**Documentation**

Note: payever\_task.py (python script) contains the code which follows flake8/pep8 linting

**1. Introduction**

This documentation presents an overview of an ML project focusing on text data preprocessing, feature engineering, model building, and evaluation. The project utilizes synthetic data generated by Faker, a Python library for creating fake data, to simulate a dataset for classification tasks.

**2. Dataset Generation**

Synthetic data is generated using the Faker library to mimic realworld data.

The dataset consists of textual data fields such as name, description, and a numerical feature (price).

Categorical target labels are also included to simulate a classification problem.

**3. Data Preprocessing**

**Textual Data:**

* Conversion to lowercase to ensure uniformity.
* Removal of punctuation and special characters to clean the text.
* Tokenization to split text into individual words.
* Removal of stopwords to eliminate common and less meaningful words.
* Lemmatization to reduce words to their base or root form.

**Numerical Data:**

* The price column is scaled using `MinMaxScaler` to normalize its range.

**4. Feature Engineering**

* Text sequences are generated using `Tokenizer` and padded to ensure uniform length.
* All feature columns (text sequences and price) are combined into a single feature matrix.

**5. Model Building**

A neural network model is constructed using Keras with the following architecture:

* **Embedding Layer:** For text input.
* **Flatten Layer:** To convert the text sequences into a 1D array.
* **Dense Layers:** With activation functions, batch normalization, dropout, and L2 regularization to prevent overfitting.

The model is compiled sparse categorical cross entropy loss function and adam optimizer for the product classification task.

**6. Model Training and Evaluation**

* The model is trained on the training dataset and evaluated on the test dataset.
* Evaluation metrics include precision, recall, F1score, support, and accuracy, calculated using `classification\_report`.
* The performance of the model is assessed based on these metrics to determine its effectiveness.

**7. Assumptions**

* The project assumes that the synthetic dataset generated by Faker adequately represents realworld data for classification tasks.
* It assumes that the preprocessing steps applied to textual data sufficiently clean and prepare the text for modeling.

**8. Limitations of the Prototype**

* **Limited Dataset Realism:** The synthetic dataset may not capture the full complexity and diversity of realworld data, potentially affecting the generalization of the model.
* **Simplified Preprocessing:** The preprocessing pipeline may not account for all possible variations and nuances in textual data, leading to potential information loss or noise. For example, code for removing URL links is not present in the file
* **Model Complexity:** The neural network model architecture and hyperparameters are chosen based on intuition and experimentation, without exhaustive optimization.

**Instructions to run the code**:

**1. Git Clone:**

git clone <repository\_url>

Replace `<repository\_url>` with the URL of your Git repository.

**2. Change Directory:**

cd <repository\_name>

Replace `<repository\_name>` with the name of the cloned repository directory.

**3. Create Conda Virtual Environment:**

conda create --name myenv python=3.12

Replace `myenv` with the desired name for your virtual environment.

**4. Activate Virtual Environment:**

**- For Windows:** conda activate myenv

**- For MacOS/Linux:** source activate myenv

**5. Install Packages from `requirements.txt`:**

pip install -r requirements.txt

This command will install all the required packages into the virtual environment.

**6. Run Jupyter Notebook:**

jupyter notebook

This command will start the Jupyter Notebook server.

**7. Navigate to Notebook:**

Open your browser and go to the URL provided by the Jupyter Notebook server (usually `http://localhost:8888`). Navigate to the directory containing your Jupyter Notebook file (`\*.ipynb`).

**8. Open and Run Code:**

- Open your Jupyter Notebook file.

- Run each cell containing the code by selecting the cell and either clicking the "Run" button or pressing `Shift + Enter` on your keyboard.