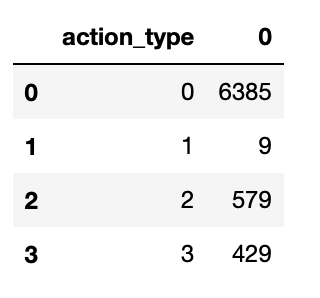
**Dataset statistics and feature ranking**

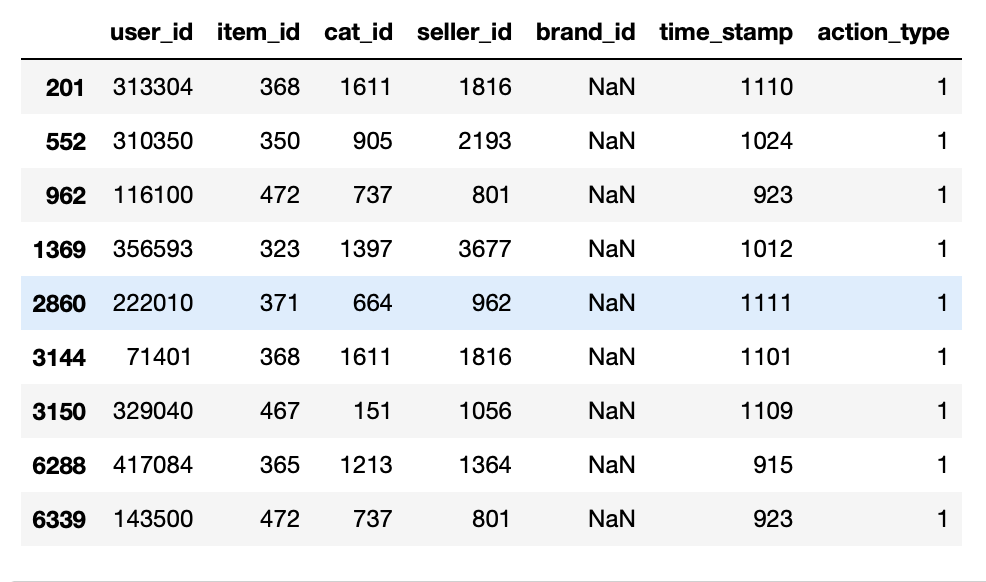
**Dataset Statistics:**

**From user\_logs , we get to know following information :**

1. Total number of sellers is 146. The seller which is popular or widely reachable among the customers is **seller Id - 801** . It is also found that customers bought products the most time from this seller.
2. There are four action types - (0 : click, 1 : add-to-cart, 2 : purchase, 3 : add-to-favorite). Below shows the customer action behavior.



1. It was discovered from the data that customers who chose action type - 1 are very less and brand id is NaN.



1. Total number of categories in the dataset is - 87 and it was found from the data that the most popular and widely reached products belong to the **category Id - 737**. Products from this category are also bought most by the customers.
2. Total number of brands found in this dataset are - 149. Popular brand which customers searched for is **brand Id - 5890**. This is also found to be the most bought brand among the customers. We can infer that brand which are searched widely are bought maximum time buy the customer
3. Below mention some of the unique behavior of the customers.

Figure 6.1 shows the maximum time a buyer purchased is 6 times. Maximum buyers purchased the product only once.

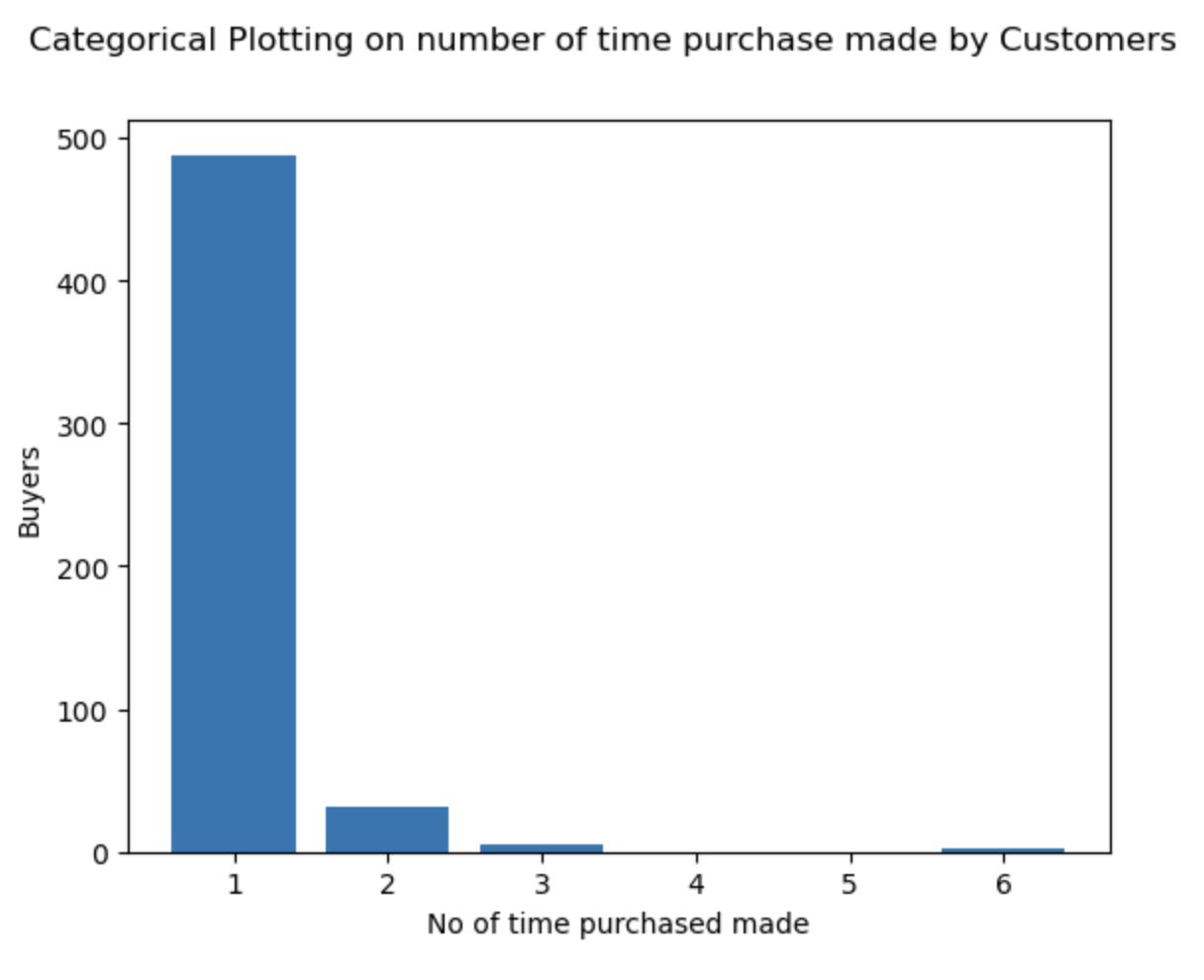


Fig - 6.1

Figure 6.2 shows the maximum customers made clicks between the range of [0-10] times. There are customers who clicked 100 times on different products.

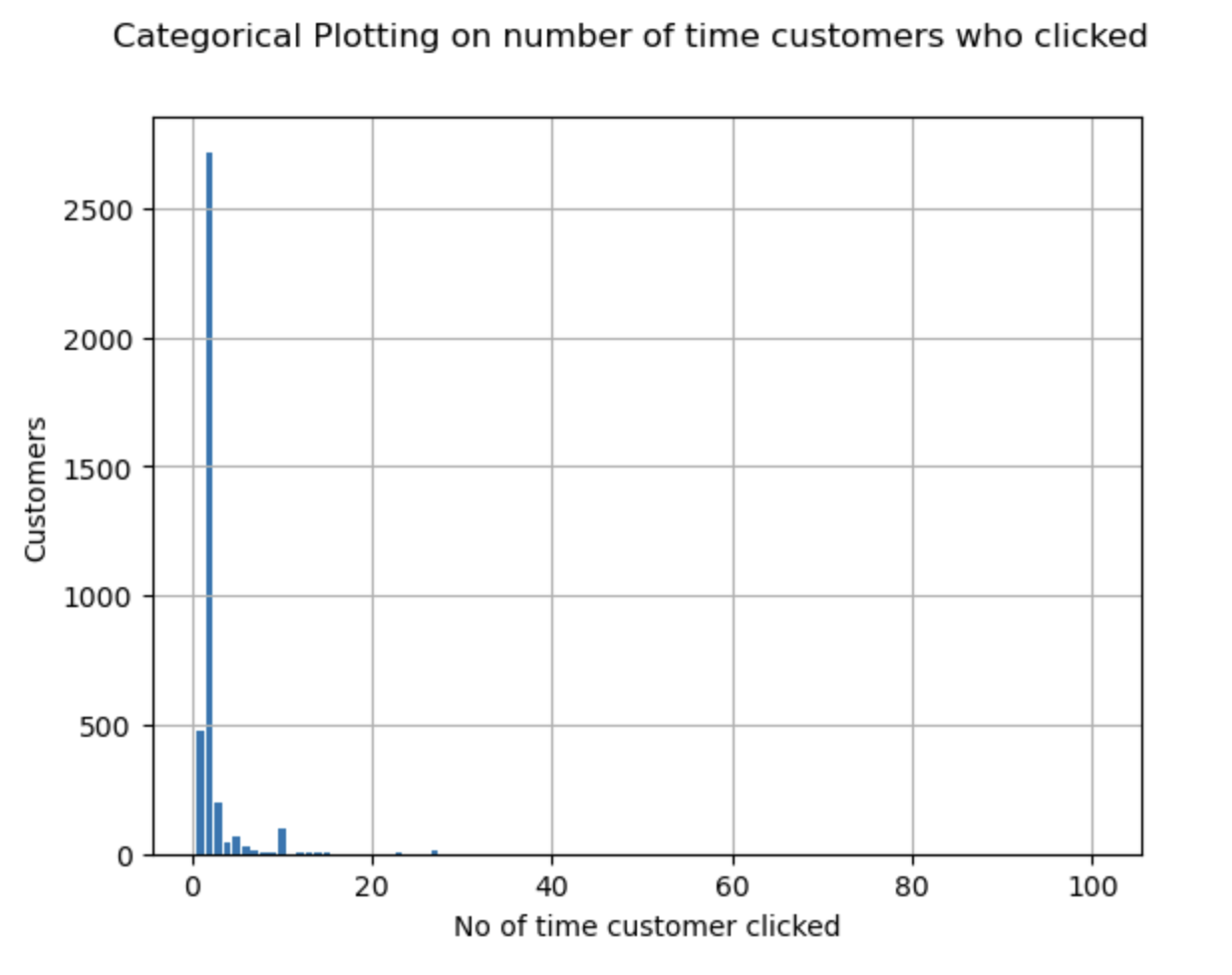


Fig - 6.2

Below Figure 6.3 shows that maximum customers perform action add-to-favorite only once. Maximum number of times a customer adds a product to their favorite is three times.

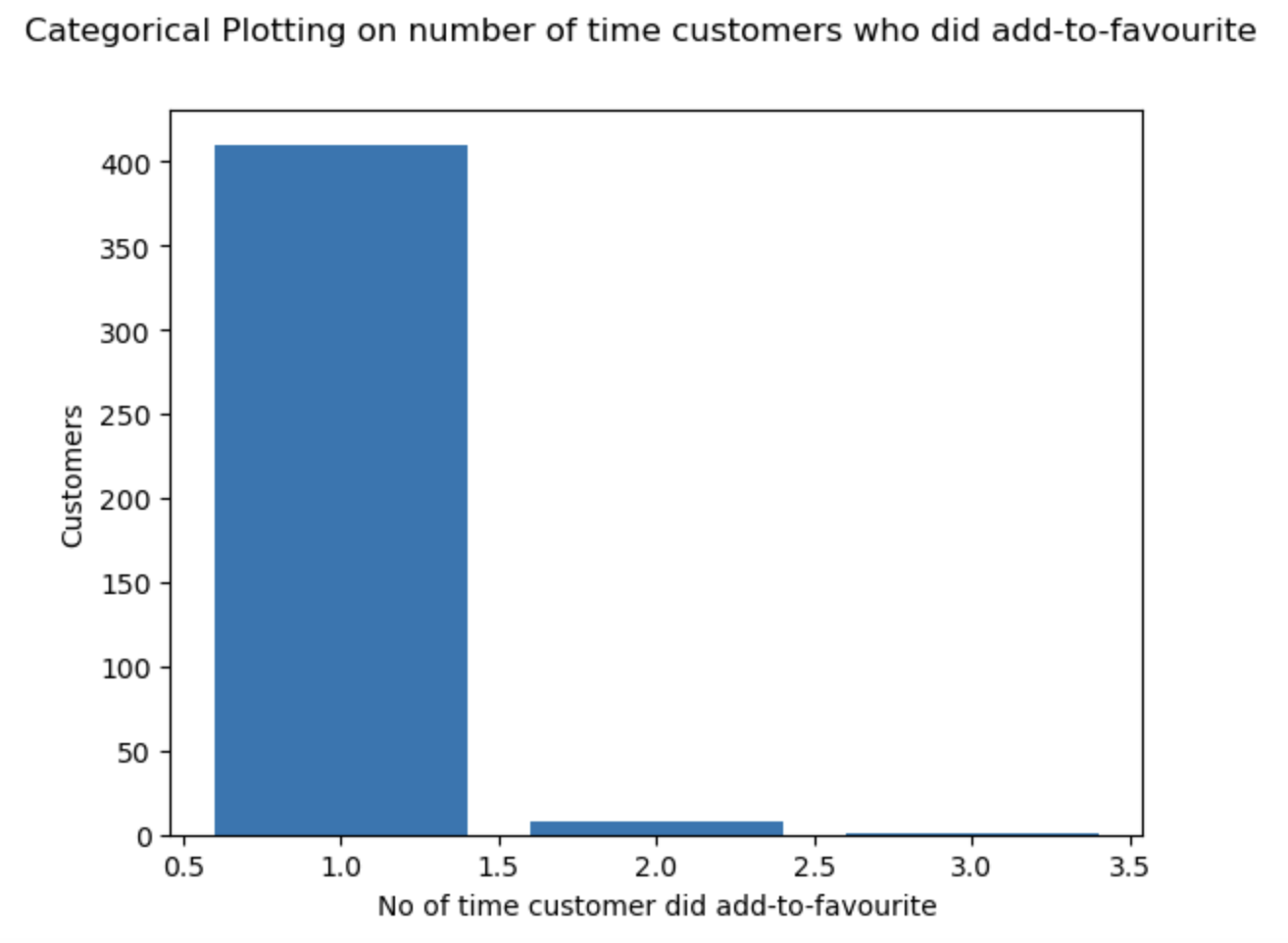
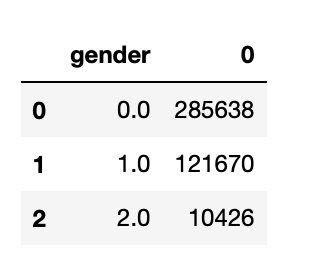


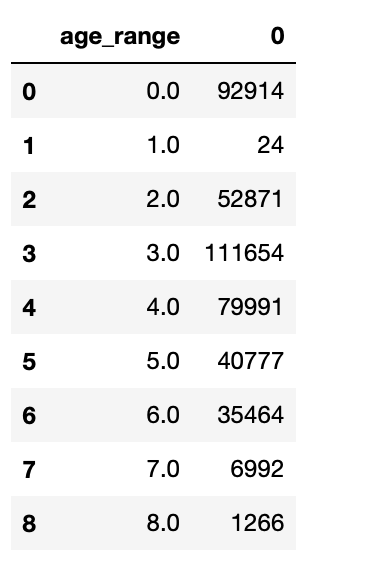
Fig - 6.3

**From user\_profile, we get to know following information:**

1. We found below gender distribution from the dataset. (Gender : 0 - Female , 1 - Male, 0 - Unknown). Female customers are twice the male customers.



1. We get to know the below age distribution of the customers. User' s age range: 1 for <18; 2 for [18,24]; 3 for [25,29]; 4 for [30,34]; 5 for [35,39]; 6 for [40,49]; 7 and 8 for >= 50;0 and NULL for unknown.



**Feature Selection**

The goal of feature selection techniques in machine learning is to find the best set of features that allows one to build optimized models of studied phenomena.

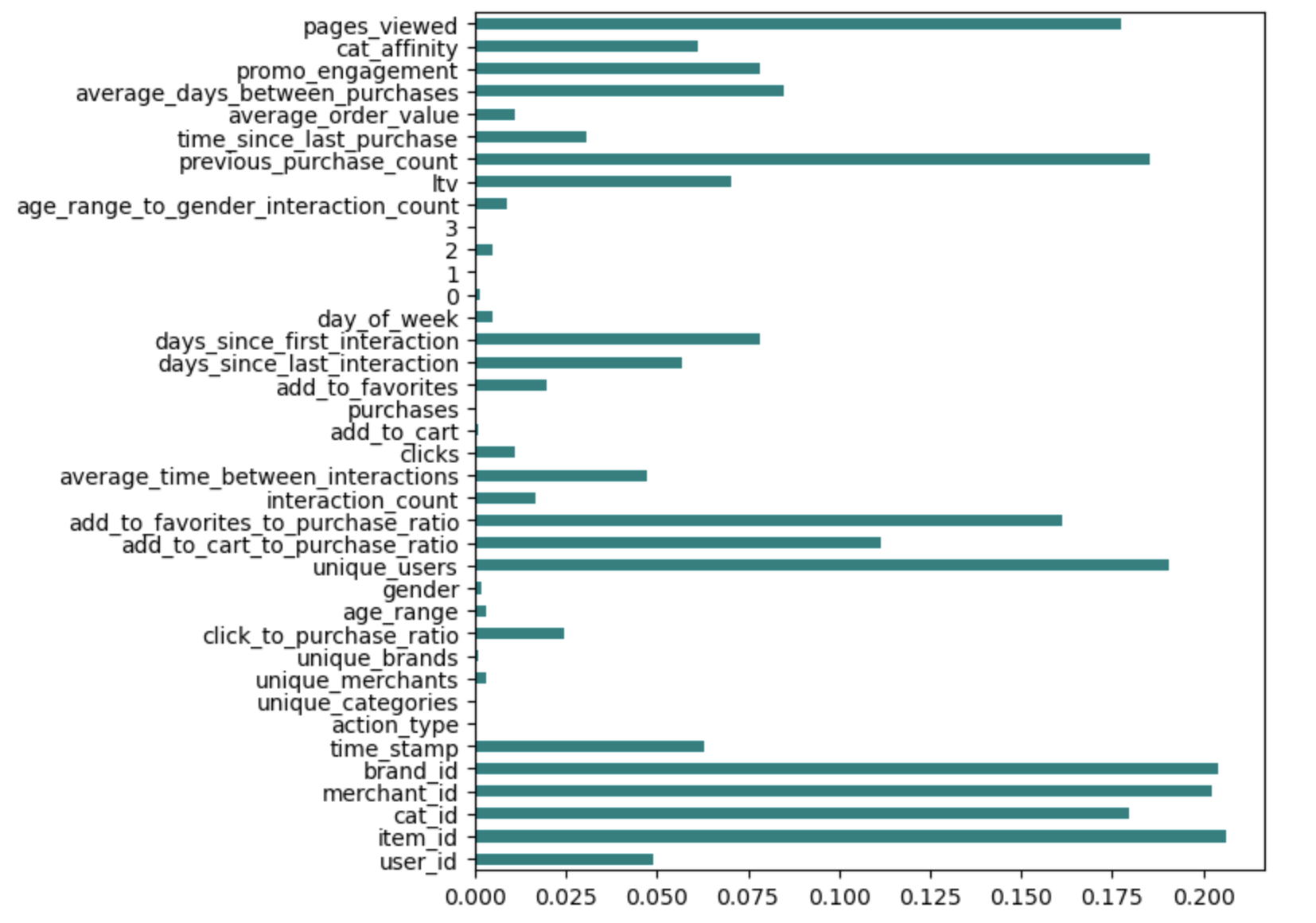
**Three steps in feature selection:**

* Data Preprocessing: Clean and prepare the data for feature selection.
* Feature Scoring: Compute scores for each feature to reflect its importance to the target variable.
* Selection: Select a subset of the most important features based on their scores, and use them for training the predictive model.

We studied different ifeature selection techniques to find the relevant features.

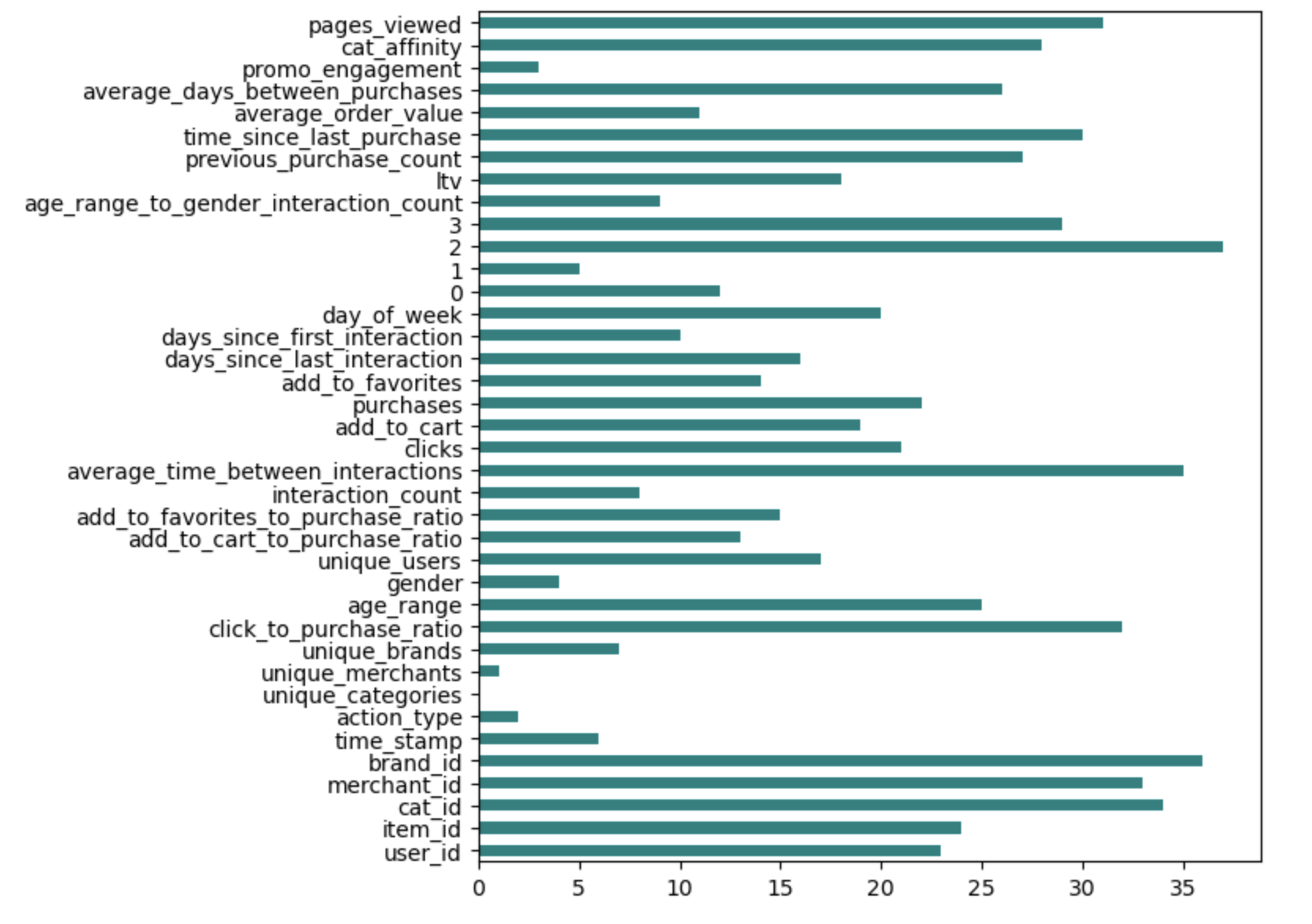
1. **Information Gain**

Information gain calculates the reduction in entropy from the transformation of a dataset. It can be used for feature selection by evaluating the Information gain of each variable in the context of the target variable.



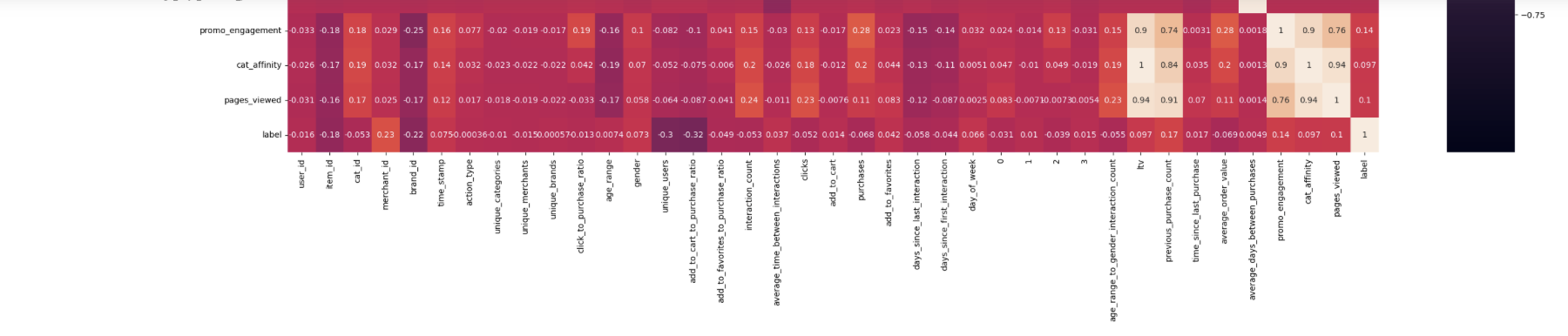
1. **Fischer Score**

Fisher score is one of the most widely used supervised feature selection methods. The algorithm we will use returns the ranks of the variables based on the fisher’s score in descending order. We can then select the variables as per the case.



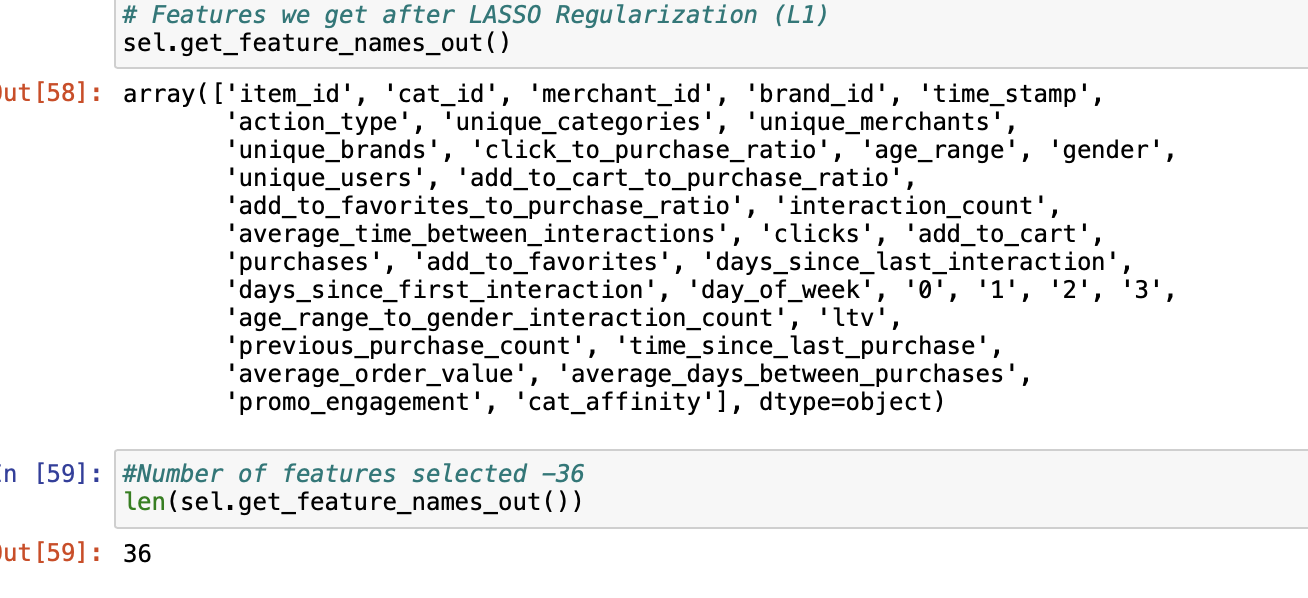
1. **Correlation Coefficient**

Correlation is a measure of the linear relationship between 2 or more variables. Through correlation, we can predict one variable from the other. The logic behind using correlation for feature selection is that good variables correlate highly with the target. Furthermore, variables should be correlated with the target but uncorrelated among themselves.



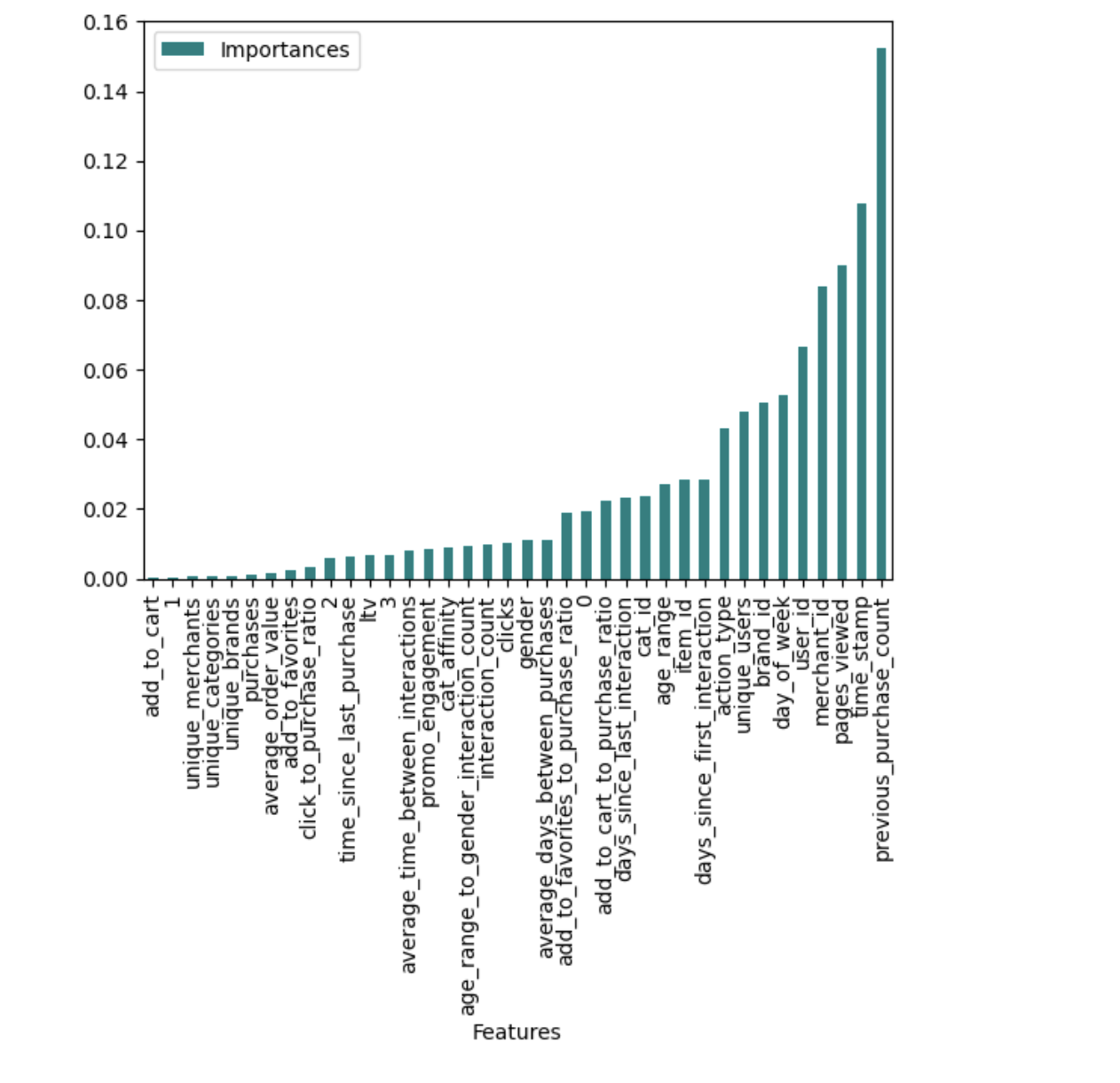
1. **LASSO Regularization (L1)**

Regularization consists of adding a penalty to the different parameters of the machine learning model to reduce the freedom of the model, i.e., to avoid over-fitting. In linear model regularization, the penalty is applied over the coefficients that multiply each predictor. From the different types of regularization, Lasso or L1 has the property that can shrink some of the coefficients to zero. Therefore, that feature can be removed from the model.



1. **Random Forest Importance**

Random Forests is a kind of Bagging Algorithm that aggregates a specified number of decision trees. The tree-based strategies used by random forests naturally rank by how well they improve the purity of the node, or in other words, a decrease in the impurity (Gini impurity) over all trees. Nodes with the greatest decrease in impurity happen at the start of the trees, while notes with the least decrease in impurity occur at the end of the trees. Thus, by pruning trees below a particular node, we can create a subset of the most important features.



**Principal Component Analysis**

PCA is a dimensionality reduction approach that converts a dataset’s columns into a new group of characteristics called Principal Components (PCs). The variance in a column is the information contained in it. The main aim of Principal Components is to express data in the smallest number of columns possible.

PCA is a process for reducing the complexity of high-dimensional data while preserving trends and patterns. It accomplishes this by condensing the data into fewer components, which can be assumed as feature summaries.

Components are unrelated features that are composites of the original features. They are also assigned so that the first has the most variance in the data, the second for the second most variance, and so on.

PCA is highly beneficial when working with large data sets with many characteristics. Image processing and genomic research are two examples of typical applications that deal with hundreds, if not tens of thousands, of columns.

We implemented PCA with following values of n\_components=10,20,25. It was found that we got the same accuracy 74.6% for our dataset.

****

**SHAP Implementation**

**SHAP (SHapley Additive exPlanations)** is a game theoretic approach to explain the output of any machine learning model. It connects optimal credit allocation with local explanations using the classic Shapley values from game theory and their related extensions.

**SHAP Mean Absolute Value**

SHAP Mean Absolute Value is a method to measure the feature importance by taking the absolute value of all SHAP values for a feature and then taking the average across all samples.

The results of using the mean absolute value of the SHAP values as a filter method for feature selection on the given data showed that the top 5 features selected were 'action\_type', 'time\_stamp', 'cat\_id', 'user\_id', and 'thal'. This means that these 5 features had the highest mean absolute value of the SHAP values and were considered the most important in terms of having an impact on the target variable. These results can be used as a guide for further analysis or for building a model with only these selected features. It's important to note that this method only provides a rough estimate of feature importance and other methods or techniques should also be considered for a more comprehensive analysis.

**SHAP Value Plot**

The SHAP value plot shows the contribution of each feature to the prediction of the model, where a positive value means a feature increases the prediction and a negative value means it decreases the prediction.

**OmnixAI**

1. **Introduction**

**OmniXAI (short for Omni eXplainable AI)**is a Python machine-learning library for explainable AI (XAI), offering omni-way explainable AI and interpretable machine learning capabilities to address many pain points in explaining decisions made by machine learning models in practice. OmniXAI aims to be a one-stop comprehensive library that makes explainable AI easy for data scientists, ML researchers and practitioners who need explanation for various types of data, models and explanation methods at different stages of ML process

1. **Tabular Explainer**

Here we are working on tabular data.The package **omnixai.preprocessing** provides several useful preprocessing functions for a Tabular instance. TabularTransform is a special transform designed for processing tabular data. By default, it converts categorical features into one-hot encoding, and keeps continuous-valued features. The method transform of TabularTransform transforms a Tabular instance to a numpy array. If the Tabular instance has a target/label column, the last column of the numpy array will be the target/label. After data preprocessing, we train a **XGBoost classifier** for this task.

1. LIME, SHAP and MACE generate **local explanations** while PDP (partial dependence plot) generates **global explanations**. explainer.explain returns the local explanations generated by the three methods given the test instances, and explainer.explain\_global returns the global explanations generated by PDP. TabularExplainer hides all the details behind the explainers, so we can simply call these two methods to generate explanations.
2. we created a **PredictionAnalyzer** for computing performance metrics for this classification task.
3. Below are screenshot of OmniXAI Dashboard.

