Project Report

Decoding Digital Purchase Intent

By: Priyanka Bhosale

Date: April 25, 2024

Table of Contents

1. Introduction

In today's digital era, where online shopping has become ubiquitous, gaining insight into the factors influencing customer purchase decisions is paramount for businesses striving to maintain competitiveness. By analyzing customer behavior and intent, businesses can create targeted marketing strategies tailored to resonate with their specific audience segments.

Studying browsing patterns and demographic data, businesses can unveil valuable insights that inform marketing strategies, website enhancements, and overall customer experience optimization. For instance, identifying key features such as page visit duration, product-related page views, or the timing of visits may reveal strong correlations with purchase intent. Armed with this knowledge, businesses can refine their website design and fine-tune marketing campaigns to enhance conversion rates.

This strategic approach not only amplifies the effectiveness of marketing initiatives but also fosters business growth by attracting and retaining a larger customer base.

2. Scope and Goals

Predicting revenue can provide valuable insights into the effectiveness of marketing campaigns, website design, and overall customer engagement strategies.

Additionally, predicting revenue can help businesses forecast future earnings and make informed decisions about resource allocation and budgeting.

In this project, we have identified the task (of predicting Revenue) as a classification problem because we are aiming to predict a binary outcome: whether a customer made a purchase (revenue = 1) during their visit to the website or did not make a purchase (revenue = 0).

The goal of this project to develop predictive model(s) that can effectively classify customers and provide valuable insights for business optimization and decision-making.

3. Dataset Details

The dataset “online_shoppers_intention.csv” (attached in the project package on iCollege) for this project was sourced from the UC Irvine Machine Learning Repository and contains e-commerce user information.

The dataset comprises 12,330 records, with each row representing a session belonging to a different user over a one-year period.

To ensure the data's integrity and avoid biases specific to campaigns, each row corresponds to a unique user within the one-year timeframe.

The dataset consists of 18 features in total, including 10 numerical and 8 categorical variables.

Numerical Features

Column	Details
Administrative	No. of administrative pages that the user visited
Administrative_Duration	Amount of time, in seconds, spent in this category of pages
Informational	No. of informational pages that the user visited
Informational_Duration	Amount of time, in seconds, spent in this category of pages
ProductRelated	No. of product-related pages that the user visited
ProductRelated_Duration	Amount of time spent, in seconds, in this category of pages
BounceRates	Percentage of visitors who enter the website through that page and exit without triggering any additional tasks
ExitRates	Percentage of page views that end at that specific page
PageValues	Average value for a page that a user visited before landing on the goal page or completing the transaction (or both)
SpecialDay	Closeness of the browsing date to special days or holidays

Categorical Features

Column	Details
Month	Contains the month the pageview occurred
OperatingSystems	Operating system that the user used when viewing the page
Browser	Browser of the visitor
Region	Geographic region from which the session has been started by the visitor
TrafficType	Type of traffic the user is categorized into
VisitorType	Whether a visitor is New Visitor, Returning Visitor, or Other
Weekend	Whether the session is on a weekend

Revenue	Whether or not the user completed the purchase

Target Variable

Fig-1: Detailed description of each feature

4. Exploratory Data Analysis

Missing Values: There are no missing values in the dataset.

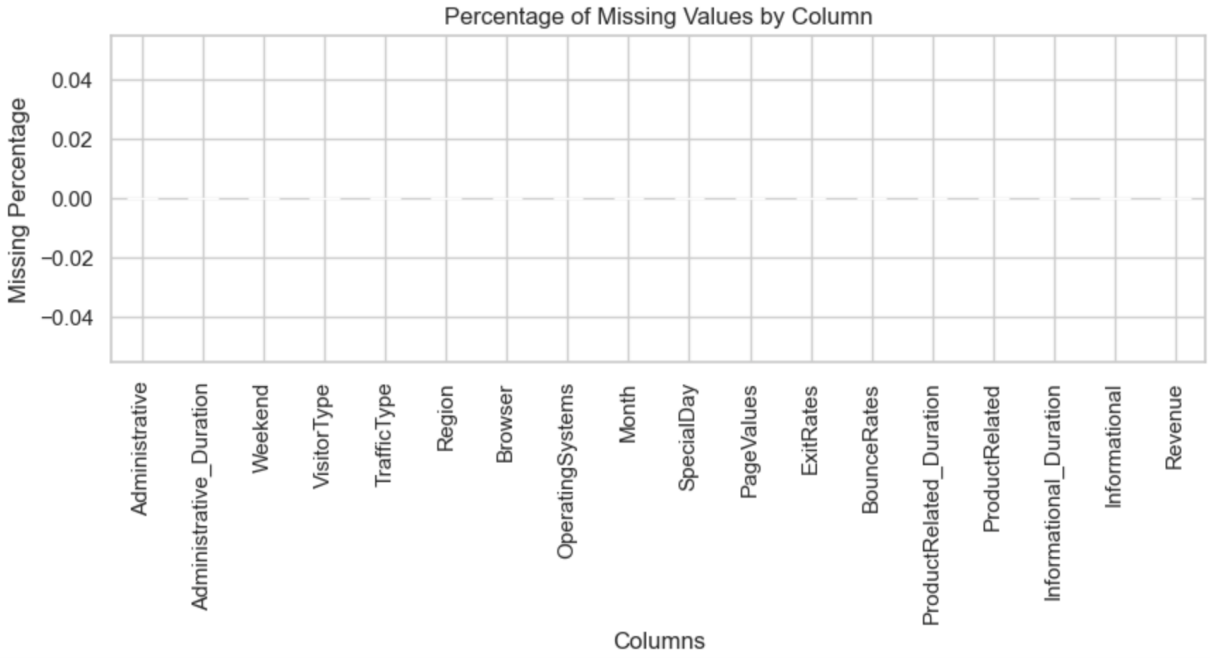

Fig-2: Percentage of Missing Values by Column

Missing Data in Months: Visualization revealed that data for the months of January, April, and June were missing in the dataset.

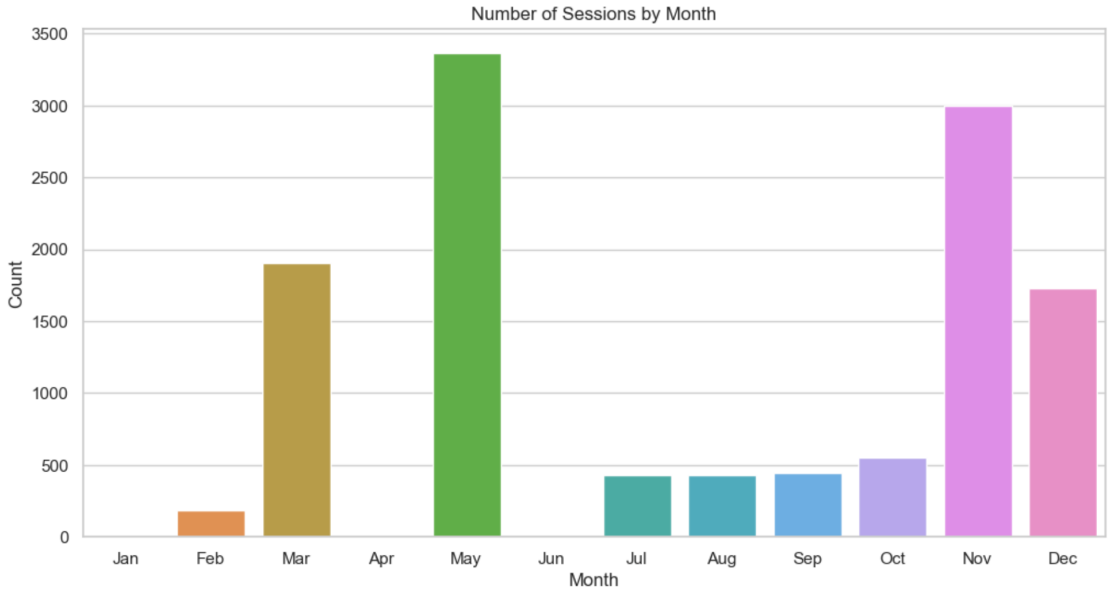

Fig-3: Missing data for Jan, Apr, Jun

Unique Values: The dataset contains the following unique values for each feature:

Feature Name	Unique Values
Administrative	27
Administrative_Duration	3335
Informational	17
Informational_Duration	1258
ProductRelated	311
ProductRelated_Duration	9551
BounceRates	1872
ExitRates	4777
PageValues	2704
SpecialDay	6
Month	10
OperatingSystems	8
Browser	13
Region	9
TrafficType	20
VisitorType	3
Weekend	2
Revenue	2

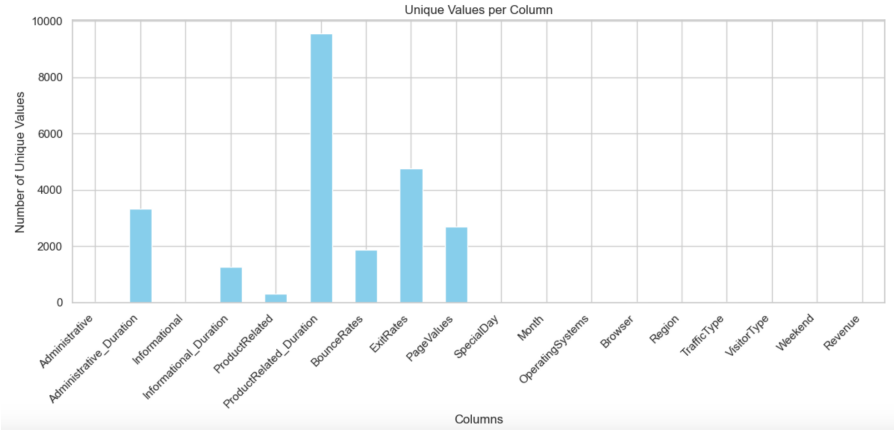

Fig-4: Unique values for each feature

Numerical Features Analysis: Visualization of numerical features including 'Administrative', 'Administrative_Duration', 'Informational', 'Informational_Duration', 'ProductRelated', 'ProductRelated_Duration', and 'PageValues' was conducted.

It was observed that 'Administrative_Duration', 'Informational_Duration', and 'ProductRelated_Duration' were on a very high scale compared to other numerical features.

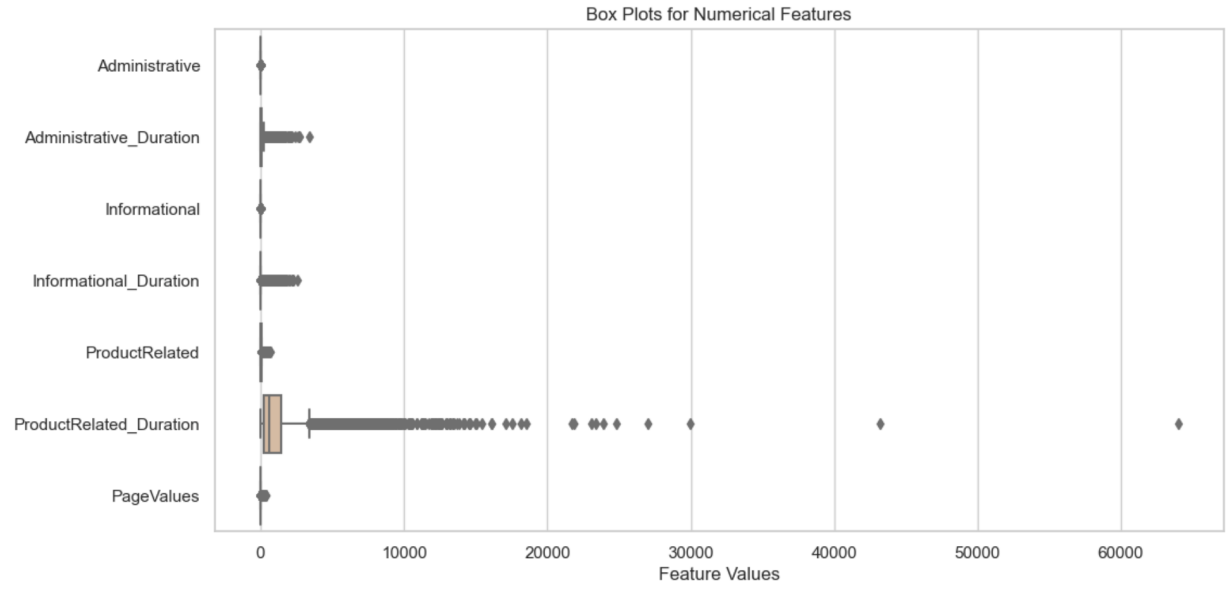

Fig-5: Box plot visualization for numerical features

Descriptive Statistics: Further, descriptive statistics for these numerical features was calculated. These statistics provide insights into the distribution of each numerical feature in the dataset.

	Administrative	Administrative_Duration	Informational	Informational_Duration	ProductRelated	ProductRelated_Duration	PageValues
count	12330.000000	12330.000000	12330.000000	12330.000000	12330.000000	12330.000000	12330.000000
mean	2.315166	80.818611	0.503569	34.472398	31.731468	1194.746220	5.889258
std	3.321784	176.779107	1.270156	140.749294	44.475503	1913.669288	18.568437
min	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
25%	0.000000	0.000000	0.000000	0.000000	7.000000	184.137500	0.000000
50%	1.000000	7.500000	0.000000	0.000000	18.000000	598.936905	0.000000
75%	4.000000	93.256250	0.000000	0.000000	38.000000	1464.157214	0.000000
max	27.000000	3398.750000	24.000000	2549.375000	705.000000	63973.522230	361.763742

Fig-6: Descriptive statistics for numerical features

Target Variable: 'Revenue'

This is a highly imbalanced feature that makes the entire dataset imbalanced (i.e., the number of positive values (purchase made by customer) is significantly smaller than the negative examples (no purchase made by customer))

Fig-7: Distribution of Target Variable

5. Data pre-processing and Feature Engineering

Why Data pre-processing and Feature Engineering:

Data pre-processing and feature engineering are crucial steps in the machine learning pipeline as they directly impact the quality and effectiveness of the models developed.

1. **Data Quality**: Pre-processing helps in ensuring that the data is clean and of high quality by handling missing values, outliers, and other anomalies. This improves the reliability and accuracy of the models.
2. **Feature Relevance**: Feature engineering involves selecting, creating, or transforming features to make them more informative for the models. This step helps in improving the model's ability to learn relevant patterns from the data.
3. **Model Performance**: Proper pre-processing and feature engineering can lead to better model performance. By normalizing data and encoding categorical variables correctly, models can learn more efficiently and make better predictions.
4. **Reduced Overfitting**: Feature engineering can help in reducing overfitting by selecting only the most relevant features and reducing the complexity of the model.
5. **Interpretability**: Pre-processing and feature engineering can also improve the interpretability of the models by creating features that are more easily understood by stakeholders.

Hence, these steps are essential for preparing the data in a way that maximizes the performance and interpretability of machine learning models.

Several steps were undertaken to prepare the dataset for modeling.

1. **Encoding Techniques:** These techniques help represent categorical variables in a way that is understandable by machine learning models
    - **Label Encoding**:
        - Certain categorical features, such as 'Weekend' and 'Revenue', were encoded into numerical labels using label encoding.
        - This conversion replaced the 'True' and 'False' values with '1' and '0', respectively.

- **One-Hot Encoding**: One-Hot Encoding is used when the categorical variable does not have an inherent order and each category is distinct. It creates a binary column for each category, indicating the presence (1) or absence (0) of that category in each observation.
    - Feature ‘VisitorType’ had three unique values ('Returning_Visitor', 'New_Visitor', 'Other') and there is no natural order and each category is distinct. This feature was one-hot encoded.
    - This conversion creates two new columns, 'VisitorType_Returning_Visitor' and 'VisitorType_Other', representing the different visitor types.
    - The rows that will have values as 0 for both these columns will be of type ‘New_Visitor’. This (k-1) column creation is to prevent introducing multicollinearity, where k = total unique values.

2. **Dropping 'Month' Feature**:
    - The 'Month' feature was dropped from the dataset due to missing data for the months of January, April and June.
    - Since the time-sensitivity information was already captured by the 'SpecialDay' column, 'Month' was seemed redundant for the analysis.
3. **Conversion of Duration Features**: This step helps in aligning the features with other time-related features in the dataset. Additionally, it prevents bias towards these features during model development and training, as the scale is now more consistent with the other features in the dataset.
    - The features 'Administrative_Duration', 'Informational_Duration', and 'ProductRelated_Duration' columns, originally in seconds, were converted to hours.
    - This conversion was done to facilitate better interpretation of the duration values and to align them with other time-related features in the dataset.
4. **Normalization**: Normalization was crucial to ensure that all features were on a similar scale, which is necessary for many machine learning algorithms to perform optimally. Techniques like MinMaxScaling is typically applied to scale numerical features to a fixed range (often between 0 and 1) to ensure that all features contribute equally to the model training process, especially when they are on different scales. Even though the duration features were converted to hours, they still have different ranges compared to other features in the dataset. MinMaxScaling, here, helps standardize these features along with the rest of the

dataset, ensuring that they do not dominate the model training process due to their larger values.

- The Min-Max Scaling technique (Python library *sklearn's* MinMaxScaler) was used to scale the numerical features down to a fixed range between 0 and 1.

6. Model Development, Optimization and Validation

Several steps were performed:

1. **Split dataset into Train and Test:** Dividing the dataset into training and test sets is an essential step in machine learning. The test set serves to evaluate the performance of the trained model on unseen data. This evaluation helps us assess how well the model generalizes to new, unseen examples.
   X_Train and X_Test contain the input features of the training and test sets, while Y_Train and Y_Test contain the corresponding target values. These datasets are used to train, validate, and test machine learning models.
   - The dataset was split into 80% Train and 20% Test.
   - Independent Variables (X_Train) of train set: Total 16,734 records and 17 features
   - Dependent Variable (Y_Train) of train set = Revenue
   - Independent Variables (X_Test) of test set: Total 2,466 records and 17 features
   - Dependent Variable (Y_Test) of test set = Revenue

2. **Resampling Target Variable:** Since the dataset is imbalanced, because of more values of no revenue (seen in *Fig-7*), it is crucial that this is rectified before model development. Imbalanced datasets can lead to biased models that perform poorly on the minority class (Revenue = 1).
   - Oversampling using SMOTE (Synthetic Minority Over-sampling Technique) was performed to resample the minority class (Revenue = 1) to have the same number of samples as the majority class (Revenue = 0).
   - This technique was performed only on the Train Set. This is to avoid data leakage, which can occur when information from the test set is inadvertently used to influence the training process.

Fig-8: Distribution of Revenue after oversampling

3. **<u>Model Selection</u>**: Considering this is a classification problem, 8 models were chosen to train the dataset. The selection of these models was based on their individual strengths and their potential to perform well in classifying customer purchase intent based on the dataset's characteristics and the project's objectives. The models developed include:

    1. Logistic Regression: Chosen for its simplicity and efficiency in binary classification tasks, especially when the relationship between features and the target variable is assumed to be linear.

    2. Decision Tree: Selected for its ability to handle both numerical and categorical data, as well as its interpretability, which allows for easy understanding of the model's decisions.

    3. Random Forest: Known for its robustness against overfitting and its capability to handle large datasets with high dimensionality, making it suitable for complex classification problems.

    4. XGBoost: A powerful ensemble method known for its high performance and speed, often used in competitions and real-world applications where accuracy is crucial.

5. Naïve Bayes: Despite its simplicity and the assumption of feature independence, Naïve Bayes can perform well in practice, particularly with small datasets and when the independence assumption holds.

6. Support Vector Machines (SVM): Effective in high-dimensional spaces, SVMs are suitable for cases where the number of dimensions exceeds the number of samples. They can also handle non-linear decision boundaries using kernel functions.

7. K-Nearest Neighbor (KNN): A non-parametric method suitable for multi-class classification, KNN is intuitive and easy to implement. It works well with small datasets but can be computationally expensive with large ones.

8. Dummy Classifier: Chosen as a baseline for comparison and evaluating the performance of other classifiers. This model provides a simple and intuitive approach that predicts the majority class label in the dataset without considering any features. By comparing the performance of more sophisticated classifiers with the Dummy Classifier, assessment of whether our models are learning meaningful patterns from the data or simply predicting the majority class can be made. This comparison helps in gauging the effectiveness of the models in classifying customer purchase intent and provides valuable insights into the relative performance and utility of each classifier.

4. **Model Fitting and Evaluation:** In the Model Fitting and Evaluation section, we trained and evaluated several classification models on the dataset using 5-fold cross-validation. 5-fold cross-validation is commonly used method for evaluating classification models, offering a good balance between computational efficiency, statistical power, and generalization performance. The evaluation metrics used include Accuracy, Precision, Recall, F1 Score, and ROC-AUC (Receiver Operating Characteristic - Area Under the Curve) Score.

7. Results Analysis and Interpretation

In this classification problem, which is predicting customer purchase intent, the ROC-AUC score is a better evaluative metric than accuracy, precision, recall, and F1 score.

This is because the dataset is imbalanced, with fewer instances of positive class (customers who made a purchase) compared to the negative class.

The ROC-AUC score considers the trade-off between true positive rate and false positive rate across all possible thresholds, providing a more comprehensive measure of model performance that is robust against imbalanced data.

Based on the evaluation results, the Random Forest model emerged as the best performer, having the highest average ROC-AUC score among all models evaluated. This indicates that the Random Forest model is effective in predicting customer purchase intent in our dataset.

Model	ROC-AUC
Logistic Regression	0.900779996
Decision Trees	0.885313229
Random Forest	0.981934931
XGBoost	0.980500356
Naive Bayes	0.848069963
SVM	0.914465662
KNN	0.940869949
Dummy Classifier	0.493187348

Fig-9: ROC-AUC score after 5 fold CV

Dummy classifier was used to compare results to see if the classifiers are producing better results than guessing (dummy). The 7 models seem to be much more accurate than guessing. The dummy classifier seems to be right about 49.31% of the time, which is expected, as it is making guesses based on the distribution of a stratified dataset.

CV	Accuracy	Precision	Recall	F1-Score	ROC-AUC
Logistic Regression	0.80966911	0.88043963	0.71686202	0.79013488	0.90078
Decision Trees	0.88616173	0.88332858	0.88644206	0.87987888	0.88531323
Random Forest	0.92954601	0.91539394	0.94082879	0.92834463	0.98193493
XGBoost	0.91341284	0.9375674	0.88966158	0.89714985	0.98050036
Naive Bayes	0.75385424	0.77045056	0.72379496	0.74618377	0.84806996
SVM	0.82185983	0.88799335	0.73694041	0.80523785	0.91446566
KNN	0.86578226	0.81383305	0.94884536	0.87607874	0.94086995

Fig-10: Other evaluative metrics score after 5-fold CV

Feature Importance: Feature importance after evaluating a machine learning model is used to understand the relative influence or contribution of each input feature on the model's predictions or outputs. According to Random Forest, feature 'PageValues' was the most important feature followed by ProductRelated_Duration_hours.

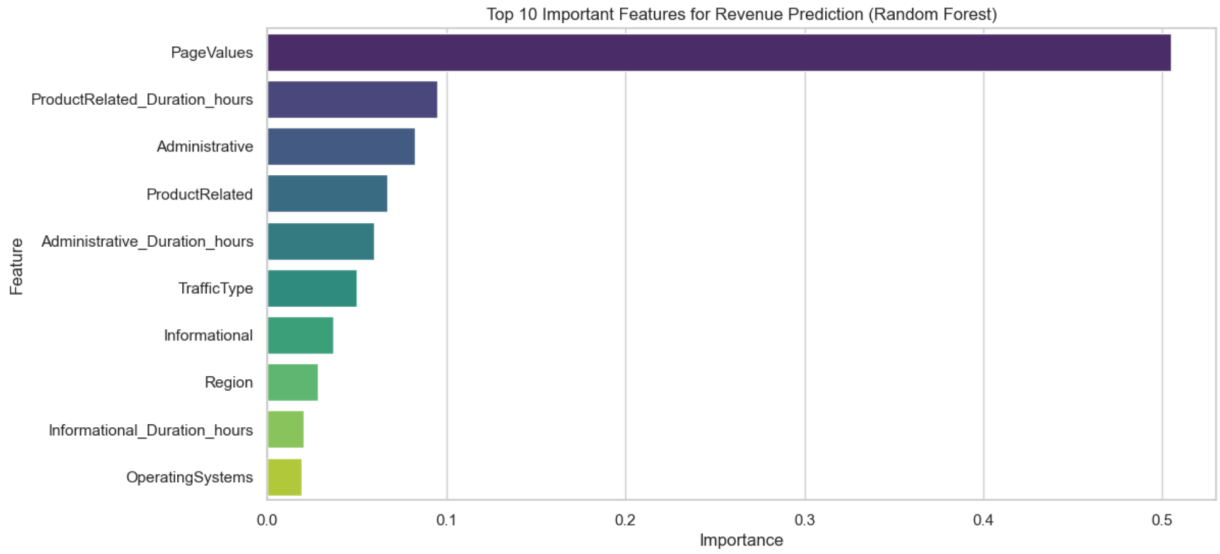

Fig-11: Top 10 Important Features for Revenue Prediction by Random Forest
"Page Values" seems to be the most impactful feature by a large margin

8. Conclusion

Random Forest model accurately predicts online purchase intent, achieving an exceptional ROC-AUC score of 0.982 - outperforming other models such as Logistic Regression, Decision Trees, XGBoost, Naive Bayes, SVM, and KNN.

By leveraging customer browsing behavior, especially pageview patterns (top feature) and time spent on product pages, Random Forest model pinpoints promising sales opportunities.

Future Work: Continuous monitoring, dimensionality reduction, feature engineering, Deep Learning models can potentially result in further improvement in revenue prediction.

9. Citation/References

Acknowledging various sources that contributed to the development of this project, including:

1. **Machine Learning lecture slides** by Professor Houping uploaded on iCollege to understand different classification models and advantages/disadvantages of using these models.
2. **Dataset sourced from University of California, Irvine Machine Learning Repository:** https://archive.ics.uci.edu/dataset/468/online+shoppers+purchasing+intention+dataset
3. **Kaggle**: Approach taken to solve classification problems by various Kaggle users for learning purpose.
4. **Large Language Models:** ChatGPT and Claude to understand concepts better.