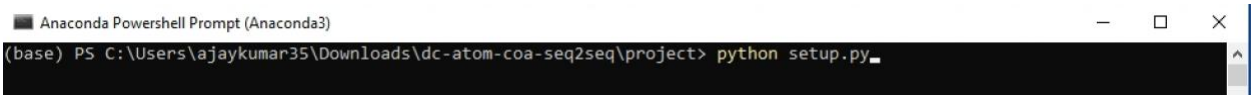
In preparation for the demo of the AI tool, we are sharing instructions for setting up and running the tool to get users acquainted. The instructions show how to run the tool using the provided sample files. However, once familiarized, we highly recommend running the tool with real chart of accounts data prior to the demo so we can review or answer any questions that come up at that time.

First time setup instructions for running ATOM.AI Python Package:

[*Requirements*: Python, PIP, Virtualenv]

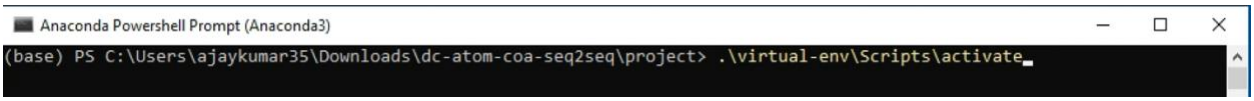
[Note: These only need to be followed the first time to get the dependencies installed]

- Download and unzip the Python package in an accessible location [For example: `C:\Users\<user-name>\Downloads\`]
- Download Anaconda for Windows from the following link: Anaconda | Individual Edition
- Install Python 3.9 from the Windows Store
- Launch `Anaconda Powershell Prompt (Anaconda3)` from the Start Menu
- Navigate to the project folder of the ATOM.AI python package
- Run the included setup script as follows



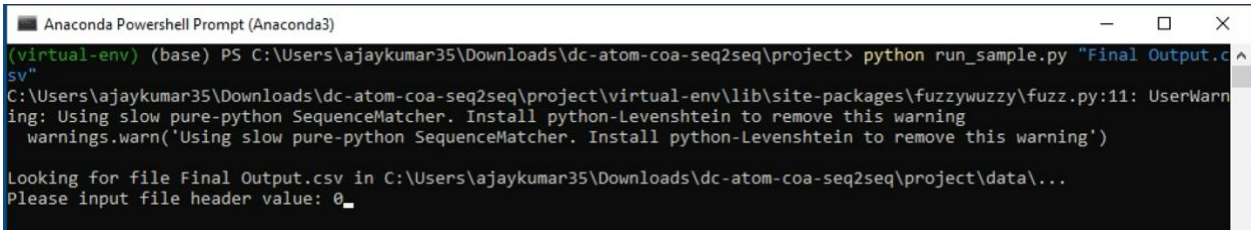
Instructions for getting predictions from ATOM.AI Python Package:

- Activate the virtual environment [that was setup in above] to load required libraries and packages as follows

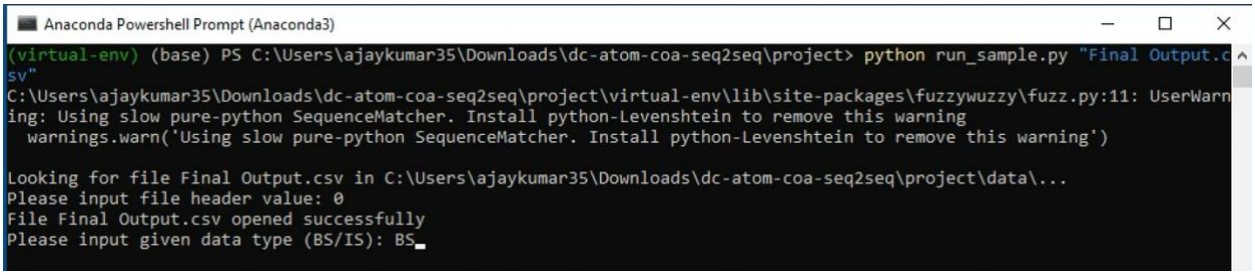


- Run the included python script `[> python run_sample.py]` that uses the trained models to predict on a given file* containing source account descriptions

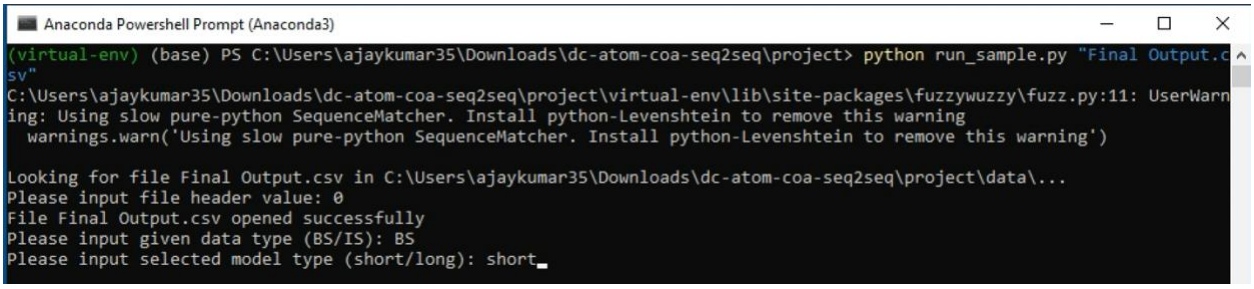
- Input the index number for the headers in the input file [default is 0] at the prompt



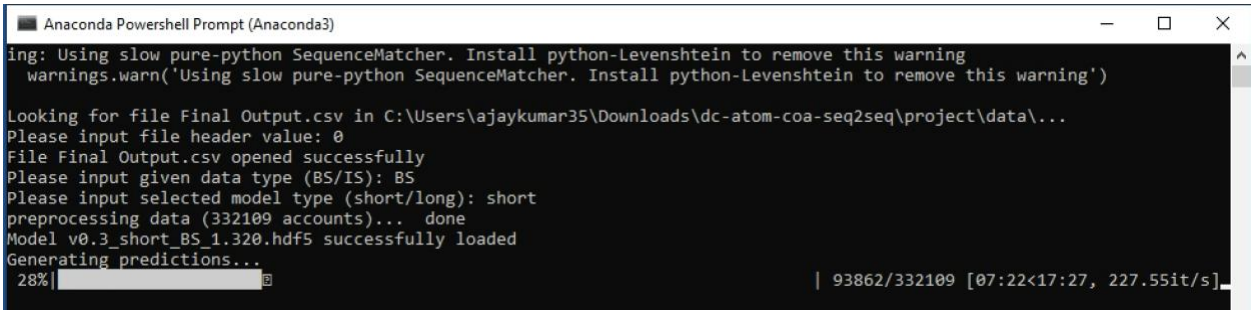
- Input the data type of the input files, Balance Sheet (BS) or Income Statement (IS) [default is BS] at the prompt



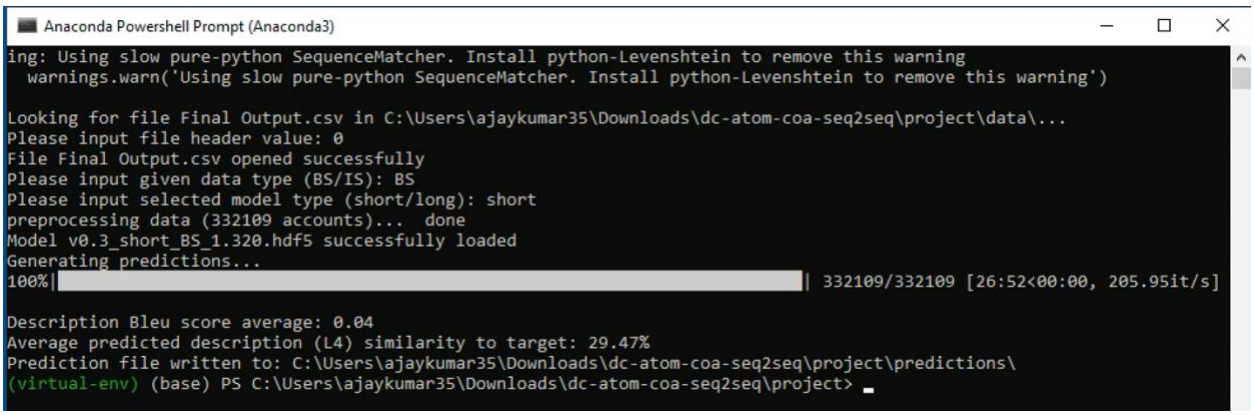
- Input the model type to use, short (only account descriptions given) or long (account hierarchy along with account descriptions given) [default is short] at the prompt



- Processing...



- One run is complete, the results are saved in the `predictions` folder with `'_predictions'` appended to input filename



- * Input file must be in the `data` folder within the package directory
- * Expected columns in input file:

    - **ERP**, *str; Required* – ERP name, ex: SAP, Oracle, MS Dynamics, etc.
    - **System**, *str; Required* – System name, ex: Atlas, SAP, etc.
    - **Source GL Account**, *int; Required* – Source account number
    - **Source GL Account Description**, *str; Required*
    - **Source L1**, *str; Optional*
    - **Source L2**, *str; Optional*
    - **Source L3**, *str; Optional*
    - **Target GL Account**, *int; Optional* – Target account number
    - **Target GL Account Description**, *str; Optional*
    - **Target L1**, *str; Optional*
    - **Target L2**, *str; Optional*
    - **Target L3**, *str; Optional*
    - **Industry**, *str; Required* – Company Industry, ex: Retail, Life Sciences, etc.
    - **FileName**, *str*; *Required*– Name of Company, ex: Fanatics, etc.

Instructions for getting predictions from ATOM.AI Alteryx Workflow:

- Select Alteryx workflow file for type of data [Balance Sheet (BS) or Income Statement (IS)] to generate predictions on. Options include:

    - `Predict New BS AcctDescOnly Workflow.yxmd`
    - `Predict New IS AcctDescOnly Workflow.yxmd`

- The data needs to be located in the `data` folder. See Example files located in `data` folder for file format.

FAQ:

- Q: What does the tool do?
- A: The tool takes a given account descriptions for a source system and predicts account descriptions for a target account system. The model is trained on historical data across companies and systems and is based on an encoder-decoder architecture. The models are separately trained on Balance Sheet and Income Statement data and is specified when invoking the script. Additionally, we support input that only includes account descriptions and account descriptions along with hierarchy. Note, to handle variability, we take the top three levels of hierarchy provided. That is, If < 2 hierarchy levels are provided, we treat it as only account descriptions are provided. If >3 hierarchy levels are provided, we trim to only include the top three levels.
- Q: Why should I use python package over the alteryx workflow?
- A: The Python package while a little bit more involved is a bit more mature in the development cycle. The alteryx workflows are under active development and will catch up soon. The rationale for releasing both is to solicit users feedback to inform development.