Architectural Decisions Document

USE CASE

Motivation

The use case is to perform categorization of transactions in a way that allows for automated generation of income tax declaration statement via CI/CD pipeline.

The dataset used stems from real-life export of banking data for three consecutive years of all my accounts with one banking institution. It is retrieved as a set of CSV files by manual request of the user authenticated to the bank's internet banking client application.

Justification

Though not constituting the sexiest ML problem there is, to me it provides great value and thus I opted for combining the necessary with the convenient.

ETL

Technology Choice

To work with the data, numpy arrays and pandas dataframes were used, aggregation of intermediate artefacts was done with pandasql package. The necessary postprocessing consists of labelling the CSV data which is facilitated in a combination of using casuistic SQL statements on top of the imported dataframes and manually purging them of errors.

Justification

The relational data is present in a csv file and so using a pandas dataframe to load the relational data was an obvious choice. The data volume and format is lean enough to be kept in a CSV repository (together with the CI/CD pipeline utilizing it) and so didn’t require any other object storage or data warehouse.

DATA QUALITY

Technology Choice

The data quality was - without further analytical effort - accepted as

- correct
- complete
- adequate

Justification

The data originates from me manually downloading banking data of my personal accounts through the client interface provided by my bank's internet client. I downloaded all there is since I opened the account and have tailored the use case around this data.

FEATURE ENGINEERING

Technology Choice

The specific feature engineering tasks implemented were:

- Concatenate meaningful text fragments from the transaction line (partner name, transaction purpose)
- Construct bag of words using NLTK word_tokenizer
- Encode using CountVectorizer

Justification

The above-mentioned feature engineering tasks were necessary for transforming the data into an input format suitable for the specific model and reflect the basic generative idea, that is to condense meaning of a transaction out of the only semantic hints there are in a recorded transaction.

MODEL DEFINITION

Technology Alternatives

The resulting model choice constitutes the second approach to the project problem and was chosen over the first one.

Classification with CNN (DL)

The first approach was to visualize each transaction as the PNG of a QR code and have a (ideally pre-trained) CNN perform classification by pattern recognition.

Classification with Naive Bayes (ML)

The second approach was to condense semantic meaning out of the transaction line in an NLP sense (analogous to sentiment analysis or spam filtering) and use a battle proven classifier on the extracted bag of words.

Technology Choice

Second approach was chosen using Gaussian Naive Bayes classifier.

Justification

The conservative ML approach was favored over a CNN based DL image recognition model because the latter never managed to exceed 30% accuracy whereas the former yielded 96% in the first pass.

MODEL TRAINING

Technology Choice

The project was written in python3. Scikit-learn was used as high level framework. The Colab notebook platform was used to train the model.

Justification

Sckikit-learn brings everything that was needed in the shape of elegant one-liners. The model was trained on the Colab platform in conjunction with Google Drive and GitHub because this allows for development to be done in a wherever (train, vacation,) / with whatever (notebook, iPad, iPhone) fashion.

MODEL EVALUATION

Technology Choice

The model performance indicators used are precision, recall, F1-score, accuracy and the confusion matrix. Scikit-learn was used to evaluate these metric values, yellowbrick for visualization.

Justification

The above-mentioned metrics were used to measure the performance of the model, as it was a classification problem. F1-score is the harmonic mean of precision and recall, and so is a better indicator compared to only precision or recall. Confusion matrices evaluate the number of TP, FP, TN and FN which provide a raw idea of the classification performance.

MODEL DEPLOYMENT

Technology Choice

The model as presented here is deployed as a jupyter notebook containing all the data analysis, model training, evaluation code and outputs. The productive use of the model inside a CI/CD pipeline is facilitated by establishing the necessary Python virtual environment in the executing Docker container and performing the very same steps.

Justification

As well the amount of data and the performance of the algorithm are lean enough to negate the need of a deployment as standalone product. As stated in the initial motivation, the attractiveness of the use case is the pain that it takes away by solving the urgent problem, not the complexity or dimension of it.