MACHINE LEARNING FINAL PROJECT INSTUCTIONS

General Requirements:

1. Each student is required to complete the assignment using different datasets individually. Each dataset represents a distinct problem type* (e.g., classification, regression, or clustering) and must be addressed separately.
2. Students must prepare one Jupyter Notebook (`.ipynb`) file per dataset, following the provided template structure.
   a. Notebooks must be self-contained, clearly organized, and follow machine learning best practices.
   b. Use Markdown cells to explain your steps and organize your code into well-labeled sections.
3. Each `.ipynb` file should be named in the following format: DS_ID_StudentID.ipynb DS_ID: dataset id StudentID: 2118*****. These files must run without errors on the instructor's computer. File paths must be written assuming datasets are in the same directory as the code. Submissions with broken code or invalid paths will be graded as 0.

Complete the following for each dataset:

- Use a fixed random seed of 13 during data split.
- Use 80% of the data for training and 20% for testing.
- Apply stratified sampling if applicable to maintain class distributions.
- Perform appropriate data preprocessing and feature extraction.
- Implement a feature extraction function and save the transformed dataset.
- Train multiple machine learning models and compare their performance and response time.
- Apply feature extraction only to training data, then save and reuse it for inference on test data.
- Train the best-performing model and save it.
- Include relevant visualizations such as:
  - Cross-correlation map
  - Heatmaps
  - Feature importance plots
  - Confusion matrix
- Clearly report model performance using appropriate metrics

Important Notes: Assignments that do not meet the above requirements will receive a grade of zero (0). For any clarifications, students are encouraged to contact the instructor at acseckin@adu.edu.tr

1. **(30p) PM 980 DATASET FOR SIGNAL CLASSIFICATION:**
   - **Inputs:**
     - 8 sensor data (1 sound, 3 accelerometer, 3 gyroscope, temperature)
     - Speed interval
     - Sampling rate: 90Hz. Use only time and frequency features (do not use time-frequency analysis like STFT, wavelet transform or MFCC).
     - Show inputs, output cross-correlation map
   - **Output:**
     - **Classification:** 9 classes (healthy, scratch, notchshort, notchlong, singlecutlong, singlecutshort, twocutlong, twocutshort, warped)
     - Use stratified 10-fold cross validation to determine best
     - **Show performance comparison including:** accuracy, f1, precision, recall, training time, test time for each class.
     - **Outputs are balanced for each class.** You must use class based classification for each class.
     - **Create a confusion matrix figure for best model.**
   - **Grading:**
     - Upload only jupyter notebook and html of the work
     - Class based accuracy ranking %50
     - code explanations, visuals and code logical pipeline layout %50

2. **(30p) FORECASTING EXCHANGE RATE BY COMBINING NEWS AND EXCHANGE RATE DATA** Download Turkish News Article and combine with CBTR data to predict exchange rate.
   - **Inputs:**
     - **Author, title, text**
     - **Combine with Central Bank of Republic of Turkiye data**
     - **Create feature extraction from texts (title, text)**
     - **Use feature engineering methods**

    - Use sliding window to forecast the usd/try see figure 1. You can choose any window width; but stride value must be 1. In this way, predictions can be made for each day. You can choose different window widths and optimize.
- Outpur
    - usd/try value
    - **Show performance comparison including:** MAE, R2 and MAPE, training time, test time
- **Grading:**
    - MAPE ranking %50
    - code explanations, visuals and code logical pipeline layout %50

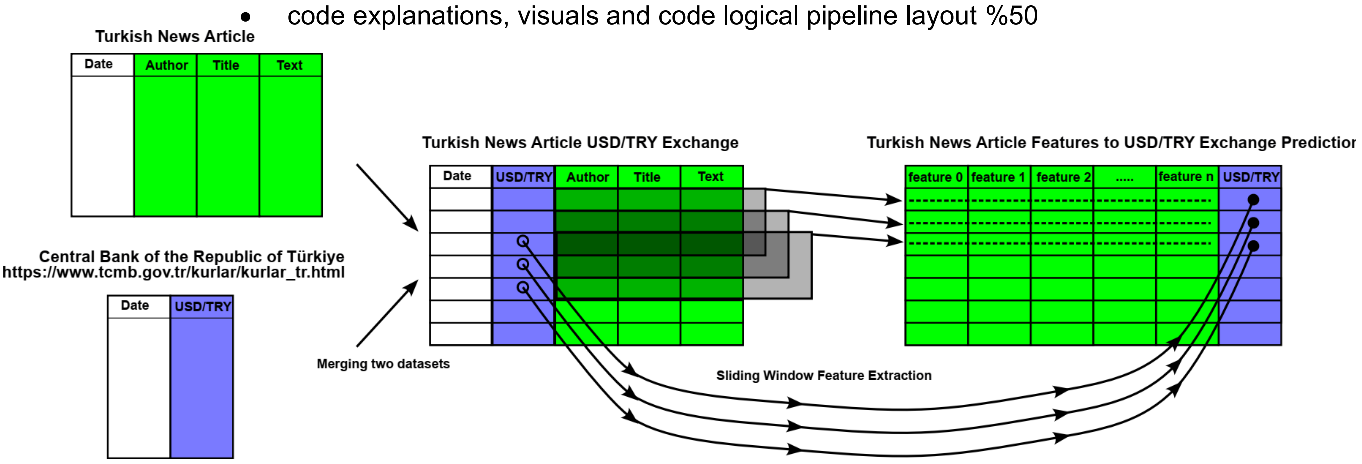
Turkish News Article
Date
Author
Title
Text
Central Bank of the Republic of Türkiye
https://www.tcmb.gov.tr/kurlar/kurlar_tr.html
Date
USD/TRY
Merging two datasets
Turkish News Article USD/TRY Exchange
Date
USD/TRY
Author
Title
Text
Sliding Window Feature Extraction
Turkish News Article Features to USD/TRY Exchange Predictior
feature 0
feature 1
feature 2
.....
feature n
USD/TRY

Figure 1: Text based feature extraction with sliding window for regression

3. **(40P) TEXTURE IMAGE SEGMENTATION**
    - Input:
        - a raw image and mask. See figure 2
        - Create tabular data by performing feature extraction with 2D sliding window from input image and mask image. You can assign any value between [1, 20] to the window size, but the stride value must be 1. Each color in the mask image corresponds to a different pattern.
        - **Use texture feature engineering methods**
    - Output:
        - Create a model that predicts the output similar to the mask image by making predictions for the entire image.
        - **Show performance comparison including:** IoU, training time, test time
    - **Grading**
        - IoU value will be used for ranking purposes.%50
        - code explanations, visuals and code logical pipeline layout %50

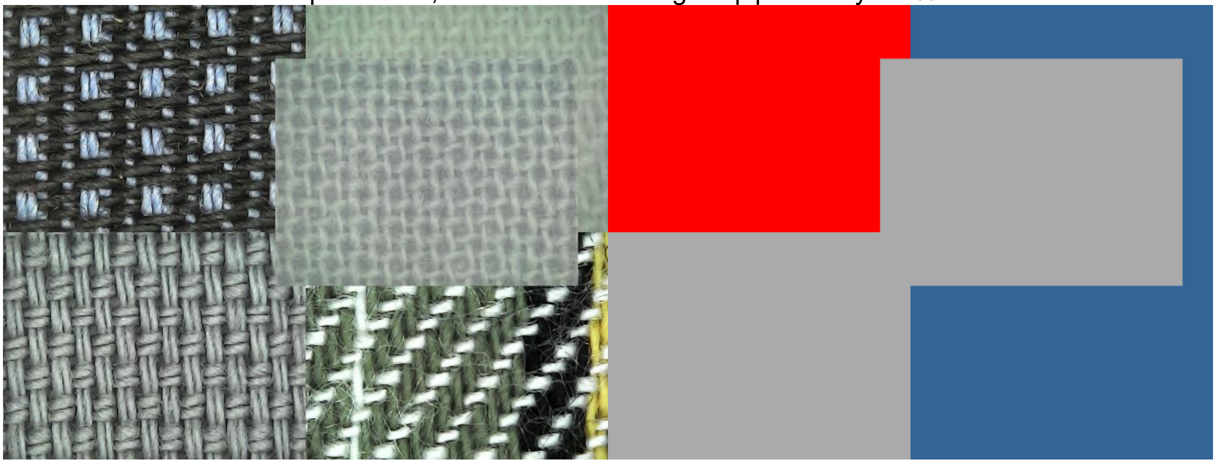

Figure 2: Input image and mask image for texture image segmentation

OPTINAL DUTIES FOR EXTRA POINT FOR FINAL

a. **(20p) Deep learning version:** write and 1D CNN and/or LSTM based architectures for **PM_980 DATASET and combined dataset**. Split data as stratified, 80% for training and %20 for test. Train CNN and LSTM and save models. Run trained model on test data and measure the test time. Plot confusion matrix.
b. **(50p)** If you have collected an original dataset and can prove it, submit a zip file containing the dataset you applied, a report describing the collection of the dataset, and the ML method.