Project vision

This product will give our industry partner a tool at hand, that can effectively increase conversion of their leads to customers, primarily by providing the sales team with valuable information. The modular architecture makes our product future-proof, by making it easy to add further data sources, employ improved prediction models or to adjust the output format if desired.

Project mission

The mission of this project is to enrich historical data about customers and recent data about leads (with information from external sources) and to leverage the enriched data in machine learning, so that the estimated Merchant Size of leads can be predicted.

Usage

To execute the final program, ensure the environment is installed (refer to build-documents.md) and run `python .\src\main.py` either locally or via the build process. The user will be presented with the following options:

```
Choose demo:
(0) : Base Data Collector
(1) : Data preprocessing
(2) : ML model training
(3) : Merchant Size Predictor
(4) : Exit
```

(0) : Base Data Collector

This is the data enrichment pipeline, utilizing multiple data enrichment steps. Configuration options are presented:

`Do you want to list all available pipeline configs? (y/N)` If `y`:

```
Please enter the index of requested pipeline config:
(0) : config_sprint09_release.json
(1) : just_run_search_offeneregister.json
(2) : run_all_steps.json
(3) : Exit
```

- (0) Coniguration used in sprint 9.

- (1) Coniguration for OffeneRegister.
- (2) Running all the steps of the pipeline without steps selection.
- (3) Exit to the pipeline step selection.

If `n`: proceed to pipeline step selection for data enrichment. Subsequent questions arise:

```
Run Scrape Address (will take a long time)(y/N)?
Run Search OffeneRegister (will take a long time)(y/N)?
Run Phone Number Validation (y/N)?
Run Google API (will use token and generate cost!)(y/N)?
Run Google API Detailed (will use token and generate cost!)(y/N)?
Run openAI GPT Sentiment Analyzer (will use token and generate cost!)(y/N)?
Run openAI GPT Summarizer (will use token and generate cost!)(y/N)?
Run Smart Review Insights (will take looong time!)(y/N)?
Run Regionalatlas (y/N)?
```

- `Run Scrape Address (will take a long time)(y/N)?`: This enrichment step scrapes the leads website for an address using regex.
- `Run Search OffeneRegister (will take a long time)(y/N)?`: This enrichment step searches for company-related data using the OffeneRegisterAPI.
- `Run Phone Number Validation (y/N)?`: This enrichment step checks if the provided phone numbers are valid and extract geographical information using geocoder.
- `Run Google API (will use token and generate cost!)(y/N)?`: This enrichment step tries to the correct business entry in the Google Maps database. It will save basic information along with the place id, that can be used to retrieve further detailed information and a confidence score that should indicate the confidence in having found the correct result.
- `Run Google API Detailed (will use token and generate cost!)(y/N)?`: This enrichment step tries to gather detailed information for a given google business entry, identified by the place ID.
- `Run openAI GPT Sentiment Analyzer (will use token and generate cost!)(y/N)?`: This enrichment step performs sentiment analysis on reviews using GPT-4 model.
- `Run openAI GPT Summarizer (will use token and generate cost!)(y/N)?`: This enrichment step attempts to download a businesses website in raw html format and pass this information to OpenAIs GPT, which will then attempt to summarize the raw contents and extract valuable information for a salesperson.
- `Run Smart Review Insights (will take looong time!)(y/N)?`: This enrichment step enhances review insights for smart review analysis
- `Run Regionalatlas (y/N)?`: This enrichment step will query the RegionalAtlas database for location based geographic and demographic information, based on the address that was found for a business through Google API.

It is emphasized that some steps are dependent on others, and excluding one might result in dependency issues for subsequent steps.

After selecting the desired enrichtment steps, a prompt asks the user to `Set limit for data points to be processed (0=No limit)` such that the user chooses whether it apply the data enrichment steps for all the leads (no limit) or for a certain number of leads.

Note: In case `DATABASE_TYPE="S3"` in your `.env` file, the limit will be removed, in order to enrich all the data into `s3://amos--data--events` S3 bucket.

(1) : Data preprocessing

Post data enrichment, preprocessing is crucial for machine learning models, involving scaling, numerical outlier removal, and categorical one-hot encoding. The user is prompted with questions:

`Filter out the API-irrelevant data? (y/n)` : This will filter out all the leads that couldn't be enriched during the data enrichtment steps, removing them would be useful for the Machine Learning algorithms, to avoid any bias introduced, even if we pad the features with zeros. `Run on historical data ? (y/n) Note: DATABASE_TYPE should be S3!` : The user has to have `DATABASE_TYPE="S3"` in `.env` file in order to run on historical data, otherwise, it will run locally. After preprocessing, the log will show where the preprocessed_data is stored.

(2) : ML model training

Six machine learning models are available:

```
(0) : Random Forest
(1) : XGBoost
(2) : Naive Bayes
(3) : KNN Classifier
(4) : AdaBoost
(5) : LightGBM
```

After selection of the desired machine learning model, the user would be prompted with a series of questions:

- `Load model from file? (y/N)` : In case of `y`, the program will ask for a file location of a previously saved model to use for predictions and testing.
- `Use 3 classes ({XS}, {S, M, L}, {XL}) instead of 5 classes ({XS}, {S}, {M}, {L}, {XL})? (y/N)` : In case of `y`, the S, M, L labels of the data would be grouped alltogether as one class such that the training would be on 3 classes ({XS}, {S, M, L}, {XL}) instead of the 5 classes. It is worth noting that grouping the S, M and L classes alltogether as one class resulted in boosting the classification performance.

- ```
  Do you want to train on a subset of features?
  (0) : ['Include all features']
  (1) : ['google_places_rating', 'google_places_user_ratings_total',
  'google_places_confidence', 'regional_atlas_regional_score']
  ```

`0` would include all the numerical and categorical one-hot encoded features, while `1` would choose a small subset of data as features for the machine learning models

Then, the user would be given multiple options:

```
(1) Train
(2) Test
(3) Predict on single lead
(4) Save model
(5) Exit
```

- (1): Train the current model on the current trainig dataset.
- (2): Test the current model on the test dataset, displaying the mean squared error.
- (3): Choose a single lead from the test dataset and display the prediction and true label.
- (4): Save the current model to the `amos--models/models` on S3 in case of `DATABASE_TYPE=S3`, otherwise it will save it locally.
- (5): Exit the EVP submenu

(3) : Merchant Size Predictor

After training, testing, and saving the model, the true essence of models lies not just in crucial task of generating forecasted predictions for previously unseen leads.

(4) : Exit

Gracefully exit the program.
```