Key Steps in Machine Learning Model Development | Black Friday DataSet

For this Demo we addressed the business case of understanding sales and customer behavior during Black Friday, focusing on high-volume products, using machine learning and deploying a model that will return the forecasted sales by receiving some feature values. The proposed machine learning solution, a Random Forest Regression model deployed via Google Cloud services, is designed to provide accurate sales forecasts. These forecasts, provided by customer segments (made of customer demographics and other variables) allows the company to make data-driven decisions, optimizing operations and enhancing results in key customer segments.

In this section we will perform a description of the identified business needs being addressed in this demo, and how the proposed machine learning solution will address this business need translated into a business goal.

As it will be detailed in this document, one key business need was identified and approached in this use case: the necessity to understand sales behavior (and customer behavior) regarding a set of high-volume products involved in its Black Friday events. By doing so, the retail company involved in this use case, expects to gain a more detailed knowledge about these behaviors and seeks to identify important insights from the provided data that could provide input for planning and actions aiming to improve its operations and results (profits).

Once this understanding is accomplished/ achieved, the next identified business need is to have good predictions for sales of products involved in Black Friday events, regarding identified features (related to customer demographics, different product categories and son on), so that the retail company could implement specific actions seeking to improve its results (like, for example, implementing specific marketing campaigns for customer segments with lower average purchases).

By putting together model sales predictions and the associated costs of making the products available, it is possible to derive good estimates of profits for each of the different considered segments (made off of specific customer demographics/ product categories) and also to detect opportunities and possible actions on specific segments to improve results.

Henceforth, the identified business goals for this Demo 2 are:

i) to be able to have important business insights from the Black Friday sales dataset and

ii) once these insights are known, to be able to have a machine learning solution that can provide reliable predictions about product sales in Black Friday events, from a set of selected features, so that the company can take actions in order to explore different customer behaviors along the different products, in order to seek for improvement in its results (like profits) in these events.

In this sense, a machine learning solution (final trained model) can be served in order to provide online (or batch) predictions for the retail company, for any given provided data and, based on its predictions, the company can then take actions on each customer product segment where it finds opportunities to improve sales .

Hence, the Machine Learning use case is defined as to present a complete machine learning workflow, ranging from data exploration to the machine learning model deployment in order to provide these black friday sales predictions. The proposed workflow aims to be composed exclusively of Google Cloud services.

As it will be shown later, the proposed Machine Learning model to keep up with these business goals was a Random Forest Regression model, deployed in Google Cloud Model Repository to an endpoint so that the deployed model can receive requests and provide forecasts for the purchases, made by different customers, with different demographics and for different product categories involved in Black Friday events.

The objective is to develop a model to predict sales of various products involved in Black Friday events, that would aid them in creating personalized actions for different customers/ products segments along with understanding which areas make more sales during Black Friday

The data used in this demo is composed of two datasets provided by Kagle for this use case: A train dataset and a test dataset.

The definition of done for the developments of this demo is a presentation of the complete workflow (ranging from data exploration to model deployment and test) in order to make an initial machine learning available for predictions of the aforementioned time durations of the taxi trips in Chicago city (regardless of the model performance). The developments made here end with the machine learning model deployment (after its training and testing) and a practical test of the deployed model.

Accordingly, the main objective of this use case is to illustrate how to implement a complete workflow for this purpose (or course suggestions for next steps in these developments and what

Data exploration

The data exploration process involved analyzing the provided train and test datasets using pandas for data inspection, missing value detection, and correlation analysis. Initial steps included reviewing data types, examining empirical distributions, and addressing missing values in key features, leading to the removal of *Product_Category_3* due to excessive null values. Label encoding was applied to categorical variables, and correlation analysis revealed significant relationships, such as the negative correlation between *Product_Category_1* and purchase amounts. These findings influenced the decision to retain certain features despite correlations, and to exclude others, such as *Product_Category_3*, from model training. Tools like pandas and visualization libraries

helped in understanding patterns, guiding the architecture towards using Random Forest Regression for predicting Black Friday sales.

Description of the types of data exploration implemented and how they were performed

For the purposes of the data exploration, the implemented steps were:

- To get to know each of the involved variables in the provided datasets (train and test datasets).

- Identify data types and possible changes in some of the initial data types present in train and test datasets.

- Identify columns that can be immediately discarded for the purposes of training and delivering a predictive model for the target variable.

- Identify empirical distributions of the variables on the dataset and if any transformations on the initial variables are to be implemented.

- Correlation analysis for the variables in the dataset and which variables are to be kept considering the correlation patterns identified in the data .

- Check the existence of missing values for each of the variables contained in the dataset and which action to be taken regarding these missing values for each initial variable.

- Check the necessity of possible transformations and scaling on the data for the purpose of developing a predictive solution for the purchases in Black Friday events.

- Check if there are outliers in the dataset that deserve special attention/ treatment.

Steps followed on the exploratory data analysis:

Initially, we have copied the train and test datasets to the already specified Cloud Storage bucket.

After copying this dataset to the Cloud Storage bucket we checked the data types of the fields in the datasets, by visualizing the data as a pandas dataframe , we have not identified any variable that has no usefulness for the purpose of prediction.

We have not identified any data type changes in the train and test datasets.

We have analyzed the distinct values of each of the columns contained in the datasets.

It was identified the existence of missing values in the Product_Categiry_2 and Product_Categiry_3 variables in both datasets. It was decided to discard the second variable in

these datasets due to excessive missing values.

After the elimination of this variable from the datasets, all data rows containing any additional missing values were discarded.

Later, we have label encoded all categorical variables in the datasets, so that they could be used in the regression models training sessions to be carried out afterwards.

In the sequence, we have analyzed the empirical distributions of the variables present in the train dataset, as well as checking if there were any important outliers in this dataset that could be discarded from consideration/ for model training purposes.

Empirical distributions of average purchases, grouped by other features values were also analyzed in order to gain insights about purchases behavior.

We checked the correlation patterns amongst some pairs of features. Some significant correlations between specific pairs were detected. Despite the identified correlation patterns, we have not discarded any additional variable becaouse of correlation patterns.

Key findings in the data exploration step

- Correlation Patterns key findings:

We have identified that Product_Category_2 and Product_Category_1 has significant magnitude (0.54). Similar fact is observed in the Pearson correlation coefficient between Product_Category_2 and Product_Category_3.

We have identified a correlation of 0.229 between Product_Category_1 and Product_Category_3, which also was not considered critical.

As it will be depicted below, Product_Category_3 was discarded from any consideration for the model building purposes (because of the number of missing data it contained).

Regardless of the identified correlation between Product_Category_2 and Product_Category_1 we have decided to keep both in the dataset for model training as the size of the identified correlation was not considered critical for considering both variables.

Another important finding in the correlation analysis step is that The correlation between feature Product_Category_1 and target variable Purchase was of -0.3437, indicating a significant pattern, were we observed a negative correlation between this feature and the considered target.

Similar pattern was identified between feature Product_Category_2 and target variable Purchase (The correlation between this feature target variable was of -02099), also indicating that these two variables tend to move in opposite directions (that is, when the values of Product_Category_2 tend to be higher, the purchases tend to be lower).

CODE SNIPET:

Correlation Analysis

Below one has the correlation matrix for the considered features

```
In [8]: continuous_features = dados[['Product_Category_1', 'Product_Category_2',
        'Product_Category_3', 'Purchase']]

        corr = continuous_features.corr()
        corr.style.background_gradient()
```

Figure 9: Correlation matrix between features in sample dataset

	Product_Category_1	Product_Category_2	Product_Category_3	Purchase
Product_Category_1	1.000000	0.540583	0.229678	-0.343703
Product_Category_2	0.540583	1.000000	0.543649	-0.209918
Product_Category_3	0.229678	0.543649	1.000000	-0.022006
Purchase	-0.343703	-0.209918	-0.022006	1.000000

- We have identified unnecessary features like unike_key (unique identifiers of each row)

Missing Value Key Findings:

As already mentioned, all missing values in the train and test datasets were detected along features Product_Category_2 and Product_Category_3. We have decided to discard Product_Category_3 because of the excessive number of missing values and after this, we have also discarded all data rows in train and test containing any remaining missing data.

We have decided to discard the entire feature Product_Category_3 instead of adopting any other data strategy (like for example, to replace missing data with records from previous rows, or with averages of other records) because we have decided tp keep records as original as possible.

Code snippets and figures below shows the missing values identified in the train and test datasets.

CODE SNIPPET:

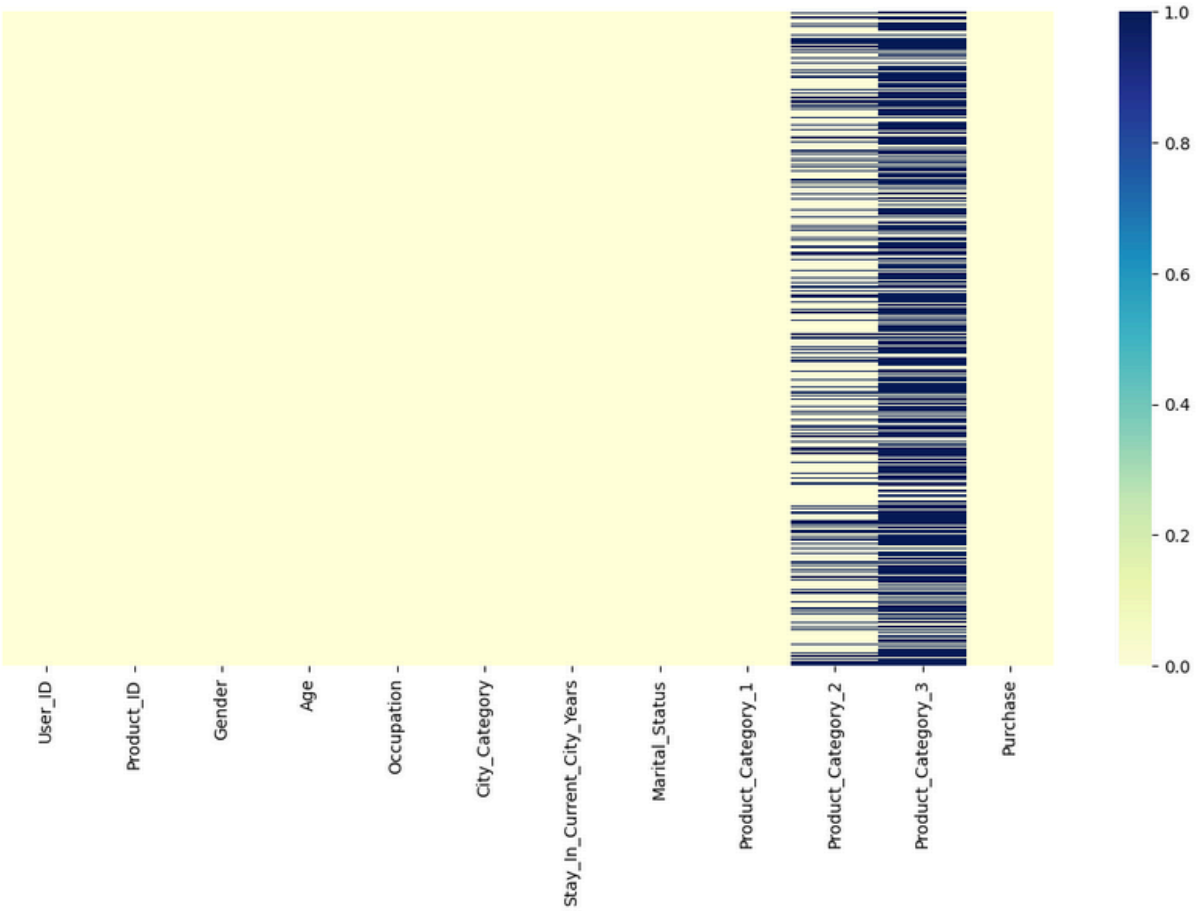
One can see that some fields have null values: Product_Category_2 and Product_Category_3

```
In [15]: dados.isnull().sum()

Out[15]: User_ID                           0
         Product_ID                        0
         Gender                            0
         Age                               0
         Occupation                        0
         City_Category                     0
         Stay_In_Current_City_Years        0
         Marital_Status                    0
         Product_Category_1                0
         Product_Category_2           173638
         Product_Category_3           383247
         Purchase                          0
         dtype: int64
```

```
In [16]: plt.figure(figsize=(14,7))
         plt.tight_layout()
         sns.heatmap(dados.isnull(),yticklabels=False, cmap='YlGnBu')
         plt.show();
```

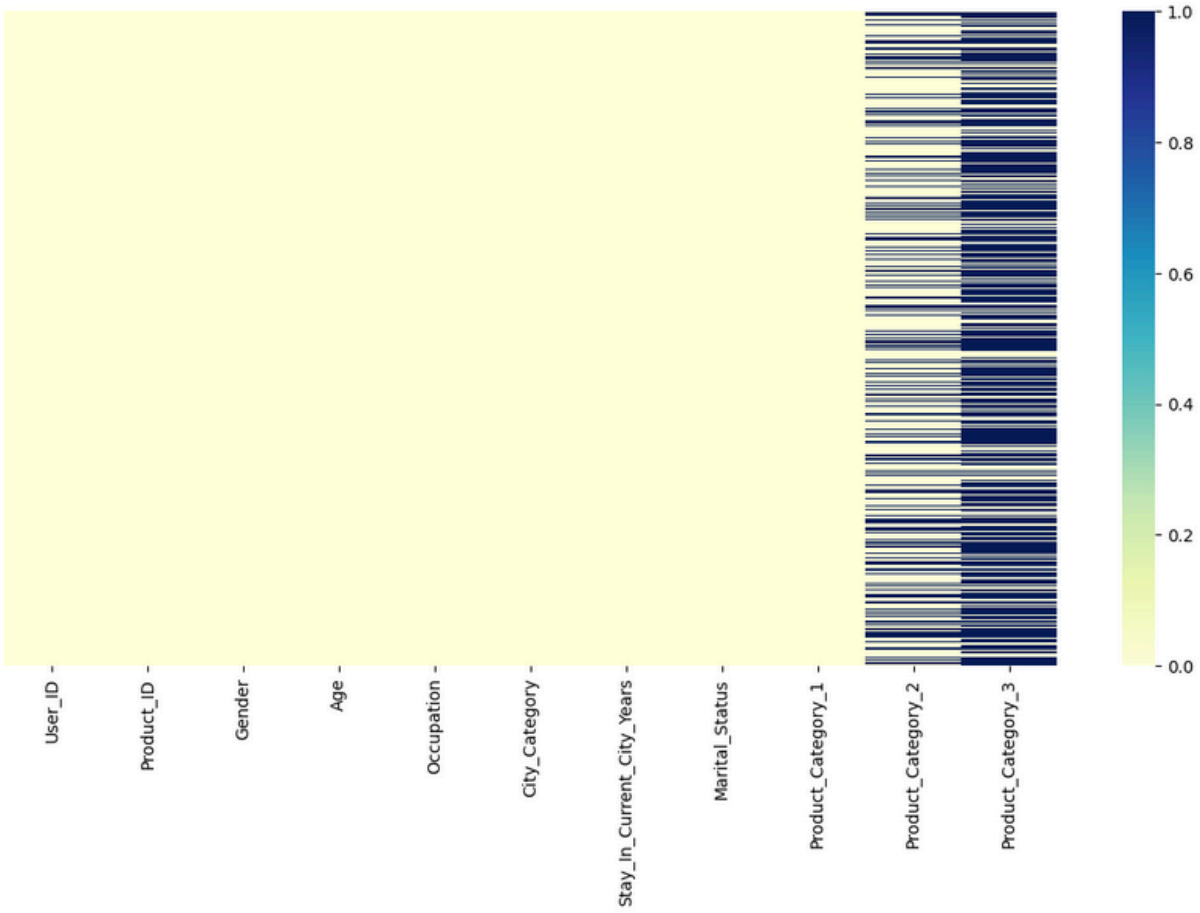
Figure 10: Null values along train dataset



CODE SNIPPET:

```
plt.figure(figsize=(14,7))
plt.tight_layout()
sns.heatmap(test.isnull(),yticklabels=False, cmap='YlGnBu')
plt.show();
```

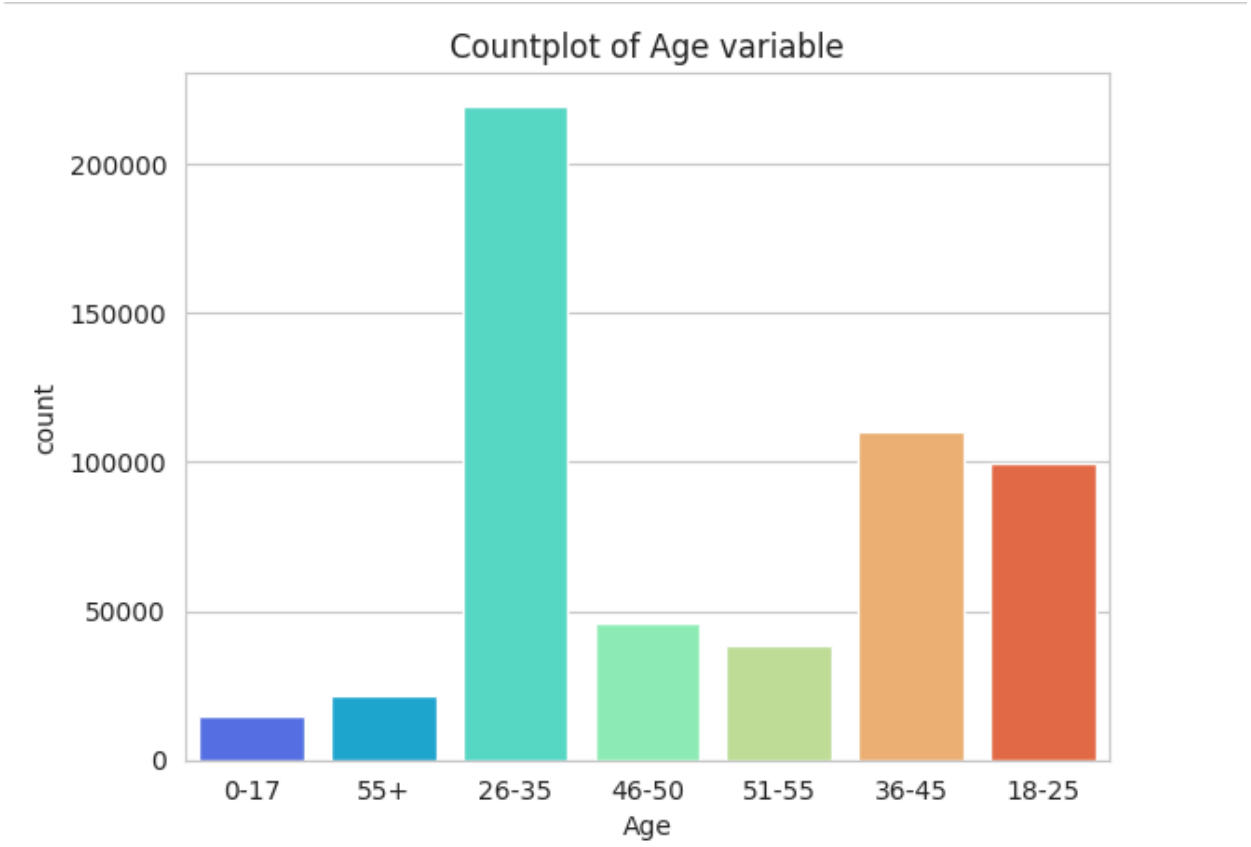
Figure 11: Null values along test dataset



variables distribution key findings:

We have found that customers with age ranging from 26-35 are the more frequent in Black Fridays than customers with other ages.

Figure 12: Age distribution in train dataset:

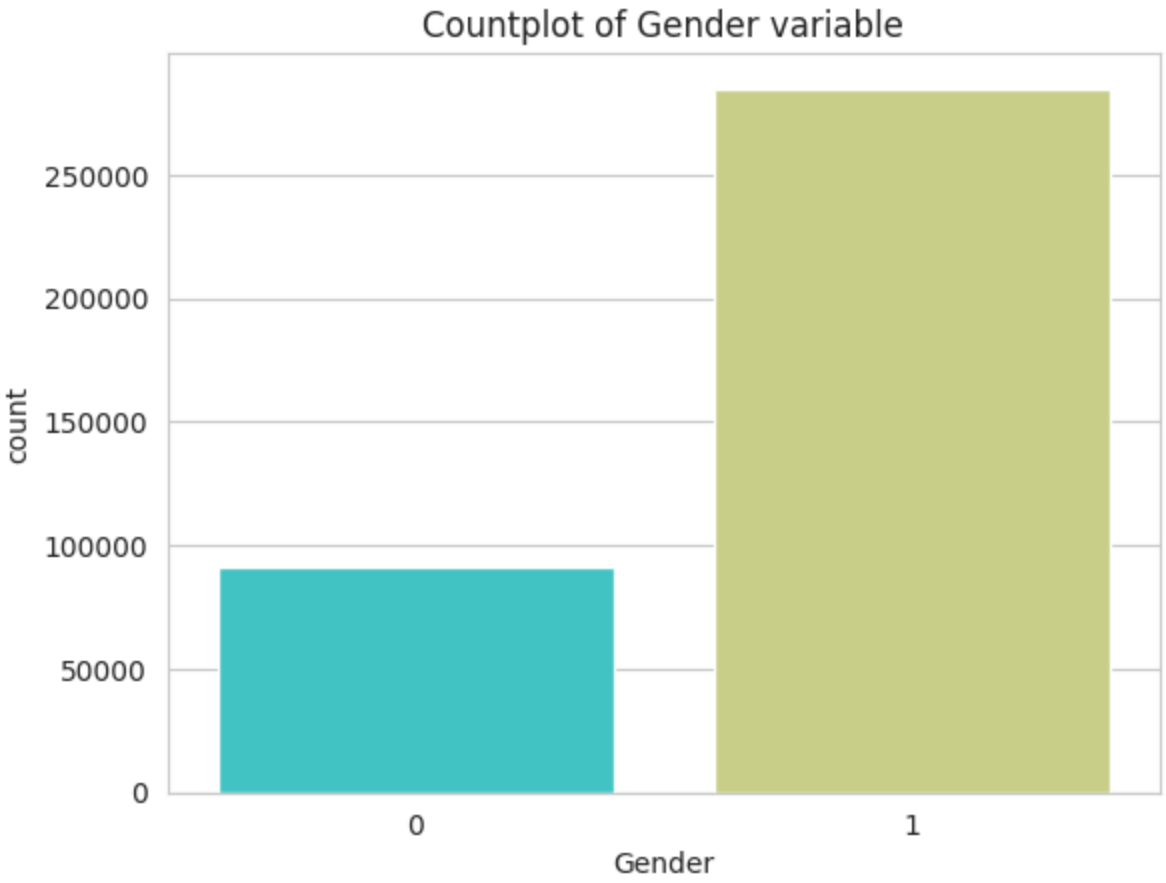


CODE SNIPPET:

```
sns.set_style('whitegrid')
sns.countplot(x='Age',data=dados,palette='rainbow').set_title('Countplot of Age variable')
plt.show()
```

Another important finding is that Male Customers are the most frequent and the majority of the customer sin Black Friday.

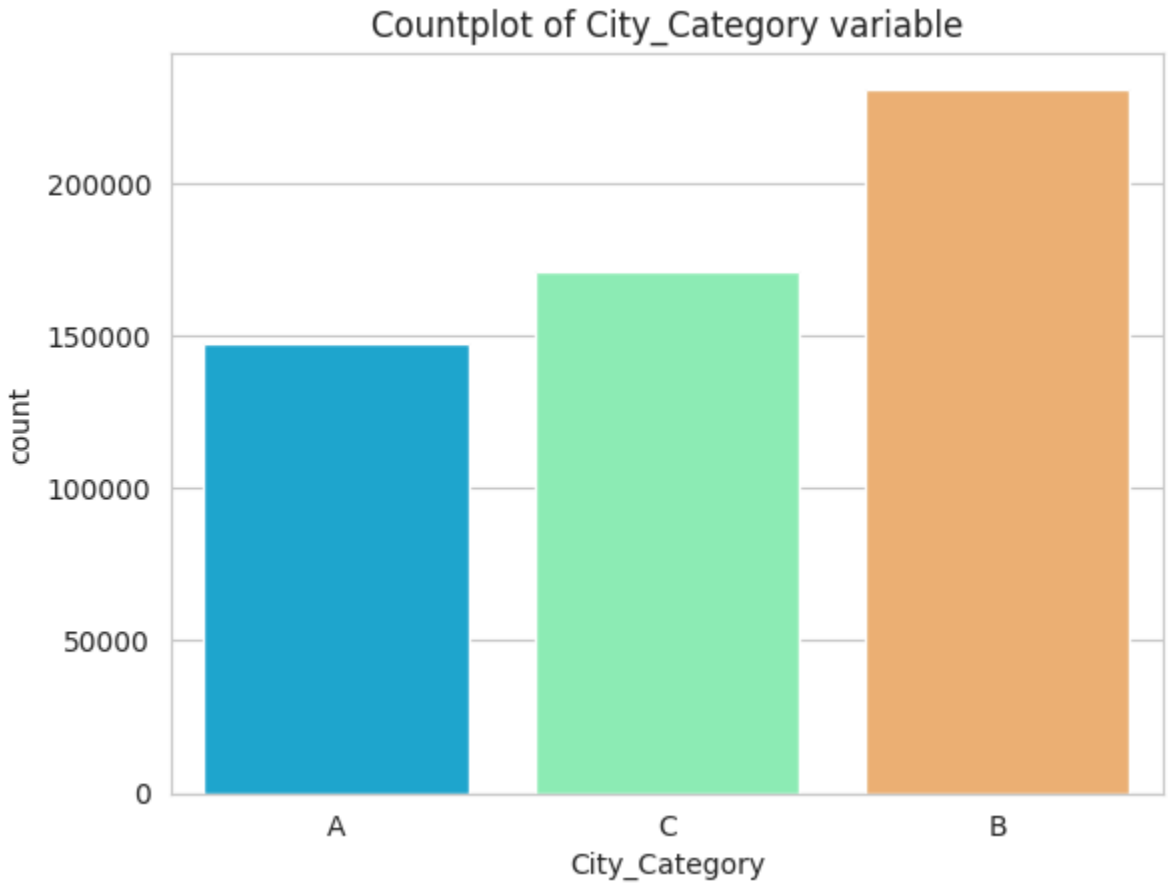
FIgure 13: Customer Gender distribution



CODE SNIPPET:

```
sns.set_style('whitegrid')
sns.countplot(x='Gender',data=train_encoded,palette='rainbow').set_title('Countplot of Gender variable')
plt.show()
```

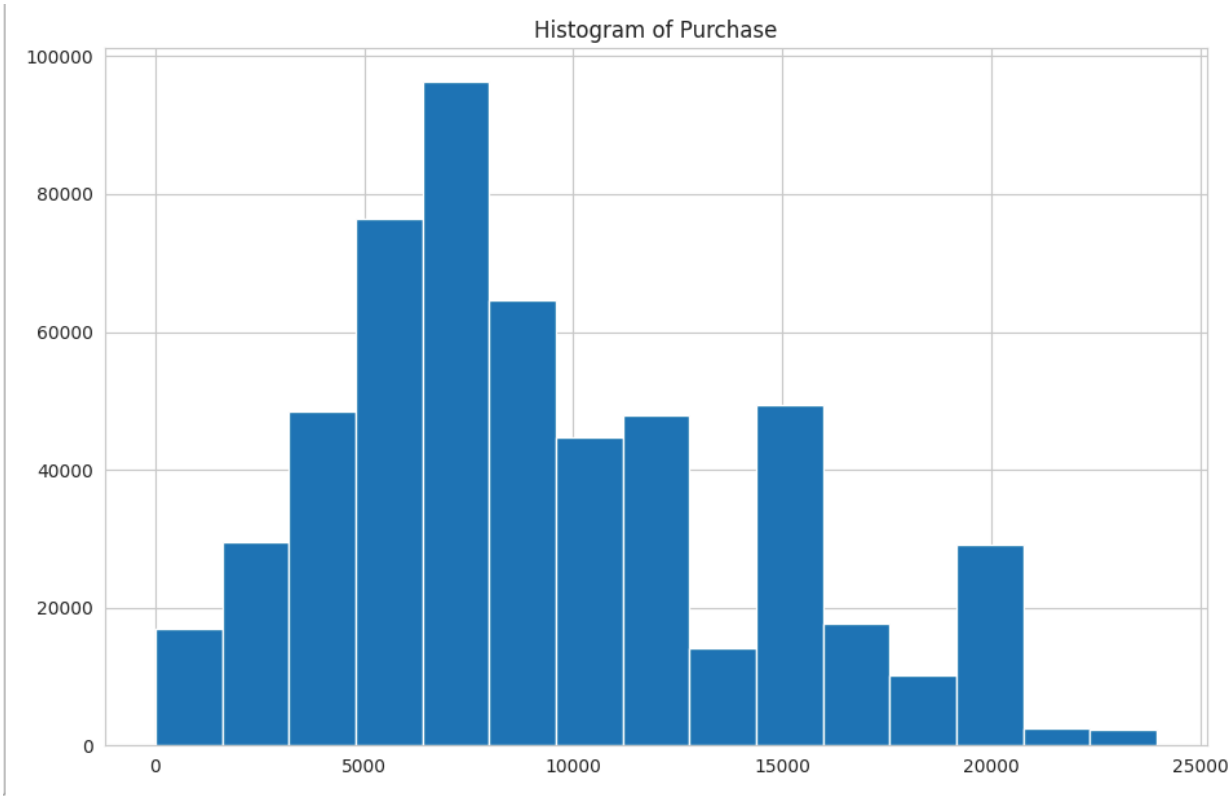
Figure 14: Another finding is that Customers from city category B are the most frequent in Black Fridays.




```
sns.set_style('whitegrid')
sns.countplot(x='City_Category',data=dados,palette='rainbow').set_title('Countplot of City_Category variable')
plt.show()
```

We found out that Purchase distribution is very irregular, with most frequent purchases lying between 5000 and 10000.

Figure 15: Histogram of Purchases




```
plt.figure(figsize=(11,7))
plt.title("Histogram of Purchase")
hist = dados['Purchase'].hist(bins=15)
#ax.set_xticklabels(ax.get_xticklabels(), rotation=30, ha="right")
#plt.tight_layout()
plt.show()
```

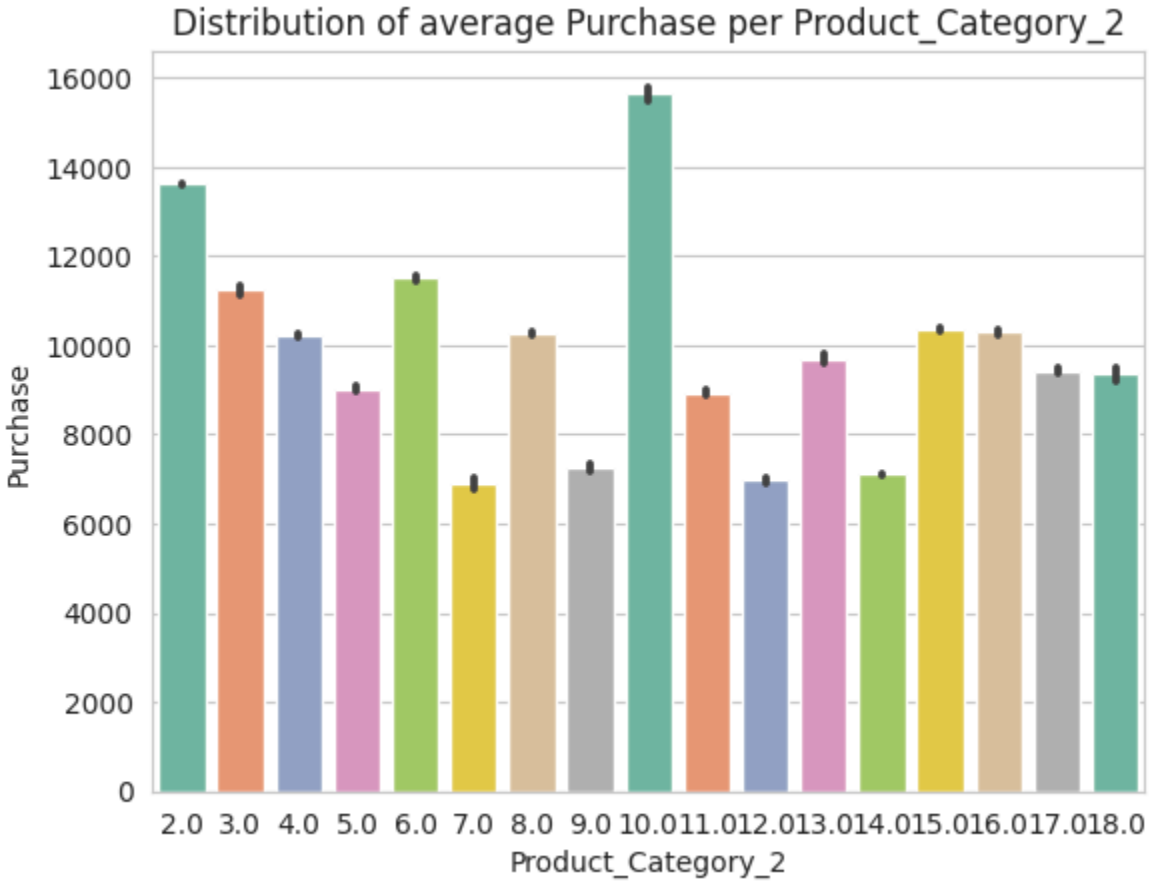
Another finding is that we have not found any important outliers in Purchase distribution. So that no additional information was discarded in the datasets because or Purchase outliers.

Figure 16: Boxplot of Purchase in train dataset

```
fig = plt.figure(figsize =(10, 7))
plt.boxplot(train_encoded['Purchase'])
plt.show()
```

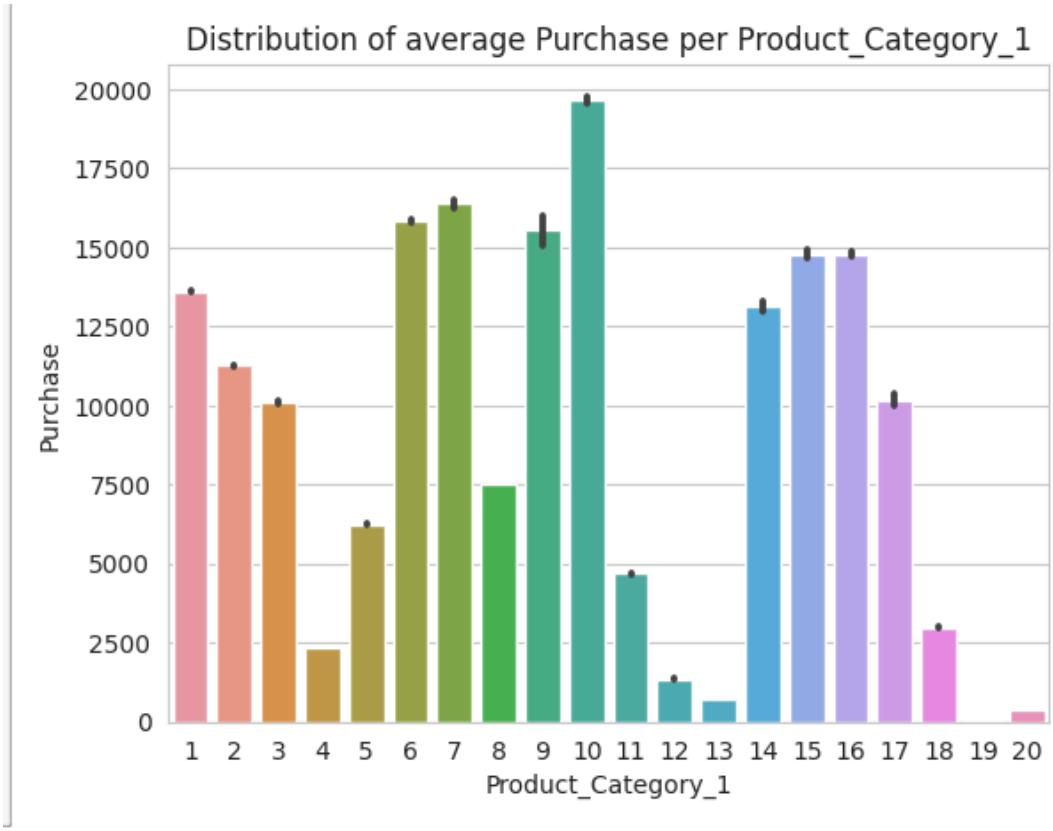
Analyzing the distribution of the average purchase per distinct Product_Category_2 values, we found out that the highest average purchases were verified in categories 10.0, 2.0 and 6.0.

Figure 17: Distribution of Average Purchase per Product_Category_2 values



Analyzing the distribution of the average purchase per distinct Product_Category_1 values, we found out that the highest average purchases were verified in categories 6, 7, 9 and 10 categories.

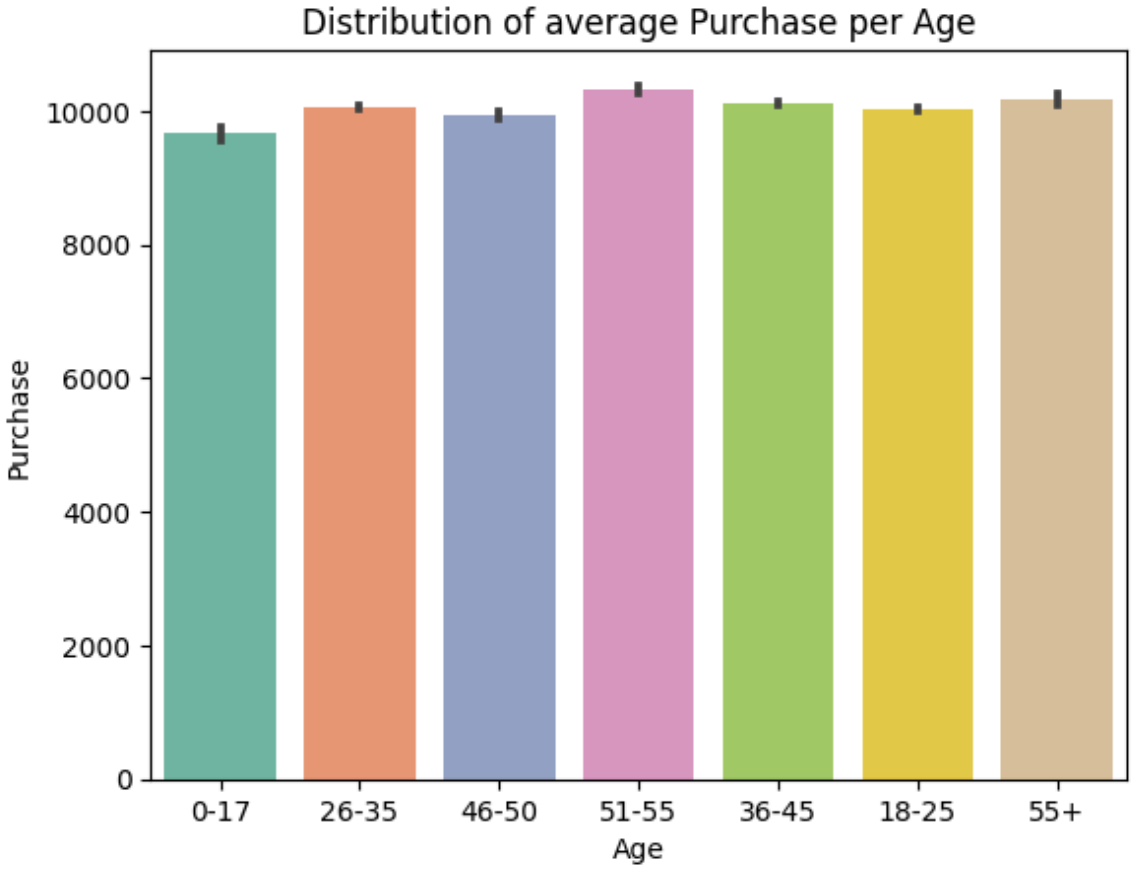
Figure 18: Distribution of Average Purchase per Product_Category_1 categories



One can see that highest average purchases occur for product_category_1 equal to 10, 9, 7 and 6.

We have also analyzed the average purchase distribution per age ranges. We found out that the highest average purchase wawa verified in the 51-55 age range.

Figure 19: Distribution of Average Purchase per Age



Feature engineering

The feature engineering process for the demo included eliminating the *Product_Category_3* feature due to excessive missing values and removing any remaining missing data from both train and test datasets. Categorical variables were label encoded to prepare them for use in the machine learning model. Features selected for the model included customer demographics and product categories, with decisions based on correlation analysis and the relevance of these features to predicting Black Friday sales behavior. The rationale behind feature selection was to retain variables with meaningful relationships to the target variable, purchase amount, while ensuring data quality through the elimination of incomplete and irrelevant data.

The feature engineering steps implemented in the demo 2 were:

- Elimination of feature Product_Category_3 in train and test datasets,
- Elimination of remaining missing values in the two datasets,
- Label encoding of the categorical variables.

Data Security and Privacy in Cloud Storage

Regarding Security and Privacy for the data used in this demo it should be pointed out that, first of all that data lies in a Cloud Storage Bucket within a specific project linked to a specific service account. Only people with proper IAM credentials can access the project and the dataset.

Secondly, the dataset is public and contains no sensitive information, because all customers are represented by IDs, so that no necessary data encryption was necessary.

Data Preprocessing and Final decisions regarding data strategies to be adopted on this use case

All preprocessing steps corresponded to the Feature Engineering steps mentioned in previous sections (It was not necessary to make any type of train-test split because Kagle already provided separated train and test datasets).

There was no need to develop a Train/ test split strategy because As mentioned in previous section, the Black Friday dataset is already provided in two separate datasets: train and test. The validation sets were composed in the context of the 4-fold Cross Validation schemas implemented during train/ validation of the predictive models.

Machine learning model design(s) and selection

For Demo #2, Random Forest Regression models were selected due to their strong predictive performance across industries and their interpretability for non-technical stakeholders. The criteria for model selection was the explained variance score in the

cross validation schemas, using a 4-fold Cross-validation alongside hyperparameter tuning using GridSearchCV, testing different numbers of decision trees (5, 10, and 15). The number of folds used in the Cross-validation was chosen basically because of the previous experience with such models, and squared error and explained variance were used as performance metrics, with a threshold of 95% for explained variance to prevent overfitting. Code snippets demonstrate the training, validation, and model selection process.

Proposed Machine Learning Model

For this Demo 2 development,we have decided to use Random Forest Regression models.

The reason for choosing such models are:

- Random Forests provide some of the best performing predictions models in academy and along different industries as whole,
- They are easy to interpret end to explain to non technical personnel,

Specifically for this Demo, we have decided to use regression Random Forest models, because of the nature of the variable which we desire to make predictions on: the total purchases made by the different customers, for each of the different products involved in the Black Friday events.

Used Libraries

For the purposes of the development of a machine learning solution for the business problem at stake, we chose to utilize scikit-learn library for Python, which provides many options for model training, evaluation and testing. For data visualization we have used Malplotlib, and seaborn libraries.

Finally, for exporting final model artifacts we have used joblib library (we have also used pickle library but in model deployment it is used exclusively by the joblib).

Model selection

Different Random Forest regression models were trained and evaluated using Cross validation in the train dataset provided by Kagle.

For the training validation purposes, we have selected 4 folds from the original train dataset.

The cross validation expedients were combined with hyperparameters combinations (using scikit-learn`s GridSearchCV functionality).

As we can see from the code snippet below, the grid search schema adopted in the cross validation section along training considered different numbers of Decision Trees for each Random Forest model being considered: models with 5, 10 and 15 decision trees were considered.

We have kept as criterion, the squared error and as scoring metric the explained variance.

For all considered models, bootstrapping were kept.

The choice of these hyperparameters were made based on previous experience on similar data, using such models. GridSearchCV values were also defined in order to get final model's explained variance and R2 metrics under 95% as an adopted threshold to indicate model overfitting.

Below we present code evidence about the train-validation schema adopted in this Demo.

CODE SNIPPET:

```
[25]: regressor = RandomForestRegressor(random_state=0)

[55]: grid_param = {
          'n_estimators': [5, 10, 15],
          'criterion': ['squared_error'],
          'bootstrap': [True]
      }

[56]: gd_sr = GridSearchCV(estimator=regressor,
                           param_grid=grid_param,
                           scoring='explained_variance',
                           cv=4,
                           n_jobs=-1)

[57]: gd_sr.fit(X_train,y_train.values.ravel())
```

The best performing model in the train validation session was a Random Forest made of 15 decision trees, as depicted in the code snippet below:

CODE SNIPPET:

```
[59]: gd_sr.best_estimator_

[59]: RandomForestRegressor
      RandomForestRegressor(n_estimators=15, random_state=0)
```

Machine learning model training and development

In this demo, model training was conducted using Vertex AI Workbench with dataset sampling provided by Kaggle, avoiding the need for any splitting method. Adherence to Google Cloud's best practices was ensured through the use of Vertex AI for distributed training, appropriate resource allocation, and monitoring. The explained variance metric was chosen for model evaluation due to its ability to measure how well the model fits the

data while controlling overfitting, which is critical for predicting purchases during Black Friday events. Hyperparameter tuning was performed using GridSearchCV, optimizing the number of decision trees to balance bias and variance. Bias/variance tradeoffs were carefully managed by adopting the threshold value for explained variance of 95%, to decide whether a given model overfitted the data or not. The Model evaluation metric used in the cross validation scoring, as explained before, was the explained variance. This choice is justified because we are considering a model that predicts well the purchases made in Black Friday events, but at the same time, a model that generalizes well on new datasets.

The explained variance (in a similar fashion as to the R2 coefficient) provides a straightforward way to get information about how well the model fitted the data.

As an adopted criteria, it was adopted the threshold value for explained variance of 95%, to decide whether a given model overfitted the data or not, meaning that models with 95% or greater explained variance will be considered overfitted. This metric was considered optimal for the case at hand because of its direct interpretation regarding model adjustment to data and to control model bias-variance tradeoff.

By these adopted criterias, we should seek for a final model that would explain about 90% of data variance, but not much higher than that alone.

Another reason why we have decided to adopt explained variance for scoring the models along GridSearchCV was its explainability, and immediate interpretability.

Accordingly, we have used the scores obtained with the explained variances computed, along the cross validation, to control the bias-variance tradeoff. Given that the best estimator from Grid Search Cross Validation is the one with the highest cross validation score, we controlled the values in the Grid Search so that the final model did not have explained variance exceeding 95%. By acting this way, we have controlled the bias.

Hyperparameters tuning and training configuration

As explained in the previous section final model hyperparameters were defined from a Grid Search CV proceedment. we have decided to train different models using different numbers of base estimators (decision trees) , considering models with 5, 10 and 15 decisions trees.

The best performing model in the train validation session was a Random Forest made of 15 decision trees, which was the one with the highest cross validation score (in the case of this development, the one with highest explained variance).

It`s worth mentioning that we had to consider combinations os a few hyperparameters (only the number of decision trees) in the GridSearchCV procedures because the time and resource constraints existent along the development of this demo.

It is recommended to consider combinations of other hyperparameters values in other future developments (in the case of Random Forest regressors, it could be included for example, the maximum number of features considered for splitting a node, the maximum number of levels in each decision tree, and so on). Because of time and resources constraints, as explained before, we considered only different models, varying only regarding the number of decision trees.

The training configuration used all available data in the train.csv file made available by Kagle, after excluding feature Product_Category_3 and after that, all remaining rows containing missing values.

The scoring metric for Cross Validation scoring strategy was the explained variance.

Dataset sampling for model training, validation and test

For this particular demo, there was no necessity of implementing any type of splitting an initial dataset into train and test sets because Kagle already provided separated train and test datasets.

Once the exclusion of feature Product_Category_3 and of the remaining rows containing missing values, all data rows in train dataset were considered (under a 4-fold cross validation strategy) in the training validation process.

Same logic follows to the test dataset.

Adherence to Google's Machine Learning Best Practices

As presented along this document, we followed Google's Machine Learning Best Practices in the planning and implementation of all the workflow depicted in this document. In some (few) cases it was not possible to follow some Machine Learning Best Practices documentation's suggestions either by time or costs constraints; but for the very most part the implemented workflow followed this Best Practices documentation (available in the link: https://cloud.google.com/architecture/ml-on-gcp-best-practices)

First of all it must be said that, we used, in each step of the Machine Learning models developments, the products recommended by Google's ML Best Practices (Link: https://cloud.google.com/architecture/ml-on-gcp-best-practices?hl=pt-br#use-recommended-tools-and-products):

- Configuration of ML environment step: we used instances of Vertex AI Workbench for this step.
- Machine Learning Development step: we used the following products in this step:

  Cloud Storage and instances of Vertex AI Workbench.Specifically, we used Vertex AI Workbench instances for experimentation and development of models .

- Data processing steps: we used Pandas Dataframes in Vertex Workbench instances mostly for feature engineering and general data manipulation.
- Operational training: we exported scikit learn models as joblib artifacts, to deployment in Google Cloud endpoint.
- Artifacts organization: we used Google's Vertex AI Model Registry solution.

We stored resources (like files containing test data) and model artifacts (joblib model artifacts) in Cloud Storage. These artifacts and resources are stored in Cloud Storage associated within a specific project where only allowed people can have access to. Additional Identity Access Management (IAM) prerogatives can be set for each user.

- For Machine Learning Environments: we used personalized models (that is, trained in a customized way, with proper code) using Vertex AI environment.

Besides adopting the above recommended products as ML Best Practices, we also followed the Best Practices below.

- Data Preparation for training

  Observing Google Machine Learning Best Practices, training data were extracted from origin/ data sources and converted to appropriate format for machine learning training purposes (this was accomplished in feature engineering phase). Final data was stored in appropriate Google Cloud Resources (Cloud Storage bucket).

- Avoid to store data in block storaging:

  We have not stored any data in block storaging style (like files in networks, or hard disks). Instead we used Cloud Storage.

  We also have not read any data directly from any specific database other than Cloud Storage for optimal performance.

- Maximize model`s predictive precision with hyperparameters adjustments

  This was done in Demo 2.

- Prepare models artifacts to be available in Cloud Storage:

This was accomplished in Demo 1 where models`s artifacts were made available in joblib format (model.joblib files) .

- Specify the number of cores and machine specifications for each project

We have defined appropriate machines (in terms of number of cores, memory and even GPU`s to be used in each Demo base on previous experience training models for each demo and also taking into consideration the dataset size).

- Plan the model data entries:

We have planned how input, new test data are to be transmitted to trained final models. So that we judged that for batch predictions, for example, input data are to be stored and made available for models from Cloud Storage, for the demo.

Finally, it must be said that all models were deployed to an endpoint and containerized, using Vertex default options, with default Google Cloud containers, in Vertex Model Registry.

As stated previously we have not followed some of the Best Practices suggestions, either because of time constraints or other factors.

So that for example, for this demo, we have not used Docker containerized models specially because of time constraints for developing this demo. Instead we used default containerization available in Google Cloud.

Another example of a point in Best practices that was not followed is the use of Datasets in Vertex.This was because the dataset was very small and did not require a proper dataset. Besides, the Demo 2 instructions required specifically that data to be used in model training and testing should be read from Cloud Storage.

3.1.3.7 Machine Learning Model Evaluation/ Model Performance Assessment

We have selected the MSE (mean squared error) metric as the main one to assess model performance on the training validation and test sets.

This is due to the fact that the MSE gives more importance to large model errors, being the variance of the model residuals. Also, it is a key/ standard metric to tackle model performance both in academia as well in companies as a whole.

Other metrics, such as the Mean Absolute Error (MAE), determination coefficient (R2), and the Root Mean Squared Error (RMSE) were also used in the python notebook.

As depicted in section 3.1.3.5.3, these metrics were used to make an assessment of final model's performance on validation and test sets.

We also used the same metrics to evaluate previous trained models in the provided notebooks for this Demo 1 (CHICAGO_TAXI_TRIPS_MODEL_TRAINNING_EDITED_FINAL.ipynb and EDITABLE_MODEL_APPLICATION_DEMO1.ipynb notebooks).

Machine Learning Model Evaluation/ Model Performance Assessment

Post-training, it was not possible to evaluate the final model's performance on an independent test dataset—reflecting real-world data distributions—due to the missing target variable (purchase data) in it. Nevertheless all possible actions we could take to maximize the model's potential to generalize well on unseen data were accomplished, for example by hyperparameter tuning.

Once the final model was selected from the Grid Search Cross Validation, its overall performance on the training set was assessed according to a given set of different regression metrics: explained variance, R2 coefficient, MSE, and MAE.

It was not possible to compute these performance metrics on the provided test set, because this dataset, as provided by Kagle, did not include observed purchases.

Below, we present the resulting regression metrics on training set for the final model:

CODE SNIPPET:

```
import sklearn.metrics as metrics
def regression_results(y_train, y_train_pred):

    # Regression metrics
    explained_variance=metrics.explained_variance_score(y_train, y_train_pred)
    mean_absolute_error=metrics.mean_absolute_error(y_train, y_train_pred)
    mse=metrics.mean_squared_error(y_train, y_train_pred)
    #mean_squared_log_error=metrics.mean_squared_log_error(y_train, y_train_pred)
    median_absolute_error=metrics.median_absolute_error(y_train, y_train_pred)
    r2=metrics.r2_score(y_train, y_train_pred)

    print('explained_variance: ', round(explained_variance,4))
    #print('mean_squared_log_error: ', round(mean_squared_log_error,4))
    print('r2: ', round(r2,4))
    print('MAE: ', round(mean_absolute_error,4))
    print('MSE: ', round(mse,4))
    print('RMSE: ', round(np.sqrt(mse),4))
```

```
regression_results(y_train, y_train_pred)
```

```
explained_variance:  0.9447
r2:  0.9447
MAE:  858.7029
MSE:  1487538.9948
RMSE:  1219.6471
```

Fairness Analysis

A profit maximization model trained on the Black Friday dataset for targeted marketing may introduce biases, particularly when using purchaser demographics such as age, gender, or location. To determine if the model has biases, one approach is to test it using fairness indicators,

which compare model performance across different demographic groups. For example, comparing how the model predicts purchases with and without demographic features can help identify any disparities in predictions. If significant differences arise, this could suggest the model is biased toward certain groups. To mitigate these biases, demographic and location features could be removed from the model, ensuring predictions are not skewed by factors like race or income. Alternatively, tools like mindiff could be used to equalize profit predictions across demographic characteristics, promoting fairer outcomes across customer segments. This ensures the model's recommendations for marketing and pricing strategies are equitable, reducing the risk of reinforcing existing social or economic inequalities.

In this section we will discuss possible implications of possible existent statistical bias in the final model as well as fairness-type of bias and profit maximization implications by the company when seeking profit maximizations based on model predictions.

From the statistical point of view, given that train dataset is relatively small and that the provided test dataset does not have observations for the target variable (purchases), it is possible that one of the following situations may happen:

- The train dataset is made of observations collected in a specific time period, with atypical influence of other uncontrollable factors (that can influence customers expectations about the future and its current/ observed consumption behavior, like political/ geopolitical scenarios, and so on). If this is the case, the consumption behavior observed in the train dataset is not representative of the purchases that would otherwise be observed outside of this atypical period.
- The train dataset was collected in a given time period with specific cyclic and seasonal influences, which may not be valid/ representative of other time periods
- The dataset may be a result of an incorrect sampling process and do not contain information representative of the contained features and the different consumption patterns.

In these situations, the resulting predictive model will be biased from a statistical point of view, given that train dataset is not representative of the different consumption behaviors regarding the different products made available by the company promoting Black Friday events.

In order to check for solving these issues, we cloud advise some of the following actions:

a) To use business experience and other similar companies experiences/ data to check if the data collection represents general observed purchase behaviors or if it is being influenced by specific factors (like political/ geopolitical) that may influence these behaviors. If such factors are identified it is advisable, for example, to retrain new models with new data not being influenced by such factors.
b) To check/ guarantee that data collected for training does not contain data from specific parts of the year, but that instead, collected data is representative of all possible consumption patterns along the different phases of business cycle and seasonal influences. This can be done using business experience.

c) All data was gathered for training or at least a representative sample was made available when constructing the provided train and test datasets for this demo. For that, it may necessary to implement more complex sampling strategies (for example sampling by conglomerates and then by random samplig) in order to have a representative sample for model training. Applying these sampling techniques/ strategies and training new models on these new datasets (and assessing their performances) and comparing their results with the final model performance (trained with provided train dataset by Kagle), can give insights if the final model is biased because of incorrect sampling strategies applied in the construction of the provided train dataset.

Most of these possible biases can be identified in testing final model with other datasets, which should include the target variable, of observed purchases, on other to detect and calculate model bias and variance along these different datasets,

We cloud tackle as well another aspect of bias related to social income distribution and social inclusiveness and to the use of the model predictions to provide a basis for planning/ actions aiming profit maximization: it may the case that the company organizing the Black Friday events discover important consumption behaviors that would reveal important economic and social aspects by part of its customers, that can reveal some important social/ economic problems related to some of their customers.

These discoveries might trigger the concern by part of the company of improving some special conditions of its customers, and this might trigger new considerations by part part of the company in terms of possible actions it might take regarding these special customer segments.

One of these possible actions is to build special pricing/ price policies or marketing campaigns for these special customer/products segments, so that the company's profit maximization strategies would have important inputs from the consideration of the knowledge obtained from the data, about these segments.

So in this sense, if the company had not identified such customers segments and had not decided to reshape its profit maximization strategies in order to try to improve/ positively impact the social/ economic situation of some of its customer segments, the model would be biased in the sense that it would the company to somewhat contribute for the continuation of the negative situation of some of its customers.

For these type of bias, the first step for company to try to reduce it, is to detect from data, insights that would lead to such discoveries; and, once these insights are obtained, to develop proper marketing/ pricing policies, based on model predictions, to improve life quality for these customer segments.

A profit maximization model trained on the Black Friday dataset for targeted marketing may introduce biases, particularly when using purchaser demographics such as age, gender, or location. To determine if the model has biases, one approach is to test it using fairness indicators, which compare model performance across different demographic groups. For example, comparing how the model predicts purchases with and without demographic features can help identify any disparities in predictions. If significant differences arise, this could suggest the model is biased toward certain groups. To mitigate these biases, demographic and location features could be removed from the model, ensuring

predictions are not skewed by factors like race or income. Alternatively, tools like mindiff could be used to equalize profit predictions across demographic characteristics, promoting fairer outcomes across customer segments. This ensures the model's recommendations for marketing and pricing strategies are equitable, reducing the risk of reinforcing existing social or economic inequalities.

Final considerations and recommendations for future developments

This document presented a whole workflow, from exploratory analysis top model deployment, for delivering a Machine Learning solution that could deliver predictions that could give input for developing specific planning and actions in order to explore opportunities for improving company's results/ profits.

The final predictive model presented very good predictive performance on train dataset.

It is recommended to train additional models with more hyperparameters combinations or even with other predictive models lineages (like neural networks) .

Lessons Learned

We learned that very good insights could be extracted from dedicated exploratory analysis of the Black Friday dataset, so that we recommend that a continuation of the data analysis exposed here be continued in search of any additional good business insight.