Milestone 3 Description

> *Deadline Saturday 30/11 @ 11:59PM*

Make sure to read all the description before starting the milestone

Objectives:

1. Loading the dataset (5%)
2. Perform some simple cleaning (30%)
    - Column renaming: 10%
    - Detect missing: 35%
    - Handle missing: 35%
    - Check missing : 20%
3. Perform some analysis on the dataset (30%)
4. Add new columns with feature engineering (15%)
5. Encode categorical columns (10%)
6. Create a lookup table for encoding only (5%)
7. Saving Cleaned dataseta and lookup table (5%)
8. ***BONUS**: Saving the output into a postgres database (5%)

Datasets

This milestone you will be working with parquet filea, so in the following Dataset Link, you can find your datasets inside the parquet directory

Deliverables

1. Python Notebook with the following naming `m3_spark_<id>.ipynb` eg. `m3_spark_52_XXXX.ipynb`
2. Cleaned Parquet file named: `fintech_spark_52_XXXX_clean.parquet`
3. Lookup table named: `lookup_spark_52_XXXX.parquet`
4. **Incase of doing the bonus**: Screenshots from PGAdmin showing the cleaned table (some of the rows) and another one showing the lookup table.

Note: All these files should reside in a folder for milestone 3, inside the root drive folder created previously in milestone 1.

Also note that: You may not need to run the spark containers since pyspark aleady creates a mini server by default.

Requirements

Part 1: Loading the dataset

Simply load the dataset from the parquet format given in the google drive above

- Load the dataset.
- Preview first 20 rows.
- How many partitions is this dataframe split into?
- Change partitions to be equal to the number of your logical cores

Part 2: Cleaning

- Rename all columns (replacing a space with an underscore, and making it lowercase)
- Detect missing
  - Create a function that takes in the df and returns any data structrue of your choice(df/dict,list,tuple,etc) which has the name of the column and percentage of missing entries from the whole dataset.
  - Tip : storing the missing info as dict where the key is the column name and value is the percentage would be the easiest.
  - Prinout the missing info
- Handle missing
  - For numerical features replace with 0.
  - For categorical/strings replace with mode
- Check missing
  - Afterwards, check that there are no missing values

Part 3: Encoding

Encode only the following categorical values

- Emp Length: Change to numerical
- Home Ownership: One Hot Encoding
- Verification Status: One Hot Encoding
- State: Label Encoding
- Type: One Hot Encoding
- Purpose: Label Encoding
- For the **grade**, only descretize it to be letter grade, not need to label encode it further

DO NOT Encode the employment title of description or any other column that is not mentioned above

Part 4: Feature Engineering

Write a function that adds the 3 following features. Try as much as you can to use built in fucntions in PySpark (from the functions library) check lab 8, Avoid writing UDFs from scratch.

- Previous loan issue date from the same grade
- Previoius Loan amount from the same grade
- Previous loan date from the same state and grade combined
- Previous loan amount from the same state and grade combined

Part 5: Analysis SQL vs Spark

Answer each of the following questions using both SQL and Spark:

1. Identify the average loan amount and interest rate for loans marked as "Default" in the Loan Status, grouped by Emp Length and annual income ranges. Hint: Use SQL Cases to bin Annual Income into Income Ranges
2. Calculate the average difference between Loan Amount and Funded Amount for each loan Grade and sort by the grades with the largest differences.
3. Compare the total Loan Amount for loans with "Verified" and "Not Verified" Verification Status across each state (Addr State).
4. Calculate the average time gap (in days) between consecutive loans for each grade using the new features you added in the feature engineering phase

5. Identify the average difference in loan amounts between consecutive loans within the same state and grade combination.

IMPORTANT: I advice you to not use ChatGPT or Copilot to do the analysis task as the whole idea is you understanding how to think about queries and design them. Completely using any of the aformentioned tools will result in some deductions in the grade.

Part 6: Lookup Table & Saving the dataset

- Create a lookup table for the encodings only
- Finally load (save) the cleaned PySpark df and the lookup table to parquet files

BONUS: Loading to Postgres

- Load the cleaned parquet file and lookup table into a Postgres database.
- Take Screenshots showing the newly added features in the feature engineering section
- Take a screenshot from the lookup table

Submission guidelines

Upload all the deliverables in your google drive milestone folder.

Best of luck.

.