**Business Goal and Machine Learning Solution**

The aim of demo 2 is to build an end-to-end machine learning pipeline using the Black Friday dataset. The overarching business question is to identify the most profitable customers, those who make the highest purchases, in order to specifically target and address them. The ultimate business goal is to increase profits, which can be achieved by focusing on the most lucrative user segment.

The chosen machine learning use case involves supervised learning, specifically a regression prediction of a continuous measure, which in this scenario is the purchase amount. The model is implemented using XGBoost in Vertex AI.

With customer data at hand, predictions can be made, for instance, during special occasions like Black Friday, estimating how much customers are likely to spend. This prediction serves as a basis for strategic marketing measures aimed at retaining the most profitable users, such as keeping them engaged on the website to reduce churn.

Additionally, the model allows for proactive initiatives, like offering incentives such as discount codes or advertising higher-priced products to the targeted customers. By deploying this machine learning algorithm in a production environment, optimization not only occurs during special shopping occasions but also lays the groundwork for customer retention management. This, in turn, enables more targeted advertising efforts, increasing the return on marketing investment.

**Data Exploration**

In the course of the data exploration, an examination of the dataset provided insights across various dimensions. Beginning with demographic details, there were a total of 5891 unique users identified, each associated with diverse demographic attributes. Genders were represented by the categories 'M' and 'F'. Age groups were categorized into seven segments: 0-17, 18-25, 26-35, 36-45, 46-50, 51-55, and 55+, offering a comprehensive overview of the age distribution among users.

Furthermore, the dataset encompassed occupation information, characterized by categories ranging from 0 to 20. However, the precise meaning of these occupation categories remained unclear. Similarly, city categories labeled as A, B, and C were present, with their specific significance yet to be discerned. The variable denoting the duration of the current stay in the city was distributed across the categories 0, 1, 2, 3, and 4 or more. Marital status was indicated by binary values, where '1' denoted married individuals, and '0' represented those who were not married.

Product-related insights revealed distinct product categories across three dimensions: Product Category 1 ranging from 1 to 20, Product Category 2 spanning 2 to 18, and Product Category 3 covering 3 to 18.

Data types predominantly comprised integers, indicating a consistent data format throughout the dataset.

A thorough inspection for null values was conducted, revealing missing entries in specific variables. Notably, product\_category\_2 exhibited 173638 missing values, and product\_category\_3 showed 383247 missing values. Further scrutiny identified actual missing values, being 1 in product\_category\_2 and 2 in product\_category\_3. This meticulous examination of null values provided crucial information for subsequent data preprocessing steps.

The analysis of the dependent variable, representing purchase amounts, revealed valuable insights. The dataset displayed a minimum purchase amount of 12 and a maximum of 23961. The mean purchase amount stood at 9263.97, indicating the average spending, while the median was 8020, representing the middle point of the distribution. These metrics provide key benchmarks for understanding the range and typical values of purchase behaviour in the dataset. A visual inspection using a histogram shows that most customers are spending between 0 and around 14,000 with two deflections at around 15,000 and 20,000. The top three users regarding number of purchases are the ones with the  User IDs = 1001680, 1004277, and 1001941.

A graph with blue lines

Description automatically generated

A visual inspection of the boxplot shows no outliers.

A blue rectangular object with black lines

Description automatically generated

Data exploration further included an inspection of the assumptions of linear regression, due to the underlying regression problem. Since the independent variables are predominantly categorical and not normally distributed (e.g., age in the following histogram), linear regression is not suited as a base model.

A graph with blue lines

Description automatically generated

The decisions influenced by the data exploration phase have a profound impact on the subsequent stages of the machine learning pipeline, encompassing data, model algorithm, and architecture considerations.

In terms of data, feature selection was a critical decision. This involved excluding user and product IDs while incorporating all other features into the model. The rationale behind this choice was driven by the limited options for independent variables and the objective of maximizing information inclusion.

For the model algorithm, the presence of numerous categorical variables posed a challenge, potentially violating regression assumptions. To address this, the algorithm selection was guided by the criteria of compatibility with categorical variables and resilience to the assumption violations inherent in linear regression.

Architecture decisions were driven by the need for flexibility in design. To achieve this, Kubeflow was chosen to provide higher flexibility in the architecture. Transformation steps were implemented using sklearn transformers instead of BigQuery views to enable dynamic settings in a production environment. Training and output components were integrated into the architecture, with hyperparameter tuning organized within buckets. Additionally, pipeline steps were included to accommodate the flexibility required for handling new, incoming data in the evolving landscape of the machine learning application. These decisions collectively contribute to the adaptability and robustness of the machine learning pipeline.

**Feature Engineering**

Feature engineering played a pivotal role in refining the dataset for the machine learning model. Several key transformations were implemented:

*Filling NULL Values:*

Missing values in both categories 2 and 3 were addressed by strategically filling them with two specific numbers, 1 for category 2 and 2 for category 3. This method ensured a seamless integration of the dataset while maintaining consistency.

*Binary Encoding for Gender:*

To expedite processing, the gender categories 'M' and 'F' were converted into numerical values, specifically 0 and 1. This feature engineering step not only facilitated faster computation but also aligns well with the XGBoost algorithm's efficiency in handling numerical representations.

*One-Hot Encoding (OHE) for Categorical Variables:*

OHE was applied to categorical variables such as age groups, city categories, and the duration of stay in the city. This encoding strategy is particularly beneficial for XGBoost, as it allows the algorithm to effectively leverage categorical information, enabling it to capture intricate relationships and patterns within these variables.

In terms of feature selection for the machine learning model, all features were initially considered, excluding user and product IDs. The decision to include all other columns stemmed from the understanding that each feature retained a causal relationship with the purchase variable. The first step involved utilizing a filter method, employing a chi-square test. While this test indicated no significance for any independent variable, notable effect sizes were observed for city category A, age group 18-25, and age group 26-35. Consequently, these variables were deemed essential and were definitively included in the model. Subsequent steps involved wrapper methods, integrated with model evaluation, to further enhance the feature selection process. This comprehensive approach aimed to ensure the incorporation of influential variables and optimize the model's predictive capabilities.

**Preprocessing and the data pipeline**

The initial preprocessing of the data was conducted using BigQuery, with the resultant output stored as a view. This encompassed three primary steps: addressing missing values in category 2 and category 3, converting gender categories 'M' and 'F' into numerical values (0 and 1), and applying one-hot encoding (OHE) for age, city category, and the duration of stay in the city.

To enhance the dynamism of the process, an alternative approach was adopted, utilizing an sklearn transformer that encapsulates the three preprocessing steps. Unlike querying data directly from BigQuery, this method involved saving the data into cloud storage. This not only ensures reproducibility but also contributes to cost efficiency by minimizing the need for repetitive data extraction.

The subsequent stages of the pipeline involved a data cleaning step and the training of the model. The training function incorporated an XGBoost model, with the final outcome encompassing both predictions and the trained model. This comprehensive approach streamlined the preprocessing workflow, promoting adaptability, and maintaining a balance between efficiency and reproducibility throughout the machine learning pipeline.

* Machine learning model design and selection

Decisions which are based on the data exploration are concerning dealing with mainly categorical independent variables. Linear regression, which assumes a linear relationship between predictors and the response variable, may face challenges when handling categorical features. Violations of the assumptions, such as non-linearity and non-constant variance, can occur, potentially leading to suboptimal model performance.

In contrast, Gradient Boosting, being an ensemble learning algorithm based on decision trees, is inherently well-suited to handle categorical variables. Decision trees can naturally accommodate non-linear relationships and interactions, making Gradient Boosting a robust choice for datasets with a mix of categorical and numerical features. The algorithm's ability to capture complex patterns and interactions is advantageous in scenarios where linear models may fall short.

To enhance the performance of the Gradient Boosting algorithm in this context, employing One Hot Encoding for categorical variables is recommended. One Hot Encoding transforms categorical variables into binary vectors, creating a binary column for each category. This approach ensures that categorical information is appropriately incorporated into the model, mitigating potential biases and allowing the algorithm to leverage the full information encoded in the categorical features. This encoding strategy helps to overcome challenges associated with linear regression assumptions and ensures that the Gradient Boosting algorithm can effectively utilize the categorical information present in the Black Friday dataset, ultimately leading to improved predictive performance and a more accurate representation of the underlying patterns in the data.

* + Which machine learning model/ algorithm(s) were chosen for demo 2
    - Base model = Regression
    - Testing of: XGB, Ensemble Stacking, FFNL
  + What criteria was used for machine learning model selection
    - Compute instances used
    - MSE, RMSE, R2
    - Model interpretability and feature importance
* Machine learning model and development
  + Dataset sampling used for model training
    - dev/test independent dataset incl. justification of sampling methods
    - Data set already split into training and test data set
    - Training data including DV and test data rather evaluation dataset
    - Sampling used = sklearn train test split including dividing into X and y
    - Train-Test-Split = 80/20 -> arguing with small amount of data at hand
  + Implementation of model training
    - Adherence to GCP best practices for distribution, device usage and monitoring
  + The model evaluation metric that is implemented
    - R2 and RMSE -> discussion of why the implemented metric is optimal given the business question/ goal being addressed
  + Hyperparameter tuning and model performance optimization
    - Grid and Random Search
    - Running the Hyperparameter Tuning trough different buckets on Cloud
  + How bias/ variance were determined
    - From train-dev datasets
    - Tradeoffs used to influence and optimize ML model architecture
* Machine learning model evaluation
  + How the ML model, post-training and architecture/hyperparameter optimization performs on an independent dataset
  + How the model addresses the business question and how it performs on independent datasets
* Fairness analysis
  + Possible fairness and bias implications of a profit maximization model trained for targeted marketing
  + How to determine model biases and how to mitigate biases
    - Biases like treating the upper class better because of higher purchase power
      * Determined by City and Occupation
    - Gender inequality
    - Age bias -> older people likely to purchase higher amounts
    - Order frequency not part of the underlying model
    - Marital status because of higher purchasing power
    - Product category as the only variable without a bias
  + Tracking more variables for the purpose of coming to an unbiased outcome
  + Not possible to perform a completely unbiased analysis based on the data at hand
* Discussion of the implications of including purchaser demographics in a model used for targeted marketing
* Test for bias
  + Fairness indicators
  + Comparing the model performance with and without demographics
* At least one way of mitigating bias
  + Removing demographics and location fields, using mindiff to equalize profit predictions across demographic characteristics
* Stating model should no be used for marketing is acceptable in lieu of a discussion of how to correct bias
  + Could result in a brand firestorm and people leaving the commerce as a consequence
* Model application on GCP
  + Proof that the ML model/ application is deployed and served on GCP with Vertex AI or Endpoint
* Callable library/ application
  + ML model is a callable ML model or application
* Editable model/ application
  + Demonstrate that the model is customizable, fully functional after an appropriate code modification as might be performed by a customer