	0.81	5770
accuracy			0.79	9915
macro avg	0.80	0.80	0.79	9915
weighted avg	0.81	0.79	0.80	9915



macro avg

It's important to note that this project is for educational purposes and serves as general guidance. The data modeling process is typically more complex and iterative in real-world projects. It involves thorough data exploration, feature engineering, extensive experimentation with various models and algorithms, rigorous evaluation, and fine-tuning hyperparameters.

Real-world projects often require a deeper understanding of the specific domain, careful consideration of potential biases or confounding factors in the data, and implementing more sophisticated techniques like cross-validation, ensembling, or advanced deep learning architectures. Furthermore, the success of a real-world project relies on collaboration with domain experts, data scientists, and stakeholders, as well as adherence to ethical guidelines and data privacy regulations.

Improvement

Gain a deeper understanding of the business context: To overcome the challenge of a limited business context, engaging with stakeholders and domain experts is essential. Seek their input and insights to understand better the business's goals, objectives, and challenges. This understanding will help formulate clear problem statements and identify relevant variables for analysis.

Incorporate additional data sources: Consider incorporating other data sources that may provide more context and insights into customer behavior. For example, customer feedback, social media, or external demographic data can enrich the analysis and provide a more comprehensive view of the customers.

Explore more advanced modeling techniques: While the project utilized decision trees and random forests, consider exploring other advanced modeling techniques such as gradient boosting, support vector machines, or neural networks. These techniques may offer improved performance and better handling of complex relationships within the data.

Conclusion

In conclusion, the Starbucks Capstone Challenge project has provided valuable insights into customer behavior and the effectiveness of promotional offers. The lack of a deep understanding of the business context posed a challenge, highlighting the importance of effective communication and collaboration with stakeholders in real-world projects. Despite this challenge, the project showcased the power of data analysis in driving informed decision-making and providing actionable recommendations. The analysis of the original dataset involved data transformation steps to enhance data quality and extract meaningful insights. Exploration analysis and visualization provided insights into customer utilization of offers, gender distribution, offer types, and age groups. These insights highlighted the need to improve offer awareness and engagement among customers. The data modeling phase focused on solving the classification problem of predicting offer effectiveness. The initial decision tree model showed unsatisfactory performance due to the imbalanced dataset. By incorporating additional data and utilizing the random forest classifier, the model's performance improved significantly. Further optimization using GridSearchCV was attempted but did not yield better results in this particular case.

It's important to note that this project serves as a general guidance and educational purpose. Real-world projects involve more complexity, iterative processes, and consideration of domain-specific factors. Approaching real-world projects with a systematic and iterative mindset, continuous validation, and refinement of models based on feedback and new insights are crucial for success.