REPORT

EXECUTIVE SUMMARY

The ConsultingCorp report prepared for FoodCorp explores churn in the context of fast-moving consumer goods retailers. Churn is defined as a customer leaving a company or becoming less loyal to it. The report proposes a methodology for defining churn based on a length of inactivity in days, and measures the percentage of customers who would be considered churners under a perfect predictor. Based on the analysis, if the period of unobserved activity is greater than 64 days, FoodCorp can expect to target 14.85% of active customers with a perfect classifier. This report provides data descriptions of the customers, products, receipts, and stores used for analysis. The data range for 6377 customers is July 27, 2020 to March 21, 2022. The analysis was conducted using the Databricks platform.

The Churn Prediction System predicts customer churn based on selected features. The chosen features are 'value', 'churn_status', and 'day', which are stored in a table called 'df_table'. A sliding window approach is used to capture historical behavior, and lagged versions of churn_status are used as input features. The target variable, representing the churn status for the current day, is created along with temporal features such as cs_28, cs_64, cs_128, woy, wom, and total_value.

The final model selected was the LightGBM classifier, which is based on the gradient boosting decision tree (GBDT) algorithm. The article provides a detailed description of the steps taken to refine the logistic regression model, which was used as a baseline for comparison with the LightGBM classifier. The logistic regression model achieved an accuracy of 84%, but the LightGBM classifier achieved an accuracy of 87% and was chosen as the final model.

The report describes the steps taken to develop the LightGBM classifier model, starting with defining a hyperparameter grid for tuning the model. The optimal set of hyperparameters was found using the GridSearchCV method, which performs an exhaustive search over the hyperparameter grid. The report also provides details on the hyperparameters selected, including the learning rate, maximum depth, feature fraction, subsample, and number of estimators.

After the hyperparameters were identified, the LightGBM model was trained on the entire training set and used to make predictions on the testing set. The article provides details on the precision and recall scores for the positive and negative classes, indicating that the model correctly identified 88% of the positive cases and 87% of all predictions. The most important features for predicting purchases were customer_id, day, woy, total_value, cs_128, cs_64, cs_28, and wom.

3299 active customers are at risk of leaving. Active customers are health-conscious and frequently purchase fruits, vegetables, milk, bread, and confectionery, while churned customers purchase alcohol, cooked meat, ready-to-eat food, and cigarettes. They are also PayPoint users and likely working bachelors who enjoy partying. To prevent further churn, the company can provide excellent customer service, offer loyalty programs, create personalized marketing plans, and actively listen to feedback. These strategies can help retain customers and improve their overall experience with the brand.

CURRENT LEVELS OF CHURN

The report prepared by ConsultingCorp for FoodCorp discusses the concept of churn in the context of fast-moving consumer goods retailers. Churn is defined as a customer leaving a company or becoming less loyal to it, and FoodCorp is interested in defining churn in a data-driven way to identify customers who may require interventions to retain them. The report proposes a methodology for determining a global definition of churn, which involves defining a length of inactivity in days that is unacceptable to the company, and using this to define churn as the period of unobserved activity greater than this length. The report then describes how to choose the length of inactivity and measures the percentage of customers who would be considered churners under a perfect predictor.

Figure 1 shows the distribution of time between customer visits, with 75% of customers not considered to have churned if the inactivity period is set to 55 days. Figure 2 shows the percentage of customers who have churned based on an increasing inactivity period, with 9.54% predicted to have churned if the period is set to at least 65 days. Figure 3 provides a summary table of the results for the churn definition in Figure 1, including the number of active customers predicted to churn based on the percentage computed in Figure 2. Based on the interpretation of Table 3, it appears that the definition of choosing the churn threshold of 64 days seems optimal. This is due to the fact that 74.90% of customers median days between visits is less than this and FoodCorp should expect to target 14.85% of active people with a perfect classifier. Choosing this definition would help the company save marketing costs but it also means that they should have a more personalised targeting plan.

DATA DESCRIPTION

Table Attribute	Data Type	Data Description
Table: customers		
customer_id	integer	Unique ID of the customer
first	string	First name of the customer
last	string	Last name of the customer
dob	date	Date of birth of the customer
Table: products		
product_code	integer	Unique ID of the product
product_details	string	Description of the product
department_code	integer	Unique ID of the department to which the product belongs
department_name	string	Name of the department to which the product belongs
category_code	integer	Unique ID of the category to which the product belongs
category_details	string	Description of the category to which the product belongs
sub_category_code	integer	Unique ID of the sub-category to which the product belongs
sub_category_details	string	Description of the sub-category to which the product belongs
Table: receipt_lines		
receipt_line_id	integer	Unique ID of the receipt line
receipt_id	integer	Unique ID of the receipt to which the line belongs
qty	integer	Quantity of the product purchased
value	decimal	Value of the product purchased
Table: receipts		
receipt_id	integer	Unique ID of the receipt to which the line belongs
purchased_at	datetime	Date and time when the receipt was purchased
customer_id	integer	Unique ID of the customer who made the purchase
store_code	integer	Unique ID of the store where the purchase was made
Table: stores		
store_code	integer	Unique ID of the store
address	string	Address of the store
postcode	string	Postal code of the store

lat	decimal	Latitude of the store
lng	decimal	Longitude of the store

No of customers: 6377

Date Range: 2020-07-27 to 2022-03-21

Databricks platform has been used for further analysis.

CHURN PREDICTION SYSTEM

1. FEATURES

The Features have been chosen considering the relevance of the features to the problem statement in a marketing context. The chosen features – 'value', 'churn_status', 'day' are stored in the table 'df_table' which is then used for further analysis. The features 'purchase_made', 'purchased_at',' calendar_date' and 'counter' were dropped as they were used to generated the chosen features.

This diagram describes the windowing strategy used in this code. It is a sliding window approach, where lagged versions of churn_status are used to capture historical behaviour, and the woy and wom features are computed based on the current day. The input and output features are defined in a temporally consistent way, with historical features lagged by a fixed number of days and the target variable based on the current day.

Table called all_ml_data_pts is created by selecting and transforming data from an existing table called df_table. The goal of this transformation is to create temporal features for predicting customer churn. First, a common table expression (CTE) called churn_value_ts is defined to group the original data by day and customer_id, and compute the maximum churn_status and total value for each group.Then, the CTE is used as input to the outer SELECT statement, which further transforms the data by creating temporal features. Specifically, the following features are created:

- to_predict: The target variable, representing the churn status for the current day.
- cs_28, cs_64, cs_128: Lagged versions of churn_status for 28, 64, and 128 days ago, respectively. These are used as input features to capture the historical churn behavior of each customer.
- woy: Week of year, computed as day % 364, where 364 is the number of days in a non-leap year.
- wom: Week of month, computed as day % 30, where 30 is an approximation of the average number of days in a month.

- total_value: The total value for the current day, computed in the CTE.

Based on the initial definition, it was found that a week's time is not enough to see a noticeable change in the churn status. A meaningful model can only be successfully created if a noticeable change in the churn status appears, Thus, a LAG value of 28 and above have been used instead of 1,7 and 30.

2. PREDICTION APPROACH

Data was already cleaned and hence no data cleaning has been done.

Final Chosen Model: LightGBM Classifier.

LightGBM is an open-source gradient boosting framework that is designed to be efficient, scalable, and highly customizable. It was developed by Microsoft and is widely used for a variety of machine learning tasks, including classification, regression, and ranking. At its core, LightGBM is based on the gradient boosting decision tree (GBDT) algorithm. This algorithm involves iteratively adding decision trees to the model, with each subsequent tree attempting to correct the errors of the previous trees.

Steps leading to the final model:

Logistic Regression:

First, the code creates a baseline by assigning the values of the 'cs_28' column in the validation set (X_valid) to the 'baseline' variable. This will serve as a comparison to evaluate the performance of the logistic regression model. Next, the code imports the LogisticRegression model from the scikit-learn library and creates an instance of it called 'lr'.Then, the model is trained on the training set (X_train and y_train) using the 'fit' method of the LogisticRegression instance.Once the model is trained, it is used to predict the labels of the validation set using the 'predict' method of the LogisticRegression instance, and the predicted labels are assigned to the 'y_pred' variable.

The logistic regression model achieved an overall accuracy of 0.84 on the validation set, which means it correctly predicted 84% of the instances in the dataset. Looking at the classification report, we can see that the model performed better in predicting instances of the '1' class than the '0' class. The model achieved a precision of 0.86, recall of 0.89, and an f1-score of 0.88 for the '1' class, indicating that it correctly identified 89% of the positive instances and had a low false positive rate of 14%. However, for the '0' class, the model achieved a precision of 0.79, recall of 0.75, and an f1-score of 0.77, indicating that it correctly identified 75% of the negative instances and had a higher false positive rate of 25%.

Refining the Logistic Regression Model:

The training data is selected from a range of days historically before the validation and test data points. The code starts by defining the reference days for the test, validation, and training sets. It then defines a function called get_temporal_X_y which selects data from the database for a given range of days. Once the datasets are ready, the logistic regression model is trained using the training data. The trained model is then used to predict the target variable for the validation set. The accuracy of the model's predictions is evaluated using the accuracy_score function from sklearn.metrics library. Finally, the accuracy score is printed to the console.

The accuracy score obtained is 0.88, indicating that the model's predictions are correct 88% of the time. This score can be used to evaluate the performance of the model and make any necessary adjustments to improve its accuracy.

Cross-validation is a technique used in machine learning to evaluate the performance of a model by partitioning the available data into training and testing sets. The goal is to estimate how well the model will perform on unseen data, and to avoid overfitting to the training data. Even cross validation was tried which returned the same result.

LightGBM Classifier along with Hyperparameter Tuning:

The GridSearchCV technique involves exhaustively searching over a specified parameter grid for the hyperparameters that yield the best cross-validation performance. The search is typically done by evaluating the model with a certain performance metric (such as accuracy or F1 score) on a training set, which is then compared to a validation set. This process is repeated for all combinations of hyperparameters in the grid.

The first step is to define a hyperparameter grid that will be used for tuning the LightGBM model. The grid includes values for learning rate, maximum depth, feature fraction, subsample, and number of estimators. Next, a GridSearchCV object is created, which performs an exhaustive search over the hyperparameter grid to find the optimal set of hyperparameters for the LightGBM model. The cross-validation parameter cv=all_cv_indexes indicates that all available folds should be used to evaluate the performance of the model.

The best hyperparameters found by GridSearchCV are printed out along with the mean test scores for the hyperparameters. (

Mean lg: -0.8820762113846636

{'feature_fraction': 0.7, 'learning_rate': 0.15, 'max_depth': 10, 'n_estimators': 100, 'subsample': 0.1}) This is done to get a sense of how well the model is performing.After the best hyperparameters are identified, a new LightGBM model is created with those hyperparameters and trained on the entire training set.The model is then used to make predictions on the testing set, and a confusion matrix is printed out to show the number of true positives, false positives, true negatives, and false negatives.

Finally, the importance of each feature in the dataset is calculated using the feature_importances_ attribute of the LightGBM model, and the top 10 features are printed out along with their importance scores. This information can be useful for feature selection or feature engineering in future iterations of the model.

In this case, the precision for class 0 (negative class) is 0.87, which means that 87% of the predictions for the negative class were correct. The recall for class 0 is 0.76, which means that 76% of the actual negative cases were correctly identified. For class 1 (positive class), the precision is 0.88 and recall is 0.93, indicating that 88% of the predicted positive cases are true positives, and 93% of the actual positive cases were correctly identified. The overall accuracy of the model is 0.87, which means that 87% of all predictions were correct. The most important features for predicting purchases were customer_id, day, woy, total_value, cs_128, cs_64, cs_28, and wom.

Reason for choosing LightGBM Classifier over Logistic Regression:

The accuracy score is almost the same as logistic regression but,

- LightGBM Classifier is a powerful machine learning model that is well-suited for handling time series data with unbalanced classes. The data involved in this project has unbalanced classes. It has the ability to automatically adjust its weighting scheme to give more importance to the minority class, which can help improve the model's performance.
- It is able to handle large datasets efficiently, and is designed to automatically capture complex interactions between features. The time series data involved in this project is large as the data is collected at a high frequency. Logistic regression can become computationally expensive and slow when dealing with large datasets. This is because the optimization algorithm used to fit the logistic regression model needs to process all the data at once. On the other hand, tree-based models like LightGBM can handle large datasets more efficiently because they use techniques such as histogram-based splitting and gradient-based one-side sampling to speed up training and prediction.

These factors make it a better choice than logistic regression in this case.

3. EVALUATION

CLASSIFICATION REPORT:

This code imports the classification_report function from the sklearn.metrics module and uses it to evaluate the performance of a LightGBM model on a test set. The test set features are stored in the X_test variable, and the corresponding labels are stored in the y_test variable.The predict() method of the LightGBM model is used to generate predictions for the test set, which are stored in the y_pred variable.

The classification_report() function is then called with the y_test and y_pred variables as inputs. This function computes and returns a report of various evaluation metrics for the model, including precision, recall, F1-score, and support for each label.

The model achieved an overall accuracy of 0.87, which means that 87% of the instances in the test set were correctly classified by the model. The model's precision, recall, and F1-score for label 0 were 0.87, 0.76, and 0.81, respectively, while the corresponding metrics for label 1 were 0.88, 0.93, and 0.90. This indicates that the model performed better at identifying churn cases (label 1) than non-churn cases (label 0).

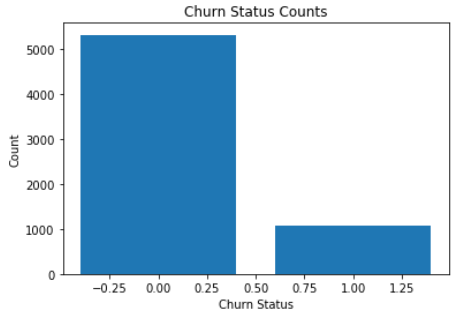
The feature importances are listed below:

	importance
customer_id	1192
day	669
woy	475
total_value	260
cs_128	149
cs_64	144
cs_28	74
wom	37

It seems like day was one of the most important features along with customer id to predict the churn status.

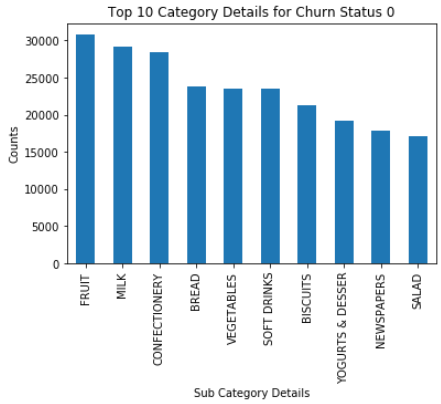
INSIGHTS

2011 customers are still considered active but 4366 customers have been classified as churned which means that 3299 customers who were active customers are now at a risk to churn and need the company's attention.



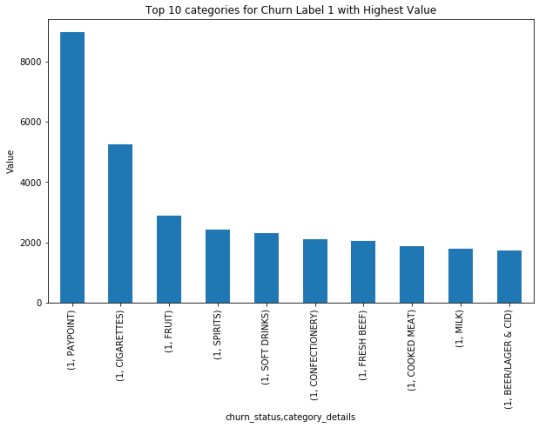
This diagram shows that previously 5310 customers were active with a median inactivity of less than 64 days. The following is the pen portraits of customers who have churned and not churned:

Active Customers:



The active customer typically makes high average spend purchases in larger quantities and the items they typically purchase, a possible category of customers is health-conscious individuals who prioritize a balanced diet and value quality and freshness. They are likely to be frequent shoppers who purchase a mix of fruits, vegetables, milk, bread, and confectionery, as well as a high purchase frequency of soft drinks, biscuits, yoghurts, newspapers, and salad.

Churned Customers:



The churned customer typically purchases alcohol, cooked meat, ready to eat food and cigarettes. They have significantly high occurrences of using PayPoint which means that they only visit the store to deposit their cash in their bank account. They are likely working bachelors who enjoy partying.

Some strategies the company can look at to address possible churners issue:

i. Provide excellent customer service: Excellent customer service is essential for retaining customers. Studies show that customers who have a positive customer service experience are more likely to remain loyal to a brand. To provide excellent customer service, businesses should address any issues or concerns promptly and professionally.
ii. Loyalty programs: Loyalty programs can also be effective in retaining customers. Studies show that customers are more likely to remain loyal to a brand when they feel appreciated and rewarded. Loyalty programs can include discounts, free items, exclusive deals, and other incentives that appeal to customers.
iii. Personalised Marketing: Since the churn profile customer purchase very specific type of products, creating a personalised marketing plan for them can prove beneficial. This includes sending personalized offers, promotions, and discounts through email or mobile notifications
iv. Act on Feedback: Finally, listening to feedback is crucial for retaining customers. By actively seeking out and listening to customer feedback, businesses can improve their products, services, and overall customer experience. This can include conducting surveys, focus groups, or other forms of market research to understand the needs and preferences of customers.