

**The Superior University Lahore**

**Faculty of Computer Science & Information**

**Technology**

**Name: Alishba Haroon**

**Roll No: BSAI-116-4C**

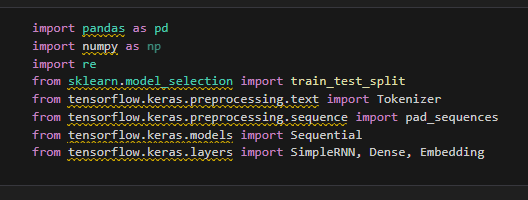
**Date: 28 March 2025**

**Subject: PAI LAB**

**Lab - Task 9**

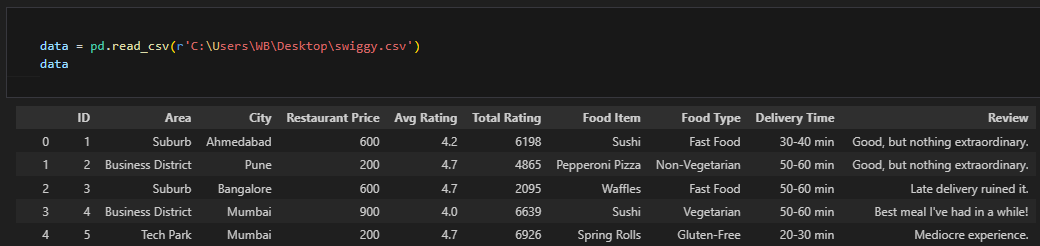
**QUESTION: 1**

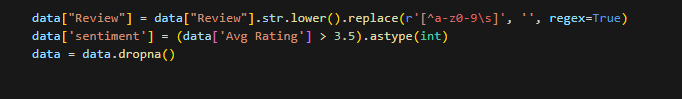
**1. Import Necessary Libraries**

****

* **pandas:** Handles data loading and preprocessing.
* **numpy:** Provides numerical operations.
* **re:** Used for text cleaning, removing special characters.
* **scikit-learn (train\_test\_split):** Splits data into training and testing sets.
* **TensorFlow & Keras:**
* **Tokenizer:** Converts text into numerical sequences.
* **pad\_sequences:** Ensures uniform input size for the model.
* **Sequential:** Defines a neural network model.
* **SimpleRNN:** A Recurrent Neural Network (RNN) layer.
* **Dense:** Fully connected layer for classification.
* **Embedding:** Converts words into vector representations.

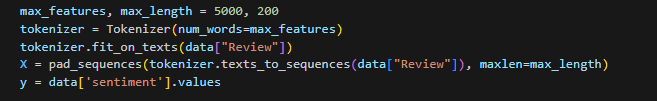
**2. Load and Preprocess Data**

****

****

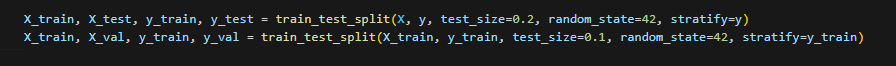
* Loads the Swiggy restaurant reviews dataset from a CSV file.
* Converts review text to lowercase and removes special characters.
* Converts ratings into sentiment labels:
* If Avg Rating > 3.5: Positive (1).
* Else: Negative (0).
* Removes missing values to prevent errors.

**3. Tokenization & Text Processing**

****

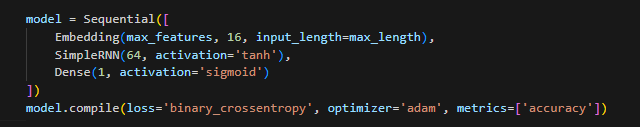
* max\_features = 5000: Limits vocabulary size to the 5000 most common words.
* Tokenization: Converts text reviews into numerical sequences.
* Padding: Ensures all sequences are of equal length (max\_length = 200).
* Stores processed text as X and sentiment labels as y.

**4. Splitting Data into Train, Validation & Test Sets**

****

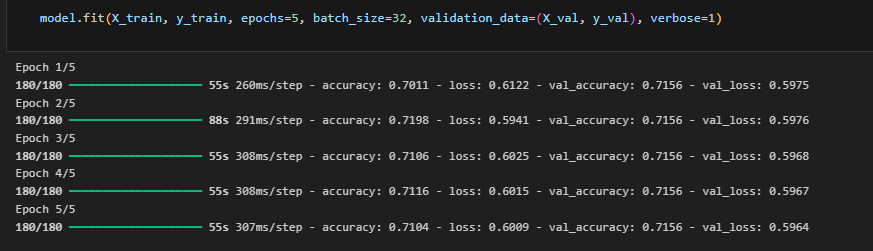
* Splits the dataset into:
  + 80% training, 20% testing (train\_test\_split).
  + Further splits training data into 90% training, 10% validation.
* Stratify ensures balanced distribution of sentiment labels.

**5. Building the Neural Network Model**

****

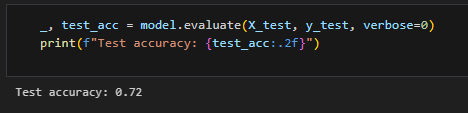
* Embedding Layer: Converts words into dense vector representations (word embeddings).
* SimpleRNN Layer:
* 64 units: Memory cells that capture word relationships.
* tanh activation: Helps in handling sequence-based data.
* Dense Layer:
* Single neuron with sigmoid activation outputs probability between 0 and 1.
* If probability ≥ 0.5: Positive sentiment, else Negative.

**6. Compile and Train the Model**

****

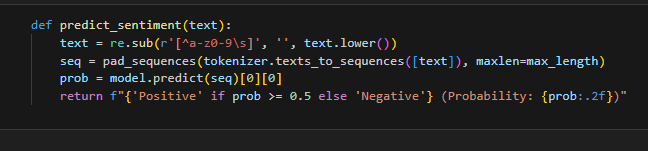
* Binary Crossentropy: Loss function for binary classification.
* Adam Optimizer: Adaptive learning rate for better convergence.
* Accuracy Metric: Used to track model performance.
* Training the model for 5 epochs with a batch size of 32.

**7. Model Evaluation**

****

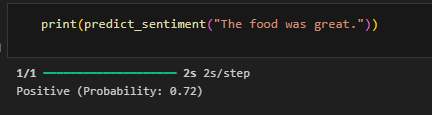
* Evaluates the model on test data.
* Prints final accuracy score.

**8. Function to Predict Sentiment for New Reviews**

****

* Cleans and tokenizes new review text.
* Predicts sentiment (positive/negative) with probability score.

**9. Example Prediction**

****

* Passes "The food was great." into the model.
* Outputs sentiment with probability.