CENTRAL EUROPEAN UNIVERSITY
DEPARTMENT OF ECONOMICS AND BUSINESS
MS IN BUSINESS ANALYTICS PROGRAM

Ákos Almási

CAPSTONE PROJECT SUMMARY

Accounting Prediction

Canon Austria

Sponsor at Canon Austria:
Martin E. Brüggemann
Business Solutions Consultant

Vienna, Austria
2023

Table of Contents

1 Project introduction and overview

The stakeholder of the project was Canon Austria, who is looking to develop and implement an accounting prediction system to enhance its document classification and data extraction capabilities. Currently professionals have to manually type in the account codes during the process of filling in invoices which is prone to human error and requires substantial amount of time. The primary goal of the project was to develop a Random Forest model that can predict account codes for incoming invoices based on historical accounting data.

2 Dataset and project setup

The data was provided by the sponsor to deliver the project which included invoice details and associated account codes. The dataset contained 4064 rows (each row representing an invoice) and 22 columns. The project was carried out in a Jupyter notebook which contains the Machine learning model using the scikit-learn library, with proper documentation and visualizations. First a data quality issue was addressed, the "Buchungstext" field, contained multiple pieces of information separated by semicolons. However, the dataset itself used semicolons as column separators. This was clearly an issue, since the additional semicolons within the "Buchungstext" field, led to the creation of unnamed columns at the end of the data frame.

3 Project summary

To deliver the desired Random Forest model many steps were performed. First, the Data preprocessing and cleaning process was completed. It involved tasks such as handling missing values, removing exact duplicates, and fixing inconsistencies. Date columns were converted into datetime objects and transformed into a uniform format, outliers were handled, and categorical features were one-hot encoded.

To gain insights and identify patterns in the data an Exploratory Data Analysis was performed. Most important findings were the skewed distribution of the target variable and important

patterns were found such as seasonality of the invoices which helped the process of Feature engineering.

Feature engineering was an important step to transform raw data into meaningful and important features, it involved extracting and creating relevant features from the available dataset. The created features can be summarized into four categories (time-based features, text-based features, binary features, frequency-based features). The time-based features were created to find trends linked to timeframes. The text-based features were constructed from the textual columns available in the dataset. Binary features provided information whether an invoice meets specific conditions. Frequency-oriented features offered insights into the reoccurrence of certain attributes.

Then the predictors and the target variable were defined and created a specific seed to make sure the same results can be achieved. The dataset was then split to train and test sets for model evaluation. I built the baseline model using the Random Forest Classifier and fitted it to the training data. I evaluated the model with the use of several metrics, namely, precision, recall, F1-score, and accuracy for both training and test sets. The results clearly indicated that the model was overfitting to the training data, as there was a large difference between the training (about 97%) and test set (around 68-72%) assessment metrics.

By evaluating the feature importance plot, I could determine which features were important (approval scheme, supplier number) and which were not (sentiment, most common words in text-based columns), and I discarded the ones that are irrelevant from the next model. To further optimize the random forest model, I used a hyperparameter optimization technique. This step not only helped in optimizing the model for superior outcomes, but also helped in reducing overfitting by maximizing various model parameters, such as the maximum depth. It involved utilizing RandomizedSearchCV, which is an approach that randomly picks a fixed number of parameter settings. Based on the results the best performing parameters were chosen and

transitioned to perform a GridSearchCV. This approach generated all possible parameter combinations and selected the best one based on the results.

4 Model performance analysis and recommendations

I decided to analyze the model prediction capabilities on the account code level to gain a better understanding of the model's performance. This process involved making predictions on the account codes from the test dataset. The results demonstrated that the account codes that had at least 50 observations in the whole dataset were predicted quite successfully, with the exception of two account codes. In contrast, the account codes that had a few numbers of observations were mostly predicted incorrectly. Unfortunately, the available dataset doesn't allow accurate predictions for certain account codes, and further improvements on the database could solve the current issues. For better performance, the following features would be beneficial to gather:

- Invoice amount: The amount of each invoice.
- Currency type: The currency of each transaction (might be important if there are international suppliers).
- Department: The specific department to which the invoice belongs.
- Country/region: The country to which the supplier is linked.
- Product/service: Information about the product /service (such as the name or the number).
- Time since first transaction: The length of the business relationship with each supplier.
- Unit: Information about the item (weight, quantity, volume etc.).

It would be beneficial to gather higher volume of data points for each account codes. The current dataset has a clear issue of class imbalance which means some account codes have many observations meanwhile others have very few (some only have five observation). As observed in the analysis, class imbalances caused the model to have a bias towards the majority classes, which means that in certain situations the model tends to predict the classes with large number of observations for the classes that are underrepresented.

Besides that, I would highly recommend gathering more data to improve the model's learning potential. Since there are more than 250 classes that needs to be predicted, enhancing the dataset with more observations could result in substantially better results.