IST 718: Big Data Analytics

ANALYZING ONLINE SHOPPER BEHAVIOR USING PYSPARK

Final Project Report

Group 2 Members:

VAIBHAV CHAUDHARI
SHUBH MODY
JAINEEL PRAVIN PARMAR
RITESH VERMA

Project Overview

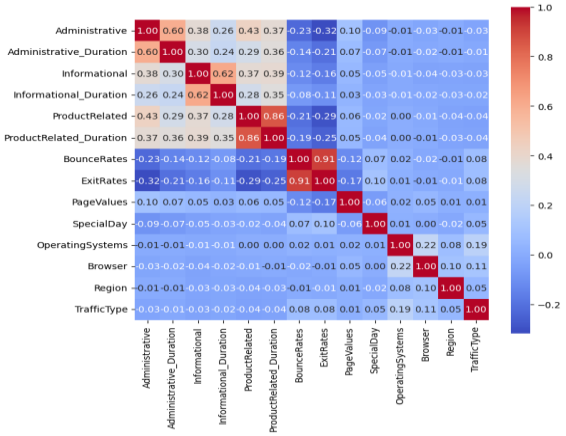
- **Objective:** This project is dedicated to exploring and analyzing the "Online Shoppers Intention" dataset to gain a deeper understanding of online shopper behavior and the effectiveness of an e-commerce website. The primary goal is to utilize PySpark for data analysis, creating insightful visualizations, and deriving key insights that can contribute to enhancing the online shopping experience.
- **Dataset Description:** The core of our analysis is the "Online Shoppers Intention" dataset, accessed from Kaggle. This comprehensive dataset comprises information about various aspects of online shoppers' interactions with an e-commerce website, ranging from basic visitor details to their engagement with the website, and ultimately, their purchasing decisions. The dataset encompasses 12,331 rows and 18 columns, offering a rich ground for extensive data analysis.
- **Scope of Analysis:**
  1. **Exploratory Data Analysis:** Conducting a thorough exploration of the dataset to understand the distribution and relationships among various features.
  2. **Visualization:** Creating intuitive visualizations to represent complex data in an easily understandable format.
  3. **Insight Generation:** Identifying key patterns and factors that influence website efficiency, user engagement, and sales.
  4. **Actionable Recommendations:** Providing strategic recommendations to optimize e-commerce operations, which could lead to enhanced marketing strategies and increased revenue.
- **Expected Outcomes:** A comprehensive understanding of factors affecting online shopper behavior. Identification of key areas for website performance improvement. Strategic insights to aid in decision-making for marketing and sales enhancement. The findings from this project are expected to offer valuable insights into online consumer behavior, assisting in making data-driven decisions for e-commerce site optimization. This, in turn, could lead to improved customer satisfaction, increased engagement, and higher sales conversions.

Prediction, Inference, and Other Goals

- **Project Aim -** The core aim of this project is to develop a predictive model that accurately determines the likelihood of a session on a website resulting in revenue. This task falls under the scope of binary classification, where the outcome is either 'revenue' or 'no revenue'. The model's ability to predict this outcome is critical for optimizing marketing strategies, tailoring user experience, and improving the overall conversion rate.
- **Model Selection -** To achieve the project aim, we explored several machine learning algorithms renowned for their efficacy in binary classification tasks. The models were selected based on their ability to handle large datasets, ease of interpretation, and performance in preliminary tests. The chosen models include logistic regression, decision trees, random forests, and gradient-boosting machines.
- **Inference Objectives -** Our inference process is designed to extract meaningful insights from the model's predictions. It involves analyzing the model's performance metrics, such as accuracy, precision, recall, and the F1 score, to understand the trade-offs between identifying all potential revenue-generating sessions and the cost of false positives. Additionally, we employ techniques like feature importance analysis to discern which factors most significantly influence the likelihood of revenue generation.
- **Interpretation of Model Performance -** The performance metrics provide a quantitative basis for interpreting the model's effectiveness. For instance: Accuracy reveals the overall correctness of the model's predictions. Precision measures the model's ability to identify only the true instances of revenue-generating sessions. Recall assesses the model's capacity to find all potential revenue-generating sessions within the data. The F1 score offers a balance between precision and recall, providing a harmonic mean of the two metrics. By evaluating these metrics, we can determine the model that best aligns with our business objectives.
- **Further Goals -** Beyond prediction and inference, the project also aims to: Provide actionable insights for the business to enhance user engagement and conversion rates. Identify key user behaviors that correlate with revenue generation to inform content creation and service offerings. Develop a sustainable, iterative model that can adapt to changes in user behavior patterns over time. In conclusion, the selected model will be integral in driving data-driven decisions to foster growth and increase profitability for the website.

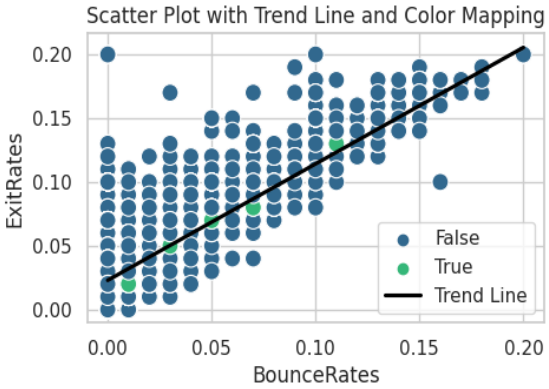
Data Exploration

1) Correlation Heatmap Analysis:



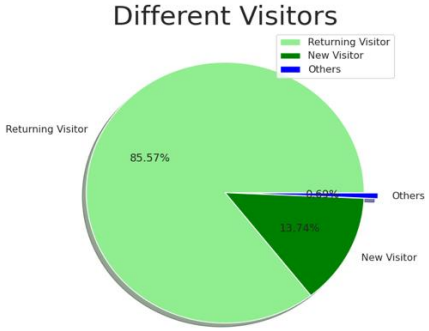
This heatmap provides a correlation matrix of various website metrics, offering insights into the relationships between different user engagement indicators. High positive correlations (closer to 1.0) indicate a direct relationship, whereas high negative correlations (closer to -1.0) suggest an inverse relationship. For example, 'Administrative' and 'Administrative_Duration' have a strong positive correlation, suggesting that the more administrative pages a user visits, the longer they spend on the site. Similarly, 'BounceRates' and 'ExitRates' have a high positive correlation, indicating that sessions with a single page view often result in the user leaving the site immediately.

2) Scatter plot Bounce Rate vs Exit Rate:



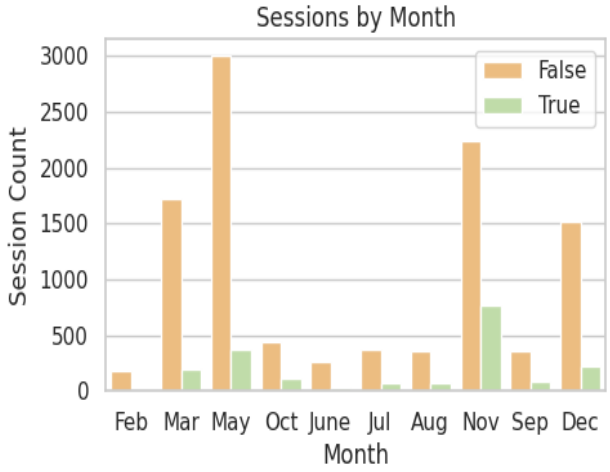
This scatter plot displays the relationship between 'BounceRates' and 'ExitRates', with a trend line indicating a positive correlation. The color mapping differentiates between 'True' and 'False' transactions, suggesting that sessions aligned along the trend line are more likely to result in transactions, hence a predictable relationship between engagement metrics and transaction likelihood.

3) Visitor Types:



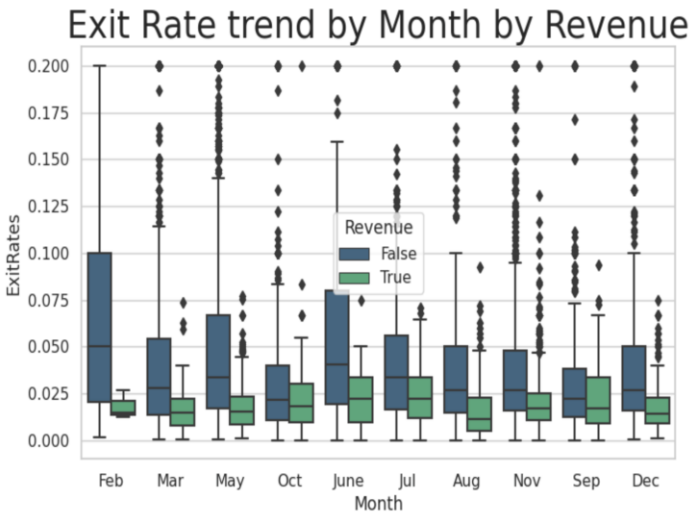
The pie chart breaks down the types of visitors to the website, showing a large majority of 'Returning Visitors', a smaller percentage of 'New Visitors', and a very small segment labeled 'Others'. This indicates a strong base of repeat visitors, which could be indicative of user loyalty or effective retention strategies.

4) Bar Chart of Sessions by Month:



The bar chart compares the number of sessions labeled as 'True' or 'False' across different months. Variations in session counts suggest seasonal trends or the impact of marketing campaigns and promotions. For instance, May and November show the highest number of sessions, while June and September have the lowest.

5) Box Plot for Exit Rate Trends by Month by Revenue:



This box plot details the exit rates by month, categorized by revenue impact ('True' for revenue-impacting sessions and 'False' for non-revenue). The median exit rates and their distribution across different months are visualized, with outliers indicated by diamonds. Higher medians in specific months suggest temporal factors that influence user exit behavior, which could be valuable for targeting improvements in user retention strategies.

Feature Engineering

For feature processing, we prepare the data as required and suitable for modelling in Spark environment. Initially, we convert boolean columns to integers to facilitate analysis. Specifically, we cast the 'Revenue' and 'Weekend' columns from boolean to integer data types.

The core of our feature engineering involves a combination of string indexing, vector assembling, and standard scaling. First, we identify categorical columns, 'Month' and 'VisitorType', which need indexing and encoding. Using Spark's StringIndexer, we convert these categorical strings into numerical indices.

Next, we assemble our features using VectorAssembler. This step integrates both the newly indexed categorical columns and a variety of numeric columns, such as 'Administrative', 'ProductRelated_Duration', 'BounceRates', among others, into a single vector column. This vectorized format is a prerequisite for most machine learning algorithms in Spark.

Finally, we apply StandardScaler to this assembled feature vector. The scaling process standardizes the feature vector, ensuring that each feature contributes equally to the analysis by negating the influence of differing scales and units.

These preprocessing steps are encapsulated within a Spark Pipeline, ensuring a streamlined and efficient transformation of our data. After fitting this pipeline to our 'Shopper' dataset, we obtain a transformed dataframe, Shopper_transformed, where each record now contains a 'scaledFeatures' vector suitable for both supervised and unsupervised machine learning models. The transformed dataframe is then displayed, showcasing the first five processed records, demonstrating the successful application of our feature processing strategy.

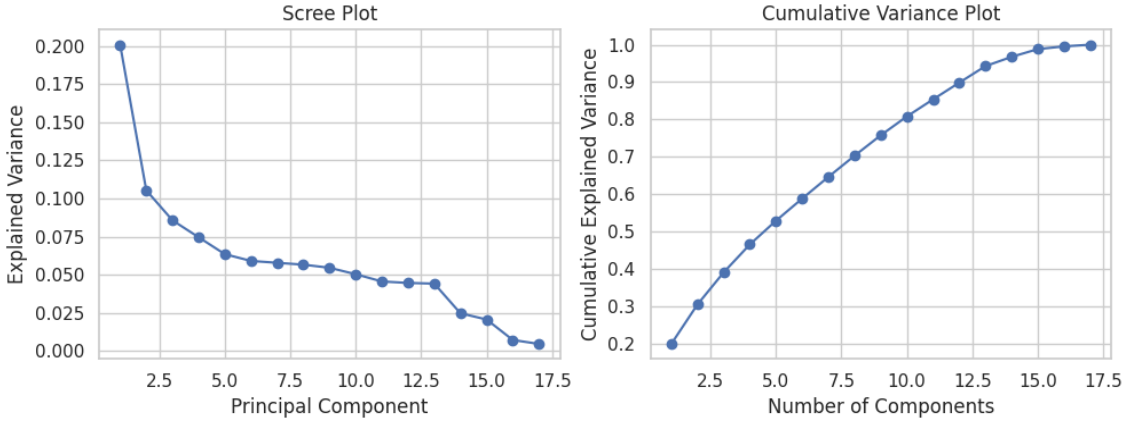
```
+--------------------+-------+
|      scaledFeatures|Revenue|
+--------------------+-------+
|(17,[0,6,8,9,12,1...|      0|
|(17,[0,6,7,9,12,1...|      0|
|(17,[0,6,8,9,12,1...|      0|
|(17,[0,6,7,8,9,12...|      0|
|(17,[0,6,7,8,9,12...|      0|
+--------------------+-------+
only showing top 5 rows
```

```
+-------+--------+
|Revenue|count(1)|
+-------+--------+
|      1|    1908|
|      0|   10422|
+-------+--------+
```

The image above illustrates the transformation of raw data into a format suitable for machine learning models.

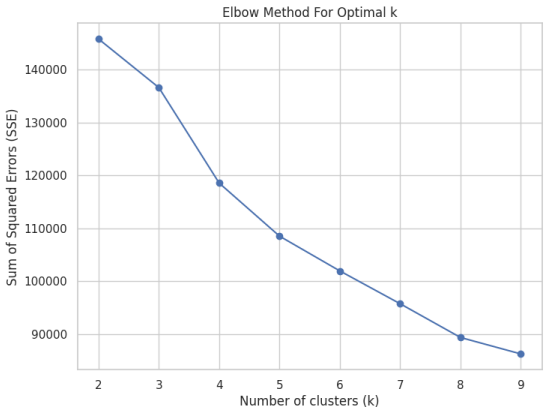
Prepared Dataset for Modeling

- The DataFrame shown is ready for predictive modeling, with the top 5 rows visible as an example. The 'scaledFeatures' column contains the standardized feature vectors, and the 'Revenue' column serves as the binary target variable. The standardization process mentioned ensures that each feature contributes equally to the model's prediction capability.
- Class Imbalance in Target Variable: The class distribution of the target variable 'Revenue' is significantly imbalanced, with 1,908 instances of the positive class ('Revenue' = 1) and 10,422 of the negative class ('Revenue' = 0). This imbalance can influence model training and may need to be addressed to prevent bias towards the majority class.



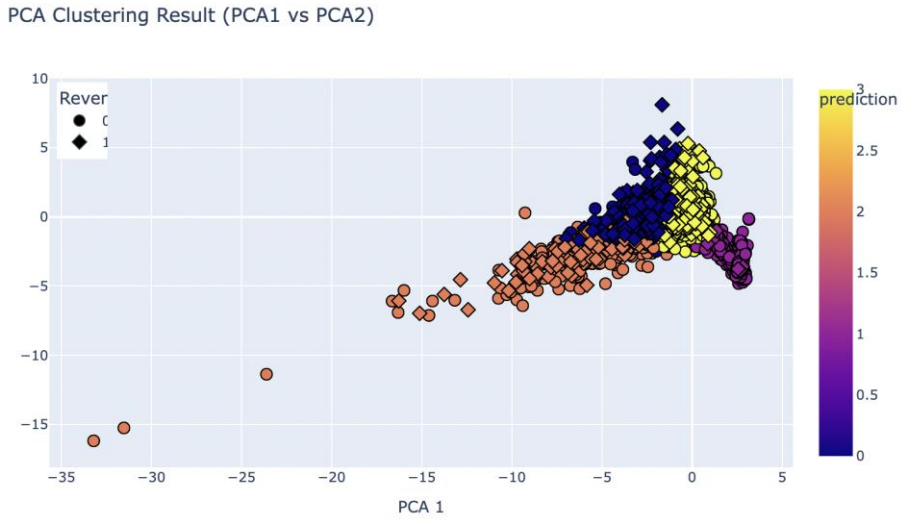
- **Scree Plot:** The Scree Plot displays the explained variance by each principal component. It's evident from the plot that the first few components account for the most variance within the dataset. The typical 'elbow' method used to select the number of components suggests that after the third or fourth component, the additional explained variance from each subsequent component drops off, indicating that the majority of the useful information is captured within the first few components.
- **Cumulative Variance Plot:** The Cumulative Variance Plot illustrates the total variance explained as more components are added. This plot shows that around 10 components explain approximately 80% of the variance, and about 15 components are required to explain over 90% of the variance. This can guide the decision on how many principal components to retain for further analysis or data modeling, balancing between dimensionality reduction and information retention.

Elbow Point Identification



- The plot demonstrates a consistent decline in the Sum of Squared Errors (SSE) as we increase the number of clusters, k. The optimal number of clusters, or the 'elbow' of the graph, is identified where the rate of decrease in SSE sharply changes. In this case, the elbow seems to occur at k=4, where the graph begins to level off, indicating a natural division in the dataset.
- **Trade-off Between Cluster Count and SSE:** The graph presents a balance between the number of clusters and the SSE. Adding more clusters results in lower SSE, signifying a finer fit to the data. However, post the elbow point at k=4, the reduction in SSE slows down significantly, suggesting that additional clusters are less justified as they do not contribute as much to the decrease in error.
- **Optimal Clusters for Analysis:** With the elbow method indicating a plateau beginning at k=4, selecting four clusters for the K-means analysis emerges as the most judicious choice for this dataset. Beyond this point, the marginal gain in explained variance is outweighed by the cost of increased model complexity and potential overfitting.

Cluster Distribution



The scatter plot is color-coded by the clustering results, with different colors representing different clusters. Cluster 2 is highlighted for having the highest percentage of sessions (30.18%) that resulted in a complete transaction, which is significant for the revenue of the website.

- **Revenue Correlation with Clusters:** The plot uses shapes to differentiate sessions that resulted in revenue (diamonds) from those that did not (circles). The concentration of diamonds within Cluster 2 suggests that this cluster has a higher correlation with successful revenue-generating sessions.
- **Cluster Prediction Percentage:** A legend or side table in the plot shows the percentage of sessions associated with each cluster. Cluster 2 is dominant for revenue generation, whereas Cluster 0 is second best with 27.54% of sessions resulting in revenue. Cluster 1 has a very low percentage (0.57%). Cluster 3 accounts for 12.8% of the sessions, indicating the third-best grouping of potential interest for revenue conversion analysis.

```
prediction    1.000000
Revenue      -0.118071
pca1          0.323799
pca2          0.304821
pca3         -0.131540
pca4         -0.177726
pca5         -0.024476
pca6         -0.410555
pca7         -0.132266
pca8         -0.008864
pca9          0.107601
pca10        -0.090507
Name: prediction, dtype: float64
```

```
Administrative_Duration: 0.6188100114180154
Informational: -0.29692375104592966
Informational_Duration: 0.18969176276193903
```

```
VisitorTypeIndex: 0.36325540854576427
Informational: 0.27046404487323356
Administrative: 0.25606405975918245
```

- **PCA and Cluster Correlation:** The correlation data underscores that PCA1 exhibits a substantial positive correlation with the clustering predictions at approximately 0.323, confirming its significant role in differentiating customer segments. PCA2, while slightly less influential than PCA1, still maintains a notable positive correlation at around 0.304, suggesting its relevance in the clustering process.
- **Influence of Original Features on PCA1:** For PCA1, 'Administrative_Duration' shows a strong positive loading of about 0.618, indicating its dominant role in this principal component, which could reflect the importance of time spent on administrative tasks in shaping customer behavior profiles. Meanwhile, 'Informational' is negatively correlated with PCA1, as indicated by a loading of approximately -0.297, suggesting a divergent influence compared to 'Administrative_Duration'. Furthermore, 'Informational_Duration' has a positive loading of about 0.189, solidifying the impact of the duration on informational pages on the composition of PCA1.
- **Impact on PCA2:** With respect to PCA2, 'VisitorTypeIndex' has a positive loading of about 0.363, signifying its strong effect on this component, while 'Informational' demonstrates a positive loading as well, at approximately 0.270. 'Administrative' also shows a notable positive loading on PCA2, with a value of around 0.257. These features, therefore, are integral in influencing the distribution of data points along the PCA2 axis and play a critical role in our segmentation analysis.

Results

In this study, we constructed a model pipeline to evaluate the performance of three distinct machine learning models: Logistic Regression, Gradient-Boosted Trees, and Random Forest. The models were trained using a predefined dataset and their predictive performances were compared using several metrics: Area Under the Curve (AUC), Receiver Operating Characteristic (ROC) curve, recall, F1 score, and precision.

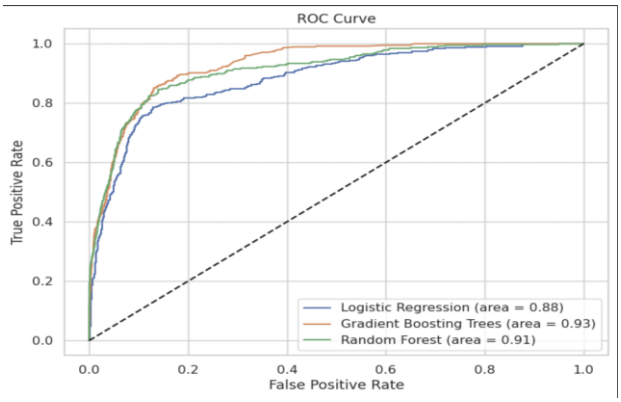
Model Accuracy

Model	Accuracy
Logistic Regression AUC	0.8810661284516278
Gradient-Boosted Trees AUC	0.9281369849127378
Random Forest AUC	0.9105187164454933

The accuracy of the models was assessed using the AUC metric. The AUC results were as follows:

- Logistic Regression achieved an AUC of 0.88.
- Gradient-Boosted Trees achieved a higher AUC of 0.93, indicating a superior performance in distinguishing between the positive and negative classes.
- Random Forest obtained an AUC of 0.91, performing slightly less effectively than Gradient-Boosted Trees but better than Logistic Regression.

ROC Curve Analysis



The ROC curve visualizations provided a graphical representation of the true positive rate against the false positive rate at various threshold settings. The Gradient-Boosted Trees model demonstrated the closest approach

to the top-left corner, indicative of its higher accuracy, followed closely by the Random Forest model. The Logistic Regression model's curve was lower, reflecting its relatively lower true positive rate.

Recall

Model	Recall
Logistic Regression	0.3660477453580902
Gradient-Boosted Trees	0.5915119363395226
Random Forest	0.4960212201591512

These results suggest that the Gradient-Boosted Trees model was the most effective at identifying positive cases, while Logistic Regression was the least effective.

F1 Score

Model	F1 Score
Logistic Regression	0.8631336790650214
Gradient-Boosted Trees	0.8937451499621274
Random Forest	0.8856288218104544

The F1 scores indicate that both Gradient-Boosted Trees and Random Forest models provided a harmonious balance between precision and recall, outperforming Logistic Regression.

Precision

Model	Precision
Logistic Regression	0.7340425531914894
Gradient-Boosted Trees	0.7034700315457413
Random Forest	0.7420634920634921

Random Forest demonstrated slightly higher precision, indicating a lower rate of false positives.

Overall, the Gradient-Boosted Trees model exhibited superior performance across most metrics, particularly in AUC and recall, suggesting it as the most robust model among the three evaluated. The Random Forest model also showed competitive performance, particularly in terms of precision and F1 score. In contrast, the Logistic Regression model consistently showed lower performance metrics. These results highlight the effectiveness of ensemble methods in handling complex datasets and predictive modeling tasks.

Results Summary:

We evaluated three machine learning models—Logistic Regression, Gradient-Boosted Trees, and Random Forest—across multiple performance metrics. The Gradient-Boosted Trees model outperformed the others with the highest AUC of 0.93 and the highest recall rate of 0.59, indicating its superior capability in classifying the positive class correctly. The Random Forest model showed a strong balance with a high precision rate of 0.74 and a competitive AUC of 0.91. Logistic Regression, while exhibiting the lowest AUC at 0.88 and recall at 0.36, maintained a competitive F1 score of 0.86. Overall, Gradient-Boosted Trees stood out as the most robust model for our specific dataset, with Random Forest as a close contender, especially in precision-driven contexts. Logistic Regression lagged slightly behind but was still a viable model based on the F1 score.

Problem Encountered

- Addressing imbalanced class distribution in 'Revenue' for model accuracy.
- Deciding on the appropriate cluster count in K-means to prevent overfitting.
- Identifying key features impacting PCA1 and PCA2 for optimal clustering.
- Utilizing multiple machine learning models, assessing varying success rates.
- Overcoming challenges in model training due to skewed data.

Summary of how well you achieved your prediction and inference goals

- The features that most influence PCA1 and PCA2, as shown by their loadings, are VisitorTypeIndex, Administrative, Administrative_duration, and Informational.
- Since PCA1 and PCA2 is highly correlated with the clusters, these features are likely pivotal in determining the cluster with the highest revenue conversion. Hence, focusing on strategies that leverage these aspects of customer engagement could be key to driving higher revenue.
- GBT leads in AUC, a critical metric for imbalanced classes, which indicates a strong predictive ability.
- RF has the highest F1 score, suggesting a balanced trade-off between precision and recall.
- LR, while having the lowest recall, maintains strong performance in AUC and F1, indicating good model performance despite its simplicity.

Citations

1. Sakar, C.O., Polat, S.O., Katircioglu, M. et al. Neural Comput & Applic (2018) https://link.springer.com/article/10.1007/s00521-018-3523-0

2. Sue, Henry. "Online Shoppers Intention." Kaggle, (2019) www.kaggle.com/datasets/henrysue/online-shoppers-intention/

3. "Machine Learning with PySpark's pyspark.ml Library." Medium, (Nov 2019) https://medium.com/@think-data/machine-learning-with-pysparks-pyspark-ml-library-c9ab66096700#:~:text=The%20PySpark.ml%20library%20offers,1

4. "PySpark Logistic Regression." Machine Learning Plus, (2023) www.machinelearningplus.com/pyspark/pyspark-logistic-regression/

5. "Gradient Boosted Tree Classifier Model Using PySpark." Medium, (Mar 10, 2022) https://medium.com/featurepreneur/gradient-boosted-tree-classifier-model-using-pyspark-73281ff10109#:~:text=Schema%20basically%20gives%20details%20about,holding%20null%20values%20or%20not.&text=Here%2C%20we%27ll%20also%20drop,t%20contribute%20to%20the%20p

6. "PySpark Random Forest." Machine Learning Plus, (2023) www.machinelearningplus.com/pyspark/pyspark-random-forest/