Task Description

Your main task is to develop a small prototype of an ETL-pipeline. The main objective is to extract, combine and transform data from different data sources, in order to be able to use these data for revenue forecasts. The result of the pipeline should be a pandas data frame, which serves as the input for the model.

Use the given repo as a template for your task.

You can find it here - https://github.com/gastromatic/data_engineering_test

As input for your ETL-pipeline, load the POS-data provided in `pos_data` and the data of the table `articles` of the `test_db`. Please load all 4 files of the POS data.

You can find the POS data here - https://www.dropbox.com/sh/6w8yu9es8r7q6ro/AACrC015jEAozdXSVbsNZ5pTa?dl=0

Task requirements

- The code should be readable and well structured.
- Code should be written in Python.
- The whole ETL-pipeline should be easily executable, e.g. in a jupyter notebook.
- Important steps in the pipeline should be explained using comments or plotting the data (or both).
- Also add a comment if you intentionally skip a step.
- Perform a prediction of the total revenue with the final results of the pipeline for the next 14 days, i. e. 2018-07-01 – 2018-07-14.
  Note: Tuning the model is not part of the main task. It‘s fine if the prediction runs through.

Task Nice-to-haves

- Adding further data to the ETL-pipeline, e.g. weather data from darksky.
- Tune the model to have good results / add an explanation why the results are bad.
- Transform the data so that it is suitable for the model of your choice, explain why.

Tech Stack

- Python
- Jupyter (Optional)
- PostgreSQL
- pipenv

Important To-Dos

- Please share the code via GitHub (other repository hosting service) or zip file.
- You can copy the given repo to use it as a template.

- The project should be easy to build and execute, e.g. by executing your jupyter notebook.
- All needed libraries should be installed over package managers (pipenv).
- Please do not make a PR with your results. Only send us the zip file with the repo.

Further Information

- The docker-image of the jupyter notebooks loads the python packages defined in the `Pipfile`. If you need to add a package, add it in the `Pipfile` and run `pipenv install –dev`. Afterwards you need to rebuild the docker-image.
- API-Key for darksky: ec406ed99bd858f99ca2a3667fb79bed (limited to 1000 calls/day)
- The .csv file is not part of the repo and will be sent separately. To have it accessible in the notebook, just upload it via the UI or load it in the notebook by editing the Dockerfile.

If you are not able to implement a feature named in the task description, please explain it in a short text so we can understand why. In case you need further information about the task, or you have any question about the technologies you should use, feel free to ask.