

## GROUP 2: EMBEDDING EAGLES

### PROJECT MILESTONE 3 WRITE UP DOCUMENT

**Introduction:** We were asked to take a dataset out of five datasets and then select any one use case out of the lists for that dataset and implement a use case in Jupyter Notebook including a writeup following certain rubric and criteria.

**Input:** A dataset and a use case are provided to us as inputs. We are free to use any method out of the all paths and alternatives that we are comfortable with.

**Output:** A Jupyter notebook with implementation of the use case must be done and code used for implementation must be provided in a clear way. A writeup document must be submitted which contains an executive summary of the project.

#### Methodology:

- **Status since the last milestones:** Compared to last time, we have made considerable progress. Last time we chose the dataset and we just explored it. We checked for various elements like finding metadata, checking consistency of the dataset. The main aim of the last milestone was exploration but now it is implementation of an application. We did not focus much on data pre-processing and data cleaning but this time we did clean the data for the sake of model. We used the dataset and built a Text Classification model as an application for categorizing the regulations in the dataset to different types of legal domains: **Health, Privacy, Corporate, Criminal, Finance**. Last time we tried applying advanced topics like Data Visualization and made considerable progress but later faced some issues.

- **Failed Ideas/ experiments:**  
While doing these milestones, we have faced two failed ideas for this project milestone.

The first failed idea was using the Zeroshot approach for getting annotations done using AI LLM via Prodigy to train the model. First we thought about annotating manually as we were doing the same for last projects. But this time, time was very limited and hence we wanted to try something different. Then we came across the professor's video on Slack on how to use AI LLM to label annotations. We stuck to the Prodigy documentation and followed the same steps mentioned in the video by taking the Slack video as the reference.

We did face a lot of issues on the open AI keys topic during executing the final command. We first typed the command with "dotnev" and later realized that was unnecessary as we are not having any sensitive information to differentiate it with the code in the project. Then we typed the command without any "dotenv" but then also we got an error that we require Open API keys. Then we used our Open AI account, got the keys and used them but we were getting another error with too many requests. Even after trying again and again after setting environment variables also, we got the same error again and again. So we switched to Excel, then added text and labels manually

The second idea which failed is after getting the annotations, we planned to build the model with Spacy approach and we tried but due to some error, we were unable to do so. I think there was a problem with the “config.cfg” file. We followed Prodigy documentation and are confident that we have used appropriate instructions but somewhere something went wrong and then we switched our approach to Prodigy train as it is very fast and simple. We thought of trying Hugging Face but lack of time really drove us to pick the Prodigy approach. It was simple and efficient and we got the model without any further issues.

The screenshot shows a VS Code editor with a file named 'ecfr.json' containing a large block of legal text, likely from the Federal Reserve's ECFR. The text is partially obscured by line numbers 259 through 281. Below the editor, the 'TERMINAL' tab is active, displaying a Python error message: 'ConnectionError: API could not be reached after 31.969 seconds in total and attempting to connect 5 times. Check your network connection and the API's availability.' The error occurs in a script located at 'C:\Users\joyvi\OneDrive\Documents\PM3\data\venv\lib\site-packages\spacy\_llm\models\rest\openai\model.py'.

## ● Blockers (Challenges):

There were a couple of challenges we faced while doing this project. They are

- ❖ First one was starting with Google Colab. We started this project with the Google Colab but we continuously faced runtime issues as for the basic plan, there was only limited run time allocated. After every runtime, the files uploaded and generated were deleted. So we did upload files and executed codes again and again.
- ❖ Second challenge was the same with Google Colab. We did not plan on using Zeroshot for labeling annotations using AI first. But when we decided to go with automating the annotation process, we needed Prodigy to install on the system but we did not know that Google Colab has a Prodigy extension. So we typed code in VS Code again in the system where we were having Prodigy installed already. This was like double work to do.
- ❖ We took no exclusive labels for the annotating process but sometimes there may be a combination of labels to go with. And the text for labeling was also very lengthy, and assigning labels manually would be very challenging.

- ❖ When we thought of cleaning the data, it was very challenging to preprocess and clean the data. Hence, we did not touch that deeply and we did clean but to a very lesser extent and we did not focus that much on the cleaning and preprocessing.
- ❖ Dataset was very complex to understand. It is ECFR dataset meaning “European Code of Federal Regulations” as we are international students we are unable to understand the data. It is very law related and requires basic knowledge about how things are done federally and all the regulations related stuff.

- **Preliminary results:**

1. Introduction to the Project

Objective: The main objective of the project is to read texts to categorize the regulations into various legal domains like labels: CORPORATE, PRIVACY, CRIMINAL, FINANCE, HEALTH.

Importance: This classification is a very important aspect as in the public sector, to divide all the regulations into their domains so that respective staff and authorities can carry out their respective duties.

2. Overview of the Dataset

Data Source: ECFR (Electronic Code of Federal Regulations) dataset contains a comprehensive collection of the regulations issued by federal agencies of the United States government. It provides a structured compilation and codification of the rules and regulations that govern various aspects of federal law. Data Preparation: Dataset needed cleaning and preprocessing like removing unwanted symbols and removing missing and null values.

3. Methodology

Tool Selection: Visual Studio Code and Prodigy. We used Visual Studio Code for writing the code in Jupyter notebooks and Prodigy for trying the annotations using AI but it failed. Then we used this tool again to generate the model.

Model Training: As we got annotations then we used the Prodigy train to train the model. 70% of the data is training models and 30% of the data is testing data.

Model Architecture: For text classification models, we simply build the model from scratch. So we did not use any pretrained models.

4. Preliminary Results

Evaluation Metrics: We used the attributes generated from prodigy trains (accuracy, precision, recall, F1 score, etc.)

All the model data is available on the Jupyter notebook.

Initial Findings and Background Details:

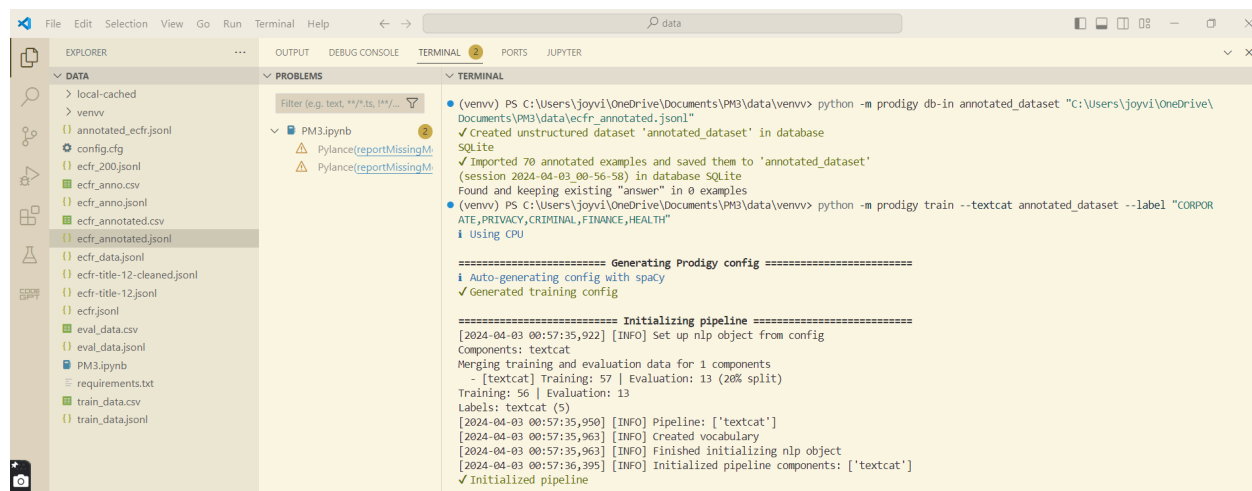
- Data was a bit messy so we cleaned the data. Initial findings was the data link was having all stuff related to the content and social stuff.
- For labels, we googled the definitions and gained knowledge about the labels to decide which label must be given to the text.
- The data was very lengthy and it exceeded the tokens, so we shifted the annotation approach to do it manually without using AI.

- Next Steps for the semester:** For next time, we would like to improve accuracy of the model and improve any weak points in the model. Every model requires improvement and every time there is a chance to improve. Conducting extensive testing with a broader set of regulations will validate the model's effectiveness across different domains and implement feedback loops for continuous improvement. We would like to try new approaches like using pretrained models like BERT and Hugging Face etc. We did not use that concept till now. We can get feedback from our friends and professors to improve extensive documentation write-up too.

## Conclusion:

Despite the challenges encountered, the progress made from Milestone 2 to Milestone 3 was considerable. It indicates a positive trajectory towards achieving our objective of developing a text classifier model for eCFR regulations. We made quite progress with this milestone as we tried implementing the Zeroshot concept which is very unique and new to us. Having annotations done by AI is very useful especially when dealing with large datasets as doing all the annotations manually will be very time consuming. By focusing on the outlined next steps, we are confident that our model will be useful and has the ability to give accurate results and we are looking forward to modifying it, retraining it to increase its performance and further contribute a valuable tool for navigating legal regulations across various domains. We are ready to learn new things and implement them to develop advanced and useful projects in future.

## Model:



```

File Edit Selection View Go Run Terminal Help
data

EXPLORER
DATA
  local-cached
  venvv
  annotated_ecfr.jsonl
  config.cfg
  ecfr_200.jsonl
  ecfr_anno.csv
  ecfr_anno.jsonl
  ecfr_annotated.csv
  ecfr_annotated.jsonl
  ecfr_data.jsonl
  ecfr-title-12-cleaned.jsonl
  ecfr-title-12.jsonl
  ecfr.jsonl
  eval_data.csv
  eval_data.jsonl
  PM3.ipynb
  requirements.txt
  train_data.csv
  train_data.jsonl

PROBLEMS
Filter (e.g. text, **/*.ts, !**/...)
PM3.ipynb
  Pylance[reportMissingM...
  Pylance[reportMissingM...

TERMINAL
(venvv) PS C:\Users\joyvi\OneDrive\Documents\PM3\data\venvv> python -m prodigy db-in annotated_dataset "C:\Users\joyvi\OneDrive\Documents\PM3\data\ecfr_annotated.jsonl"
✓ Created unstructured dataset 'annotated_dataset' in database SQLite
✓ Imported 70 annotated examples and saved them to 'annotated_dataset' (session 2024-04-03 00:56:58) in database SQLite
Found and keeping existing "answer" in 0 examples
(venvv) PS C:\Users\joyvi\OneDrive\Documents\PM3\data\venvv> python -m prodigy train --textcat annotated_dataset --label "CORPORATE, PRIVACY, CRIMINAL, FINANCE, HEALTH"
i Using CPU

===== Generating Prodigy config =====
i Auto-generating config with spaCy
✓ Generated training config

===== Initializing pipeline =====
[2024-04-03 00:57:35,922] [INFO] Set up nlp object from config
Components: textcat
Merging training and evaluation data for 1 components
- [textcat] Training: 57 | Evaluation: 13 (20% split)
Training: 56 | Evaluation: 13
Labels: textcat (5)
[2024-04-03 00:57:35,950] [INFO] Pipeline: ['textcat']
[2024-04-03 00:57:35,963] [INFO] created vocabulary
[2024-04-03 00:57:35,963] [INFO] Finished initializing nlp object
[2024-04-03 00:57:36,395] [INFO] Initialized pipeline components: ['textcat']
✓ Initialized pipeline
  
```

File Edit Selection View Go Run Terminal Help

data

EXPLORER

- DATA
  - local-cached
  - venv
  - annotated\_ecfr.jsonl
  - config.cfg
  - ecfr\_200.jsonl
  - ecfr\_anno.csv
  - ecfr\_anno.jsonl
  - ecfr\_annotated.csv
  - ecfr\_annotated.jsonl
  - ecfr\_data.jsonl
  - ecfr-title-12-cleaned.jsonl
  - ecfr-title-12.jsonl
  - ecfr.jsonl
  - eval\_data.csv
  - eval\_data.jsonl
  - PM3.ipynb
  - requirements.txt
  - train\_data.csv
  - train\_data.jsonl

PROBLEMS

- PM3.ipynb
  - Pylance(reportMissingM)
  - Pylance(reportMissingM)

TERMINAL

```
===== Training pipeline =====
Components: textcat
Merging training and evaluation data for 1 components
- [textcat] Training: 57 | Evaluation: 13 (20% split)
Training: 56 | Evaluation: 13
Labels: textcat (5)
Pipeline: ['textcat']
Initial learn rate: 0.001
E # LOSS TEXTCAT CATS_SCORE SCORE
---
0 0 0.16 15.24 0.15
3 200 11.96 26.84 0.27
7 400 2.49 29.33 0.29
10 600 2.12 29.33 0.29
14 800 0.21 29.33 0.29
17 1000 0.00 29.33 0.29
21 1200 0.00 29.33 0.29
25 1400 0.00 29.33 0.29
28 1600 0.00 29.33 0.29
32 1800 0.00 29.33 0.29
35 2000 0.00 29.33 0.29
✓ saved pipeline to output directory
CORPORATE,PRIVACY,CRIMINAL,FINANCE,HEALTH\model-last

===== Textcat F (per label) =====

P R F
CORPORATE 0.00 0.00 0.00
CRIMINAL 0.00 0.00 0.00
FINANCE 66.67 100.00 80.00
CORPORATE 100.00 50.00 66.67
PRIVACY 0.00 0.00 0.00

===== Textcat ROC AUC (per label) =====

ROC AUC
```

Ln 1, Col 1 Spaces: 4 UTF-8 CRLF JSON Lines Prettier

File Edit Selection View Go Run Terminal Help

data

EXPLORER

- DATA
  - local-cached
  - venv
  - annotated\_ecfr.jsonl
  - config.cfg
  - ecfr\_200.jsonl
  - ecfr\_anno.csv
  - ecfr\_anno.jsonl
  - ecfr\_annotated.csv
  - ecfr\_annotated.jsonl
  - ecfr\_data.jsonl
  - ecfr-title-12-cleaned.jsonl
  - ecfr-title-12.jsonl
  - ecfr.jsonl
  - eval\_data.csv
  - eval\_data.jsonl
  - PM3.ipynb
  - requirements.txt
  - train\_data.csv
  - train\_data.jsonl

PROBLEMS

- PM3.ipynb
  - Pylance(reportMissingM)
  - Pylance(reportMissingM)

TERMINAL

```
14 800 0.21 29.33 0.29
17 1000 0.00 29.33 0.29
21 1200 0.00 29.33 0.29
25 1400 0.00 29.33 0.29
28 1600 0.00 29.33 0.29
32 1800 0.00 29.33 0.29
35 2000 0.00 29.33 0.29
✓ saved pipeline to output directory
CORPORATE,PRIVACY,CRIMINAL,FINANCE,HEALTH\model-last

===== Textcat F (per label) =====

P R F
CORPORATE 0.00 0.00 0.00
CRIMINAL 0.00 0.00 0.00
FINANCE 66.67 100.00 80.00
CORPORATE 100.00 50.00 66.67
PRIVACY 0.00 0.00 0.00

===== Textcat ROC AUC (per label) =====

ROC AUC
CORPORATE None
CRIMINAL 0.86
FINANCE 0.85
CORPORATE 0.82
PRIVACY 0.92

C:\Users\joyvi\OneDrive\Documents\PM3\data\venv>
```