Classification of Online Shopper Intention

Thompson Pham

CS 4375

Spring 2024

Data Description

Introduction

The Online Shoppers Purchasing Intention Dataset offers a comprehensive view into the dynamics of e-commerce and online consumer behavior, showing invaluable insights into the different factors that shape purchasing decisions within the digital landscape. The dataset was donated on 8/30/2018, it contains 12,330 instances, each presenting a distinct interaction between online shoppers and an e-commerce platform.
By using the power of classification modeling, we can unlock the potential of this dataset to predict online shoppers' purchasing intentions. With the help of machine learning techniques, businesses can gain insights to optimize their marketing strategies, refine website design, and improve user experience. This, in turn, facilitates the augmentation of conversion rates and drives sustainable revenue growth.

Within this dataset, 84.5% (10,422) of sessions are classified as negative instances, signifying instances where online browsing did not culminate in a purchase. Conversely, the remaining 15.5% (1,908) represent positive instances, indicating sessions that concluded with successful purchases. Which makes sense, since most users will be either just browsing or using each page to compare with similar products. As a result, a majority of them will be less prone to buying products.

Predictor Attributes

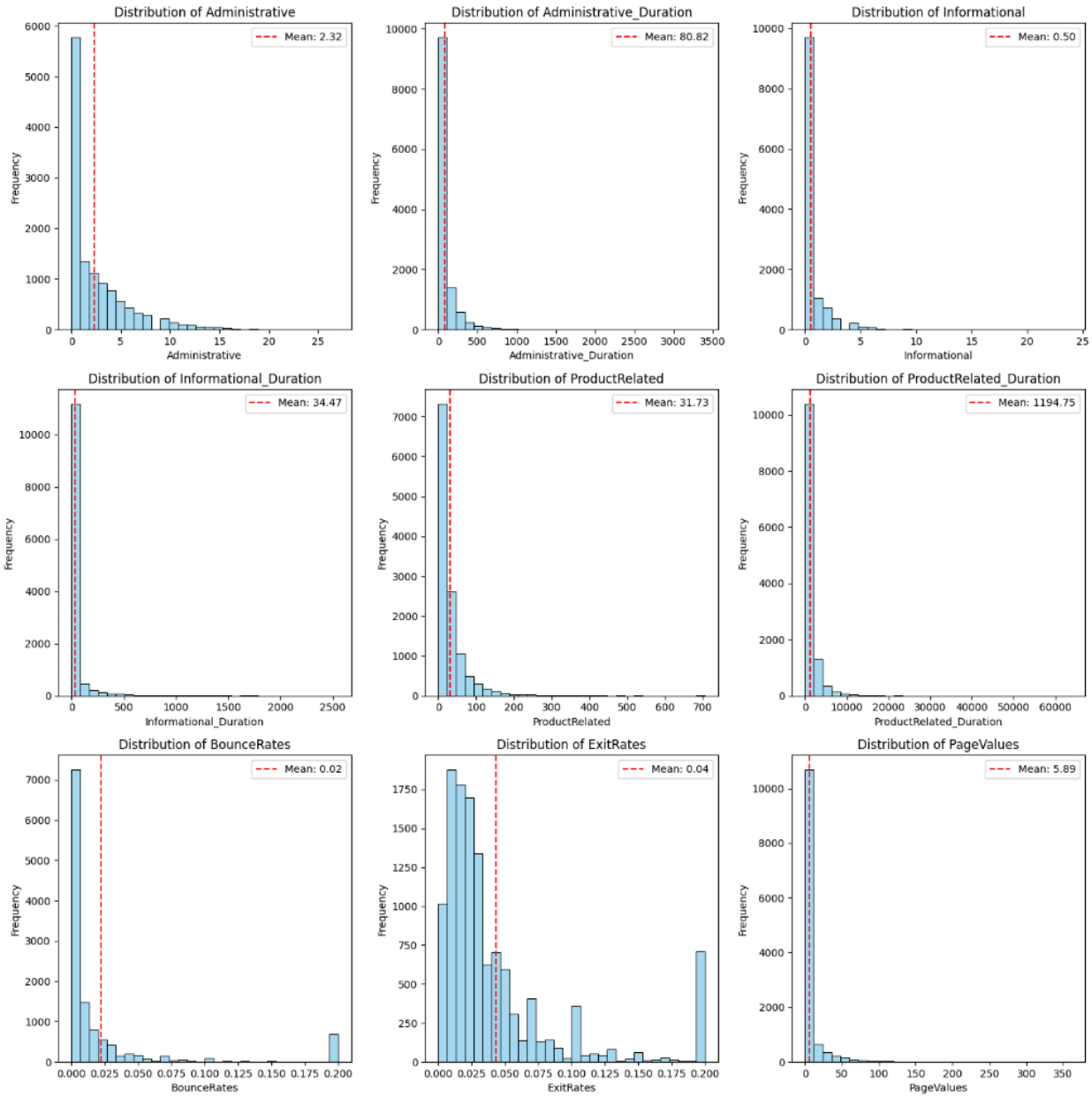
- **Administrative:** Number of pages user visited labeled under administrative tag
- **Administrative_Duration:** Time spent on administrative pages
- **Informational:** Number of pages user visited labeled under informational tag
- **Informational_Duration:** Time spent on informational pages
- **ProductRelated:** Number of pages user visited labeled under product related tag
- **ProductRelated_Duration:** Time spent on product related pages
- **BounceRates:** Percentage of users who enter the website of the given tag and exits out without doing anything
- **ExitRates:** Percentage of pageviews that stop at a given page
- **PageValues:** Page Value is the average value for a page that a user visited before landing on the goal page or completing an Ecommerce transaction (or both).
- **SpecialDay:** The relative closeness to special days or holidays (eg Christmas or Valentines Day) where more transactions are likely to occur
- **Month:** Contains the month a page is viewed
- **OperatingSystems:** An integer value to represent the os the user was on when viewing the page.
- **Browser:** An integer value to represent the browser the user was using to view the page.
- **Region:** An integer value representing which region the user is located in.
- **TrafficType:** An integer value representing what type of traffic the user is categorized into.
- **VisitorType:** A string to classify visitors as either New Visitor, Returning Visitor, or Other.

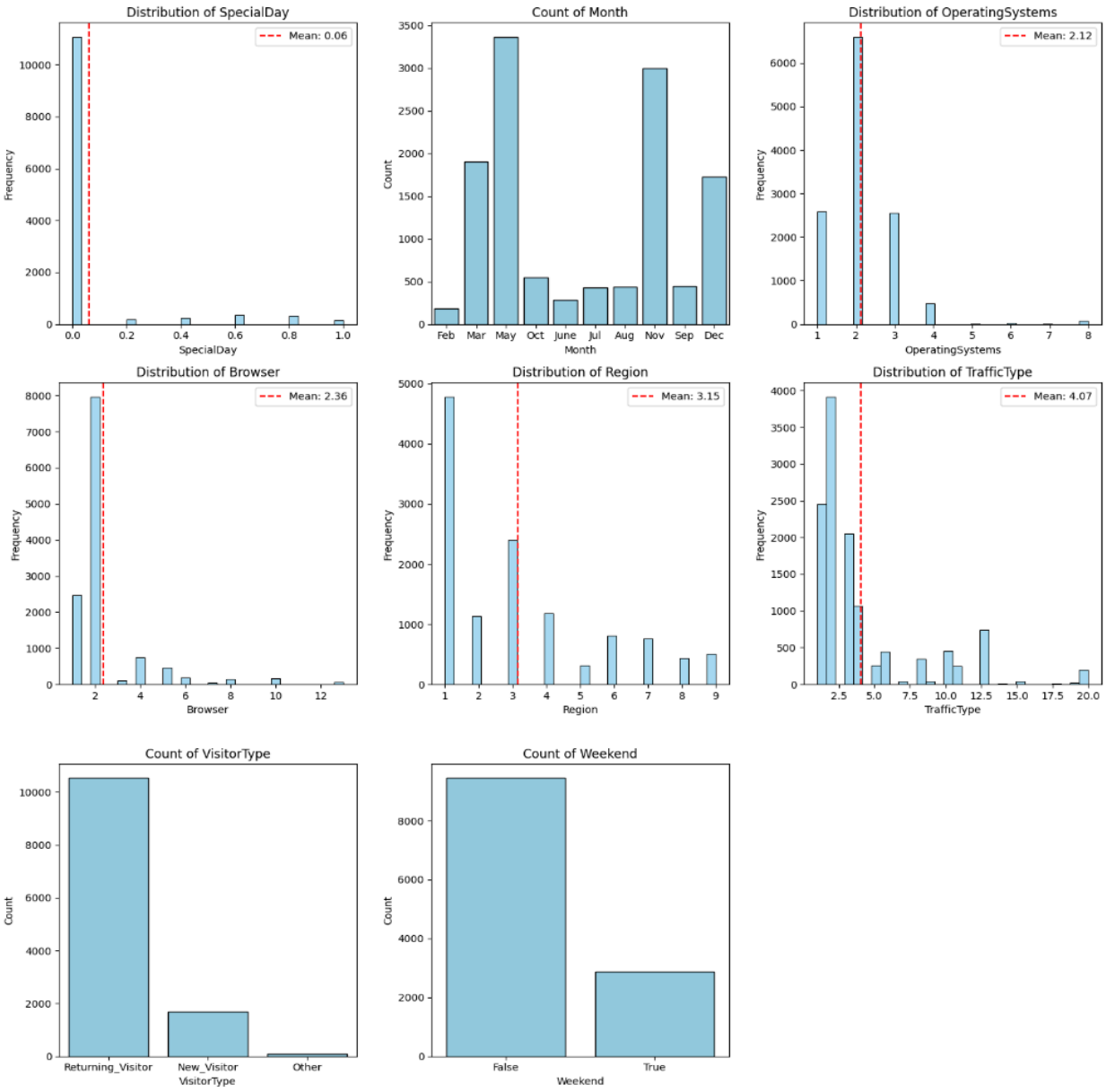
- **Weekend:** A boolean representing if its a weekend or not

Target Attribute

- **Revenue:** A boolean representing if a transaction is completed or not

Exploratory Analysis





Findings:

Based on a majority of distributions around the attributes, they align with the notion that a majority of users don't result in a finished transaction. This is shown by the general left skew of most of the attributes, compared to ones which were either not really correlated with the topic or told the same story in a different manner. For instance exit rates, if it were to skew to the left it meant more people would've stayed on the pages and possibly bought items. However it was right skewed, entailing the opposite. The mean for the most part tells a similar story, excluding the product related duration attribute being quite high. This deviation could entail the amount of time users spend looking at pages of related products. They are more likely to complete a transaction by the end of it.

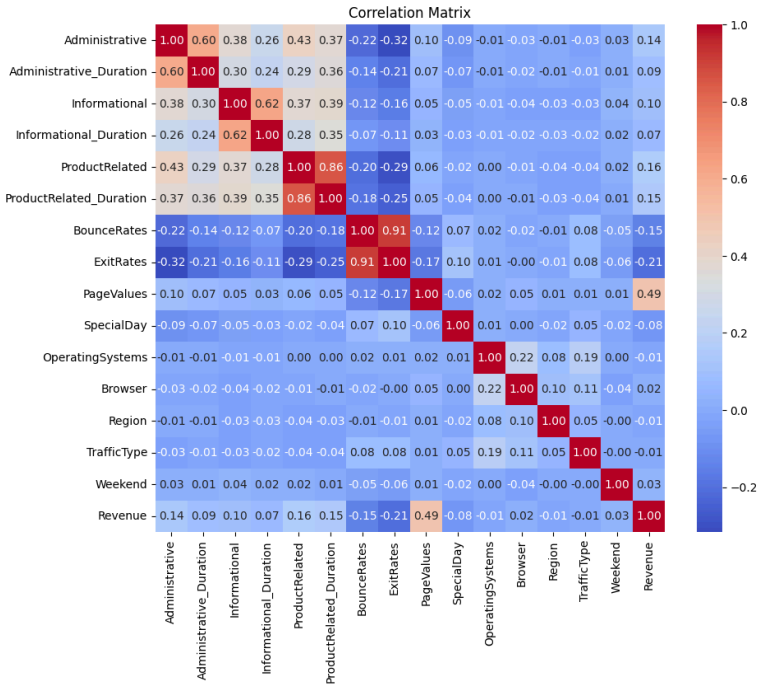
```
Mean of continuous numerical features:
Administrative                2.315166
Administrative_Duration      80.818611
Informational                 0.503569
Informational_Duration       34.472398
ProductRelated               31.731468
ProductRelated_Duration    1194.746220
BounceRates                   0.022191
ExitRates                     0.043073
PageValues                    5.889258
SpecialDay                    0.061427
OperatingSystems              2.124006
Browser                       2.357097
Region                        3.147364
TrafficType                   4.069586
dtype: float64

Variance of continuous numerical features
Administrative             1.103425e+01
Administrative_Duration    3.125085e+04
Informational              1.613297e+00
Informational_Duration     1.981036e+04
ProductRelated             1.978070e+03
ProductRelated_Duration    3.662130e+06
BounceRates                2.351117e-03
ExitRates                  2.361624e-03
PageValues                 3.447868e+02
SpecialDay                 3.956808e-02
OperatingSystems           8.305129e-01
Browser                    2.949039e+00
Region                     5.767640e+00
TrafficType                1.620199e+01
dtype: float64
```

When looking at variance, since the distribution is predominantly in favor of visitors who don't complete a transaction. The instances where we see a high variances high values may indicate attributes that are indicative of users buying and finishing a transaction. Since it deviates from the general norm of the dataset of non-complete transactions.

However we also need to be cautionary around high variance values since it can leave an impact on our overall model. Such influences could include the model learning and capturing noise in the training data, making our overall predictions less reliable, and possibly introducing complexity. This could cause our model to overfit and capture the general test dataset best. But won't be as effective when it comes to real world applications.

One thing I found quite interesting when looking at the correlation matrix is that the attributes that had a higher variance looked to have a bit more elevated correlation as opposed to the attributes with low varience. However, a high variance doesn't entirely lead to a high correlation since they each independently derive information differently.



Attributes that we should highlight when looking at the correlation matrix are Pagevalues, ExitRates, and BounceRates since they have a higher correlation in relation to the target variables compared to the other predictor variables. This high correlation can help make accurate predictions when building a model. However we need to be a bit alert around the suite of different pages and their respective duration, since there's a high correlation between predictor attributes possibly entailing multicollinearity. This can cause the model to be less interpretable and stable, which in turn may cause overfitting. While the variables from Month to Weekend has

quite a low correlation. With such a low variance, it may not contribute as much to our model when it comes to predicting which in turn may be possible candidates for removal.

Models

In this project, I employed four distinct classification algorithms—logistic regression, decision trees, random forests, and gradient boosting—to effectively utilize various attributes in classifying visitors based on their likelihood of completing a transaction.

Initially, the data preprocessing phase involved handling categorical and binary variables such as Revenue, Month, VisitorType, and Weekend. These variables were transformed using either label encoding or one-hot encoding, converting them into numerical representations for enhanced interpretability by the models.

Subsequently, a range of hyperparameters was defined for each algorithm, allowing them to select optimal parameters for improved predictive performance. Additionally, scaling was applied to standardize the weights across different attributes, enhancing the overall stability of the models.

The dataset was split into an eighty percent training set and a twenty percent testing set. These sets were then utilized to train the four different models. The code provided outputs the best hyperparameters discovered during the tuning phase for each model, and the trained models were evaluated using various performance metrics.

	model_name	metrics	Is_Revenue(False)	Is_Revenue(True)
0	Logistic Regression	precision	0.90	0.73
1	Logistic Regression	recall	0.97	0.46
2	Logistic Regression	f1-score	0.93	0.56
3	Decision Tree	precision	0.92	0.68
4	Decision Tree	recall	0.94	0.61
5	Decision Tree	f1-score	0.93	0.64
6	Random Forest	precision	0.90	0.77
7	Random Forest	recall	0.97	0.49
8	Random Forest	f1-score	0.94	0.60
9	Gradient Boost	precision	0.92	0.73
10	Gradient Boost	recall	0.96	0.57
11	Gradient Boost	f1-score	0.94	0.64

Legend:

- class 0 (visitors who did not complete a transaction)

- class 1 (visitors who completed a transaction)

Logistic Regression:

Accuracy: 87.31%

Summary: The logistic regression model achieved a respectable accuracy of 87.31%. It demonstrated higher precision and recall for class 0 compared to class 1. The model's f1-score was higher for the majority class, indicating better performance in predicting non-transactions.

Decision Tree:

Accuracy: 88.85%

Summary: The decision tree model achieved an accuracy of 88.85%. It exhibited higher precision, recall, and f1-score for class 0, indicating better predictive performance for non-transactions. However, its performance for predicting transactions (class 1) was relatively lower.

Random Forest:

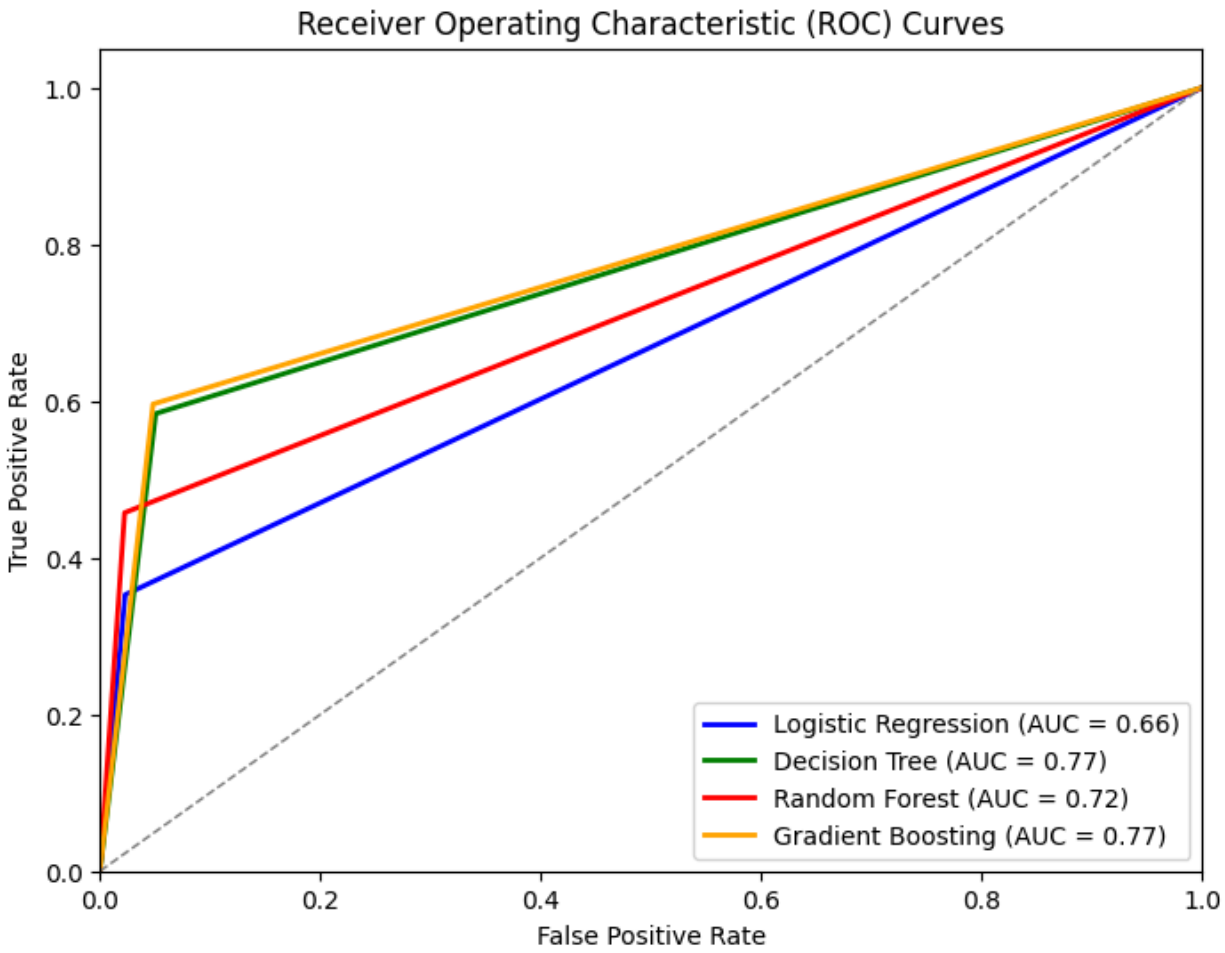
Accuracy: 88.93%

Summary: The random forest model achieved an accuracy of 88.93%. It displayed higher precision and recall for class 0, indicating good performance in predicting non-transactions. However, its performance for predicting transactions (class 1) was weaker compared to class 0.

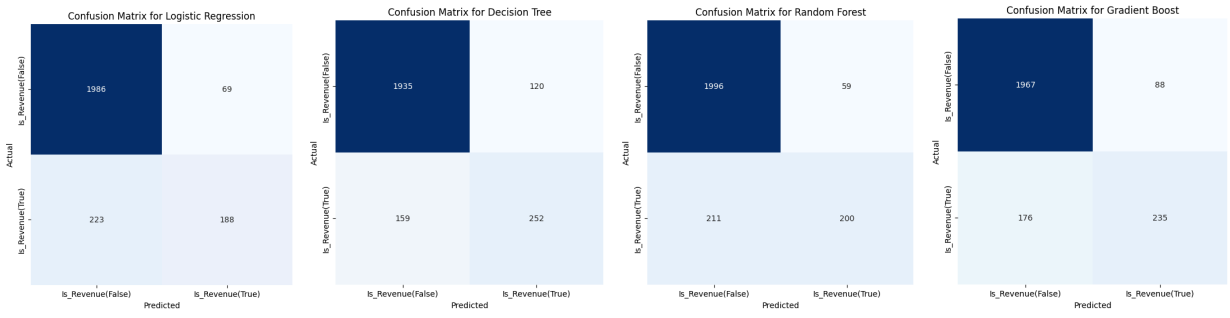
Gradient Boosting:

Accuracy: 89.13%

Summary: The gradient boosting model achieved the highest accuracy of 89.13%. Similar to the decision tree and random forest, it exhibited higher precision, recall, and f1-score for class 0, indicating better performance in predicting non-transactions. However, its performance for predicting transactions (class 1) was relatively weaker compared to class 0.



Area under the ROC curve (AUC) helps measure the performance of the classification model. Helping represent the degree of separability from users who complete and do not complete transactions. Looking at the graph above, we can see that logistic regression is not as effective as the others with the lowest AUC value followed by random forest. With decision trees and gradient forest being the highest and tied together. A given AUC value of 0.77 suggests that the model can moderately separate the two different types of users. Using the confusion matrices below we can see that as we move from left to right, the amount of misclassifications decreases. This reflects the general accuracy metric we found above with logistic regression being the lowest and gradient boosting being the highest.



But when comparing them to the ROC we see that the decision tree edges over the random forest. This is due to the fact that even though random forest has more correct classifications, the frequency of misclassifications in one class is more pronounced than in the other. As opposed to when the decision tree model misclassified the users, more often but it is more balanced between class 1 and class 0 resulting in a higher AUC score.

Conclusion: After looking at the different evaluation metrics, when it comes to overall performance in gradient boosting in this case is the best model for the task. It offers a nice balance of accuracy and predictive performance without the overly misclassifying one class over the other. But in case of time constraint, resource constraint, and scalability, the decision tree is a competitive alternative that is marginally less accurate. But offers some of the features seen in gradient boosting in terms of classification.

Citations

- “Online Shoppers Purchasing Intention Dataset.” *UCI Machine Learning Repository*, archive.ics.uci.edu/dataset/468/online+shoppers+purchasing+intention+dataset. Accessed 8 May 2024.