Targeted Marketing Classifier - Selling Laptops

Context

I am the owner of a retail website. I'm planning on running a promotion on a laptop and I want to send emails out about it. However, I only want to send it to people that may be interested in it so as not to annoy people that aren't interested.

Data

I have access to data about who clicked on similar emails in 2020 to help predict which users may be interested in the promotion. I have three datasets that contain information about each user, details about web pages visited by each user, and labels on whether the user clicked the email or not.

Feature Engineering

Features used:

- Past Purchase Amount - this data is given per user.
- Age - this data is given per user
- Total Time - the total number of seconds a user spends on the website.
- Average Time - the average number of seconds a user spends on the website.
- Total Time Laptop - the total number of seconds a user spends on the laptop website page
- Average Time Laptop - the average number of seconds a user spends on the laptop website page
- Total Badges - sum of gold, silver and bronze badgers per user

Other features:

- Laptop Visits - frequency of the laptop website page visits per user.
- Non-laptop Visits - frequency of any website page visits other than laptops per user (initially had a visits feature for each item, but I combined it to reduce dimensionality)
- No visits - binary feature where 1 means the user did not visit the retail website and 0 means the user had at least 1 visit
- One Hot Encoding
    - Gold - number of gold badges per user
    - Silver - number of silver badges per user
    - Bronze - number of bronze badges per user

Model Performance

The model is trained on unseen data and evaluated with an accuracy metric that measures the proportion of correct predictions made by the classifier. I have included StandardScaler in the model pipeline to ensure the data is transformed to work better with the classifier.

I initially hit a plateau when trying to improve the model's performance. I tried to add more features, reduce the dimensionality of the data, use recursive feature elimination to find the strongest features, but the model performance barely improved. However, the accuracy significantly increased when I tried using different classifiers for the model and conducted polynomial feature transformation to handle non-linear relationships in the data.

I selected the Logistic Regression because it offers a high accuracy with a low latency, making it suitable for real-time applications like email marketing. Although the Random Forest model is more accurate, their high latency would cause delays in sending out emails, which would negatively impact the campaign's effectiveness.

Business Outcome

After training the classifier to classify whether a user is interested in purchasing a laptop or not, the next step is to put it into production and implement it into our email marketing campaign. This will ensure that the emails are only sent to users who are genuinely interested in purchasing a laptop, increasing the effectiveness of the campaign and reducing the likelihood of users unsubscribing or marking the emails as spam.

We can then track metrics such as total campaign revenue, average order size and conversion rates to evaluate the effectiveness of the promotion.

Key Lessons

Data - when transforming the data during feature engineering, especially when performing joins, it's important to keep checking the shape of the data and ensuring that I did not leave out any data unintentionally. This can be done by inserting print statements of data.shape as a sanity check.

Model - when trying to improve model performance, I made the mistake of spending too much time trying to add more features. Explore other solutions such as recursive feature elimination, using different classifiers and checking for non-linearity in the data.