Predicting Online Shopping Behaviour

Introduction

Online shopping has become such a normal part of our lives that most of us do not even think about how we behave on a website anymore. We click around webpages, compare products, add items to the cart, and sometimes also leave without buying anything at all. For an online store, this behaviour matters a lot. They need to know which visits are likely to turn into purchases and what kind of visitors they are attracting to their platforms.

In this project, I used two datasets:

1. **UCI Online Shoppers Purchasing Intention dataset**
2. **My own online survey**

The main goals was for this project were:

- To build supervised models using Logistic Regression and Decision Trees to predict whether a session will lead to a purchase.

- To use unsupervised learning to find different types of online sessions and shoppers based on behaviour.

- To compare the results from the enormous UCI dataset with my own small survey results and see if I could find any patterns. While the UCI dataset shows the objective behavioural reality, the survey I conducted is more subjective to how people perceive themselves. When both are examined together, it can highlight whether shoppers accurately understand their own behaviour or whether there are interesting mismatches between intention and action.

UCI Online Shoppers Purchasing Intention Dataset

The UCI dataset is what I chose as my main dataset for this project. It contains clickstream data from an online store, where each row is a single session (or visit to the store). With over 12,000 online sessions, it helps make statistical learning more stable. It also includes features like duration-based metrics, bounce rates, product reviews, and temporal variables. The fields I used in my project were:

- **Administrative, Administrative_Duration:** refers to the number of administrative pages visited and time spent.
- **Informational, Informational_Duration:** refers to the number of informational pages and time.
- **ProductRelated, ProductRelated_Duration:** refers to the product page visits and time.
- **BounceRates, ExitRates, PageValues, SpecialDay**
- **Weekend:** refers to whether the visit was on a weekend (True/False)
- **Revenue – target variable:** 1 if the visit ended in a purchase, 0 otherwise.

First, I dropped rows with missing values and converted Revenue and Weekend to integers.

```
# in uci dataset
# target: 'revenue' (whether purchase was made)
uci_df_clean = uci_df.dropna().copy()
uci_df_clean["Revenue"] = uci_df_clean["Revenue"].astype(int)
uci_df_clean["Weekend"] = uci_df_clean["Weekend"].astype(int)
```

Then, I chose the following numeric features for modelling:

```
# model
uci_features = [
    "Administrative",
    "Administrative_Duration",
    "Informational",
    "Informational_Duration",
    "ProductRelated",
    "ProductRelated_Duration",
    "BounceRates",
    "ExitRates",
    "PageValues",
    "SpecialDay",
    "Weekend"
]
```

This gave me X (features), and y (target). X were the 11 behaviour features and y is the revenue.

After cleaning, the train/test shapes were:

- Training set: (8631, 11)
- Test set: (3699, 11)

Survey Dataset

This dataset comes from my own survey that I conducted via Google Forms, titled ‘Online Shopping Behaviour’. The questions I asked in my survey reflected behavioural features in the UCI dataset such as frequency, browsing time, page views, etc. With 20 responses, it covered questions such as:

- How often do you visit shopping websites?
- Do you usually use the same stores?
- How long do you browse before deciding?
- How many product pages do you view?
- Do you compare prices?
- Do you read reviews?
- Do you shop more on weekends?
- How likely are you to actually buy the product while browsing?

- Did you complete your last online purchase?
- What type of shopper would you describe yourself as? (impulsive, bargain, window shopper, etc.)

I renamed the long text questions to shorter names for the sake of efficiency, like Q1_HowOften, Q5_ComparePrices, etc., and then created mappings to convert the answers into numeric codes. For example:

- Q1: “Rarely” = 1. Up to “Everyday” = 5
- Q9: “Very Unlikely” = 1, up to “Very Likely” = 5
- Q11 (completed purchase): “Yes” = 1, “No” = 0
- Q 12 (shopper type): impulse = 0, bargain = 1, research-heavy = 2, occasional = 3, window shopper = 4

Any rows where mapping failed (because of small wording differences), became NaN and were dropped. This left me with 9 fully clean rows. Since this was a tiny set of data, I treated this survey more as a small supporting analysis, and not the main model.

Methods Used

For both datasets, I did a standard train/test split with 30% for testing and random_state = 42. For models that care about scale (specifically logistic regression and k-means), I used:

```
# scaling for uci models
uci_scaler = StandardScaler()
X_uci_train_scaled = uci_scaler.fit_transform(X_uci_train)
X_uci_test_scaled = uci_scaler.transform(X_uci_test)
```

I also applied the same idea later on the survey features.

Supervised Learning on UCI

I used two supervised classifiers, namely logistic regression and decision tree. I wrote a small helper to train each model and print accuracy + confusion matrix.

```python
def evaluate_model(name, model, X_train, y_train, X_test, y_test):
    print(f"\n=== {name} ===")
    model.fit(X_train, y_train)
    preds = model.predict(X_test)
    acc = accuracy_score(y_test, preds)
    print("Accuracy:", acc)
    print("Confusion matrix:")
    print(confusion_matrix(y_test, preds))
    return acc
```

Logistic regression used the scaled features and the decision tree used the unscaled features as trees don't need scaling.

Unsupervised Learning on UCI

To explore session groups, I applied k-means clustering on all 11 UCI features.

```python
# uci: k-means
X_uci_all_scaled = uci_scaler.fit_transform(X_uci)

kmeans_uci = KMeans(n_clusters=3, random_state=42, n_init=10)
uci_clusters = kmeans_uci.fit_predict(X_uci_all_scaled)
uci_df_clean.loc[:, "Cluster"] = uci_clusters
```

The cluster sizes were:

Cluster 0: 9653 sessions, Cluster 1: 1622 sessions, Cluster 2: 1055 sessions

To visualize them, I used PCA to reduce the scaled data down to 2 components and then plotted a scatterplot of a random sample of 500 points with:

x-axis: Behaviour Dimension 1

y-axis: Behaviour Dimension 2

Survey Models and Clustering

I used similar steps for the survey but kept the expectations realistic because of the sample size. I ran a logistic regression and decision tree. I also applied k-means with k=3 to the survey behaviours and used PCA to plot the 9 points in 2D. The axes were labelled:

1. 'Shopping Behaviour Dimension 1'
2. Shopping Behaviour Dimension 2'
3. Colourbar label: 'Shopper Group (from K-means)'

This gave a clear visual of how the survey respondents grouped together, and I could compare those groups with their self-reported shopper type.

Results

1. **UCI: Logistic Regression**

   From the output, accuracy: ~0.8816, and the confusion matrix:

```
Confusion matrix:
[[3060   64]
 [ 374  201]]
```

   Interpretation:

   - 3060 sessions with no purchase were correctly predicted as no purchases (true negatives).
   - 201 sessions with a purchase were correctly predicted as purchase (true positives).
   - 64 sessions were predicted "purchase" but actually didn't buy (false positives).
   - 374 sessions predicted "no purchase" but did buy (false negatives).

     The model gets about 88% of the test sessions right, which is quite strong.

2. **UCI: Decision Tree**

   From the output, accuracy: ~0.8523, and the confusion matrix:

```
=== Decision Tree (UCI) ===
Accuracy: 0.8532035685320357
Confusion matrix:
[[2837  287]
 [ 256  319]]
```

   The tree does slightly worse than logistic regression overall (about 85% accuracy), but it makes fewer false positives for the non-purchasing class than logistic at the cost of some other errors. This matches the fact that trees are flexible and can fit non-linear patterns, but they can also overfit and don't always beat simpler models like logistic regression on large datasets.

3. **UCI: K-Means and PCA**

   Cluster 0: 9653, Cluster 1: 1622. Cluster 2: 1055

   The PCA scatterplot of 500 sampled sessions showed three visible colour groups:

   - One large, dense cluster around the central area of the plot
   - Two smaller, more separated groups, one stretching along the first PCA axis and one forming a smaller 'branch'.

   Since PCA is a mix of all behaviour variables, the plot doesn't give direct feature names, but it does show that k-means isn't just splitting the data randomly: there are actual distinct patterns in session behaviour.
   In a fuller version of the project I could look at the average feature values per cluster (e.g., average page values, time on product pages) to label them more precisely like "high-engagement sessions" vs "quick bounces", but for this project the main point is to demonstrate the method.

4. **Survey: Logistic regression and Decision tree**

   From the earlier run on the cleaned survey:

   - Logistic Regression (Survey)
     Accuracy ≈ 0.67 (2 out of 3 test cases correct)
     Confusion matrix: [1 0]
     [1 1]

   - Decision Tree (Survey)
     Accuracy = 0.0
     Confusion matrix: [0 1]
     [2 0]

   Interpretation:
   - Logistic regression gives unstable results because of the size of the data.
   - The decision tree completely fails on the test set which is expected behaviour with the small dataset. The tree basically memorizes the training examples and can't generalize.

5. **Survey: K-Means and PCA**

   The survey k-means with k=3 assigned each of the 9 respondents to a cluster. Then I compared cluster labels with the self-reported shopper type. There wasn't a perfect match but a few patterns were observable. For example, some people who called themselves 'occasional shoppers' fell into the same cluster, and impulse shoppers and window shoppers sometimes grouped together.

   The PCA plot for the survey showed three colours spread across a small space with a clear separation between at least two of the groups. With only 9 points, I treated this as more of a visual illustration meant to compare data rather than a statistical finding in itself.

<u>Limitations</u>

- The UCI dataset comes from a specific time window during the 2015-2016 holiday season. Shopper behaviour changes over the years, especially post-pandemic.
- The survey responses recorded were mainly from a narrow network (Toronto-based), not a wide population. Also, age, income, and tech literacy may be unevenly distributed. There might also be response bias, as people tend to overestimate 'rational' behaviours like comparing prices or reading reviews. There were also a couple of mapping issues caused because of differences in wording, which caused NaNs and forced row removal.
- I only used a subset of the UCI features to keep the models manageable for the project and there are probably more patterns that I didn't fully explore.
- The survey dataset is extremely small so its results could not be generalized.
- I did not do hyperparameter tuning (e.g., changing max depth for the decision tree or trying different numbers of clusters), which might improve performance or change the clusters.

Conclusion

This project applied techniques like logistic regression, decision trees, k-means clustering, and PCA to real online shopping data. On the UCI dataset, I was able to predict whether a session would lead to a purchase with around 88% accuracy using logistic regression, and around 85% using a decision tree. K-means and PCA showed that sessions naturally group into a few clusters based on browsing behaviour, which could be used for targeting or personalization. From my survey, the same methods behaved differently due to the tiny sample size, however it also reflected how self-reported behaviour does not always align with actual behavioural patterns. By comparing both datasets, it became clear that people may see themselves as certain types of shoppers such as careful, impulsive, or research-driven, but their numerical responses and the clusters they fell into did not always match these identities.

This comparison helped highlight one of the main goals of the project: examining whether shoppers accurately understand their own online behaviour or whether there are meaningful mismatches between intention and action. Combining objective behavioural logs with subjective survey responses ultimately provided a more complete and realistic picture of how online shopping decisions are made.