BAN5600-02-F24 Advanced Big Data Comp & Prog Final Project - Carol Chu

1. Project Introduction

This dataset provides insights into user behavior and online advertising, specifically focusing on predicting whether a user will click on an online advertisement. It contains user demographic information, browsing habits, and details related to the ad's display. After careful checkup, I chose this dataset for building binary classification models to predict user interactions with online ads and, more importantly, using Spark to filter out the conditions that could lead to the best ad results.

2. Dataset Source and Dictionary

- Data Source: Kaggle - 📣 Ad Click Prediction Dataset
- Data Scope: 10,000 entries, 9 columns (2 identifier columns, 6 independent columns, 1 dependent column)
- Data Dictionary:
  1. id: Unique identifier for each user.
  2. full_name: User's name formatted as "UserX" for anonymity.
  3. age: Age of the user (ranging from 18 to 64 years).
  4. gender: The gender of the user (categorized as Male, Female, or Non-Binary).
  5. device_type: The type of device used by the user when viewing the ad (Mobile, Desktop, Tablet).
  6. ad_position: The position of the ad on the webpage (Top, Side, Bottom).
  7. browsing_history: The user's browsing activity prior to seeing the ad (Shopping, News, Entertainment, Education, Social media).
  8. time_of_day: The time when the user viewed the ad (Morning, Afternoon, Evening, Night).
  9. click: The target label indicating whether the user clicked on the ad (1 for a click, 0 for no click).

3. Project Goal

To predict whether a user will click on an online ad based on their demographics, browsing behavior, the context of the ad's display, and the time of day. The individual project results can be used to improve ad targeting strategies, optimize ad placement, and better understand user interaction with online advertisements.

4. Data Cleaning Process

Among all 10,000 entries, there are around 48% of age and browsing history data missing, and around 20% of device type, ad position, and time of day data missing. Finally, I've decided to remove the 20% of rows where the ad_position value is missed. This is crucial because ad_position directly influences ad pricing and having accurate data for this variable is key to anyone who needs this result as a future advertising strategy, especially my main objective is to identify the most cost-effective ad strategy that generates the highest number of clicks for a general consumer market. Therefore, maintaining reliable data on ad positioning is essential. For the other missing value: The numeric column (age)- I imputed missing values with the average of each column to preserve as much data as possible. Binary or categorical columns (gender, device type, browsing history, time of day): I randomly imputed data based on the original data distribution percentage to fill in and ensure the overall balance of categories is maintained. This approach allows me to focus on the most important factors for the analysis while handling missing data in a way that minimizes bias and preserves data quality.

5. EDA

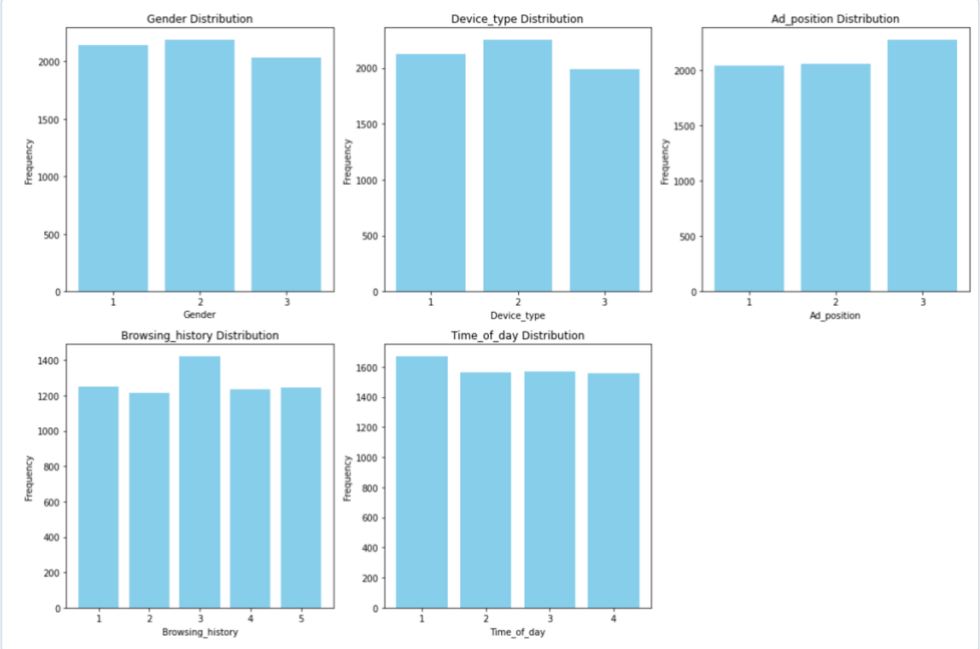
a) Descriptive Analysis

```
from pyspark.sql import functions as F

#mode value
def calculate_mode(df, col_name):
    mode_df = df.groupBy(col_name).count().orderBy(F.desc("count"))
    mode_value = mode_df.first()  # Get the first row, which is the mode
    return mode_value[col_name]
# List of columns for which to calculate the mode
columns_to_calculate = ["age", "gender", "device_type", "ad_position", "browsing_history", "time_of_day", "click"]

# Dictionary to store the mode results
mode_results = {}

for col in columns_to_calculate:
    mode_results[col] = calculate_mode(train_data, col)

# Display the results
print(mode_results)
```

```
{'age': 40, 'gender': 2, 'device_type': 2, 'ad_position': 3, 'browsing_history': 3, 'time_of_day': 1, 'click': 1}
```

After data cleaning, we can tell that the mean value is age of 40, female audience (1.9≈2), who use Desktop (1.9≈2), enjoy browsing entertainment page(3.01≈3), position of ad pop up on the side of website they are browsing, and they view ad in afternoon time(2.48≈2), but tend to not click the ads at this moment. However, there's a bit of difference in the mode descriptive data, which is the most frequently appeared profile in the dataset, and the result here shows our basic demographic are the same in gender, age, and device_type, but the ad pops up more in the bottom of the webpage, our common audience seems to view ad in the morning and prefer to have a click more!

b) Visualized Analysis



Through the visualized data, I first went through all binary and categorical columns.
We can tell the distribution over every categorical variable. Distribution in gender, device, ad position, browsing history, and time of day seems to be closer to equal distribution and normal distribution because I impute null value with train dataset distribution of these few columns, so it's more likely to similarly show the distribution of the non-null value. These plots help reveal how

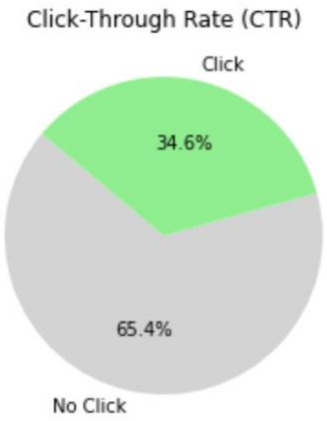
these features interact with the ad click behavior. These plots reveal that the majority of users are in the afternoon when interacting with ads, prefer to use a desktop, and often browse entertainment content. These insights may help refine the targeted ad strategies.
Next part, I went through the ad_click column, also the only column has no missing values in this dataset, indicating that this represents a complete and accurate distribution of call-to-action interactions. However, based on industry benchmarks released by Facebook and Google, typical click-through rates (CTR) are significantly lower than what the dataset suggests. This discrepancy indicates that the calculated ad click rate might pertain to specific platforms like Twitter or TikTok, rather than more general platforms such as Google and Facebook.

For context, average CTRs vary by platform and ad type:

Next part, I went through ad_click column, also the only column has no missing values in this dataset, indicating that this represents a complete and accurate distribution of call-to-action interactions. However, based on industry benchmarks released by Facebook and Google, typical click-through rates (CTR) are significantly lower than what the dataset suggests. This discrepancy indicates that the calculated ad click rate might pertain to specific platforms like Twitter or TikTok, rather than more general platforms such as Google and Facebook.

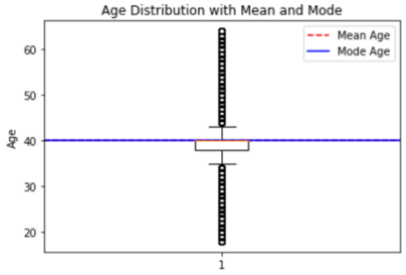
For context, average CTRs vary by platform and ad type:

- **Search ads:** 1.91%
- **Display ads:** 0.35%
- **Facebook ads:** 1.11%
- **Instagram ads:** 0.22%
- **LinkedIn ads:** 0.22%
- **Twitter ads:** 0.86%
- **Social feeds:** 1.1–1.3%



Given these figures, the dataset's high CTR of 34.6%
likely reflects either a specific platform, a unique channel, or specialized ad content.

Finally, I went through age data (numeric column) exploration. The mode age in this dataset is 40, however the average age is 26. But through the box plot spots out of IQR, we can tell the reach rate of this market research through the sample popularity rate generally reach every age level.



6. Model Training

i) Feature Encoding:

Categorical variables (gender, device_type, ad_position, etc.) were converted into numeric indices using PySpark's StringIndexer and further encoded into binary vectors with OneHotEncoder. For example, categorical columns will be code as column_option1, column_option2, column_option3 as below:

device_type (3 categories) → [device_type_1, device_type_2, device_type_3]
ad_position (3 categories) → [ad_position_1, ad_position_2, ad_position_3]
time_of_day (5 categories) → [time_of_day_1, ..., time_of_day_5]

ii) Models:

I trained three classification models to predict ad clicks:

(1) Decision Tree Classifier:

A simple yet interpretable model, the Decision Tree (DT) is known for its simplicity and interpretability with an AUC (Area Under the ROC Curve) of 0.548 here. This modest performance suggests that DT might be insufficient for capturing complex patterns in the dataset such as below problem:

- **Overfitting Risk:** DT models tend to overfit training data, especially when the dataset has high variance. This overfitting can lead to poor generalization, which explains the low AUC score when applied to unseen data.
- **Lack of Depth:** DT splits data based on simple threshold rules, making it less capable of capturing intricate relationships or interactions between features. For instance, complex patterns in user behavior, such as the combined effect of ad_position and browsing_history, might be missed.
- **Feature Importance:** DT's lower performance suggests that relying solely on hierarchical splits may overlook nuanced patterns in categorical variables, like time-of-day browsing habits, which might influence ad clicks.

While DT provides quick and interpretable insights, it may serve better as a baseline model or in conjunction with other methods through ensemble techniques.

(2) Gradient Boosted Tree Classifier:

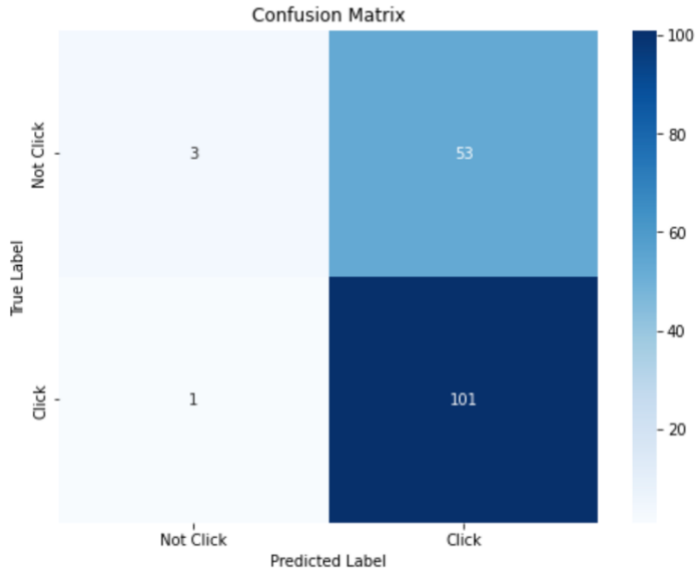
The Gradient Boosted Tree (GBT) emerged as the top performer with an AUC of 0.736 over models. GBT's iterative boosting process allows it to learn from previous mistakes, making it more accurate and effective in identifying click patterns compared to DT. This result indicates GBT's suitability for datasets with complex relationships and varied feature importance as below:

- **Iterative Learning:** Unlike DT, GBT builds models sequentially, with each iteration correcting errors made by the previous one. This iterative process allows GBT to handle complex, non-linear relationships between features. For example, the interaction between device_type and time_of_day may have subtle patterns that GBT captures effectively.
- **Feature Sensitivity**: GBT assigns different weights to data points, making it adept at focusing on hard-to-predict instances. This ability is particularly valuable in datasets where class distributions are imbalanced or where important features are masked by noise.
- **Regularization Benefits**: GBT incorporates regularization, reducing the risk of overfitting compared to DT. This feature ensures that the model remains robust, even with a diverse set of categorical inputs.

The superior performance of GBT suggests its suitability for applications where understanding nuanced user interactions is crucial, such as personalized ad targeting or multi-channel marketing analysis.

(3) Random Forest Classifier:
The Random Forest (RF) model achieved an AUC of 0.643. While better than the Decision Tree, it fell short of GBT's performance. RF also offered valuable insights through metrics:
Precision: 0.6558 — Indicates that approximately 66% of predicted clicks were correct.
Recall: 0.9902 — Demonstrates the model's ability to capture almost all true positive clicks.
F1 Score: 0.7891 — A balance between precision and recall, suggesting decent overall predictive performance.

```
Precision: 0.6558441558441559
Recall: 0.9901960784313726
F1-Score: 0.7890625000000001
```



Among all models, I focused on **Random Forest**, which aggregates multiple decision trees to improve the model's accuracy and generalizability. Before feeding the data into the Random Forest model, I first assembled all the features into a single vector column using VectorAssembler. I set the label as click (the target variable) and the features as the assembled vector column. After training the model, I evaluated its performance using the Binary Classification Evaluator, which measures the AUC (Area Under Curve) and accuracy of the model.

With an AUC of 0.643, the Random Forest (RF) classifier strikes a balance between interpretability and performance. It outperforms the DT but doesn't quite match GBT's accuracy. The reason might suggest as below:

- **Ensemble Strengths**: RF combines multiple decision trees, averaging their outputs to improve predictive performance. This ensemble approach helps mitigate overfitting, making RF more reliable than a single DT.
- **Robustness Across Features:** RF handles categorical and continuous features well, maintaining performance despite missing values or noisy data. For example, it effectively utilizes features like browsing_history and device_type without the iterative complexity of GBT.
- **Precision and Recall Insights**:

High Recall (0.9902): The model captures almost all true positives, meaning it rarely misses a user who would click an ad. This high recall is critical in ad campaigns where missing potential customers could be costly.
Moderate Precision (0.6558): Approximately 66% of predicted clicks are accurate. While not perfect, this indicates that the model can distinguish genuine clicks from noise reasonably well.
F1 Score (0.7891): This balanced measure suggests that RF provides a good trade-off between identifying click opportunities and avoiding false positives.

RF's moderate performance and high recall make it suitable for scenarios where maximizing ad exposure to potential clickers is more important than reducing false positives. However, it may require further tuning or feature selection to match GBT's predictive power.

7. Insights and Observations

- Model Comparison: The superior performance of GBT highlights its ability to handle complex datasets with high-dimensional features. Its higher AUC score indicates better discrimination between clicked and non-clicked ads compared to DT and RF.
- Feature Importance: The encoded categorical features (device_type, browsing_history, etc.) contributed significantly to model predictions. Understanding the role of these features can help in refining ad targeting strategies. The ad_position feature, imputed based on relevance, proved crucial for predicting click outcomes, aligning with real-world observations that ad placement significantly impacts user engagement.
- Data Imputation Effects:
  Filling missing values based on data distributions maintained the dataset's integrity. However, the mode and mean imputation strategies for age introduced some biases, as seen in discrepancies between the most frequent (26) and average (40) ages. Such differences could impact model accuracy and highlight the need for careful imputation strategies tailored to specific datasets.

8. Conclusion and Future Directions

This analysis underscores the importance of choosing the right model and preprocessing strategies for ad click prediction. The Gradient Boosted Tree model demonstrated strong predictive capabilities, suggesting it as the optimal choice for this dataset. Future enhancements could involve:

- **Incorporating Additional Features:** To capture more nuanced user behaviors.
- **Exploring Advanced Techniques:** Such as deep learning or ensemble methods combining multiple classifiers.
- **Platform-Specific Modeling:** Tailoring models to individual platforms (e.g., Google vs. TikTok) for more accurate predictions.

By refining these models, businesses can better understand their audiences and optimize their advertising strategies, ultimately improving user engagement and ROI.

- **Overall Implications for Ad Click Modeling:** Future researcher select the right model based on business goals based on my testing:
  (1) For Maximum Accuracy: Gradient Boosted Trees excel in capturing complex relationships, making them ideal for detailed customer behavior analysis.
  (2) For High Recall: Random Forests are valuable when it's crucial not to miss any potential clicks, such as in high-stakes marketing campaigns.
  (3) For Baseline Comparisons: Decision Trees provide quick insights and serve as a benchmark for more sophisticated models.

9. Limitation

- **Imputation introduces potential biases**

This situation particularly happened when filling in missing values for a variable like age in this project, and this can affect the overall integrity of the data and the predictive model's performance. Here's a detailed explanation of these biases:

- Loss of Variability and Misrepresentation:
  - Issue: When missing values are replaced with the mean age (40), it artificially reduces variability. This leads to a concentration of data points around the mean, misrepresenting the actual distribution.
  - Impact: Real-world data is often skewed or multi-modal, especially for variables like age. Using a single value for imputation assumes a normal distribution and can mask important patterns or age-specific trends in ad-click behavior.
- Mode vs. Mean Discrepancy:
  - Issue: In your dataset, the mode age (most frequent) is 26, but the imputed mean is 40. This difference indicates that the dataset may be skewed toward younger users. Imputing with the mean might disproportionately shift the dataset's demographic representation.
  - Impact: This bias could lead to inaccurate insights about the primary audience for ad clicks. For instance, if younger users (around 26) are more likely to click ads, but the data imputes missing values as 40, the model might underestimate the click potential for younger users.
- Impact on Model Training:
  - Issue: Imputed values can distort relationships between features. For example, age might correlate with device type (younger users might prefer mobile devices), but imputing a uniform value disrupts this relationship.
  - Impact: The model might learn patterns that don't exist in reality, leading to overfitting on certain age groups or failing to generalize across diverse demographics.
- Class Imbalance Impact:
  - Issue: Since the dataset has a relatively high proportion of missing values (about 48% for age), imputing with a single value can create a pseudo-class imbalance. Users with imputed ages might behave differently from those with actual ages.
  - Impact: This imbalance could skew performance metrics. For instance, the model might perform well on training data but poorly on unseen data where age distribution is more varied.

Also, there's another limitation about correlation between click predicted results and all other variables, however because I already used one-hot encoding method, so it brings conflict between variables correlation calculation because correlation wouldn't be able to calculate when all expanded options do not collapse as a whole variable. It became the limitation instead of my research goal in deliver #2 proposal.

One-hot encoding converts each category within a variable into separate binary columns (0 or 1). For example, a device_type with three categories (Mobile, Desktop, Tablet) will create three separate columns: device_type_Mobile, device_type_Desktop, and device_type_Tablet. The original variable loses its unified representation, which complicates correlation calculations between the target variable (click) and these features. This method turned out to be a problem during the correlation measures, such as Pearson’s correlation coefficient, are designed for continuous variables, not binary variables resulting from one-hot encoding. When the categories are split into separate columns, traditional correlation metrics can't capture the relationship between the target variable (click) and the original categorical variable as a whole. Each one-hot encoded column is treated independently, ignoring their combined influence. Instead of understanding how device_type affects ad clicks overall, I’m only evaluating the individual contribution of Mobile or Desktop, which fragments the insight. As a result, I might miss the broader relationship between the original categorical variable and the target. For instance, you can't assess the overall impact of device_type on click; you only see partial correlations for each category.

With one-hot encoding, the binary columns are often correlated, introducing multicollinearity. This can distort the importance of variables in models like linear regression or decision trees.

There might be some alternative approaches in the future such as:

- Target Encoding: Replace categories with the mean target value (click rate) for each category. This retains the original structure and allows correlation calculation while reducing dimensionality.
- Effect Encoding (Deviation Coding): This method represents the relationship between each category and the overall mean, maintaining the categorical variable's relationship with the target.
- Principal Component Analysis (PCA): For reducing the dimensionality of one-hot encoded data, PCA can combine the encoded columns into fewer components, simplifying correlation analysis.
- Categorical Correlation Methods: Use measures like Cramér's V or Chi-square tests to evaluate relationships between categorical variables and the target.

After all, the use of one-hot encoding complicates correlation analysis because it disperses categorical information across multiple columns. While one-hot encoding is useful for many machine learning models, it introduces limitations for understanding variable relationships. Addressing this issue with alternative encoding or correlation methods can help maintain the integrity of your analysis and align with my research goals.