Tutorial Assignment II - Classification

Instructions: Please read the below instructions carefully to complete this assignment.

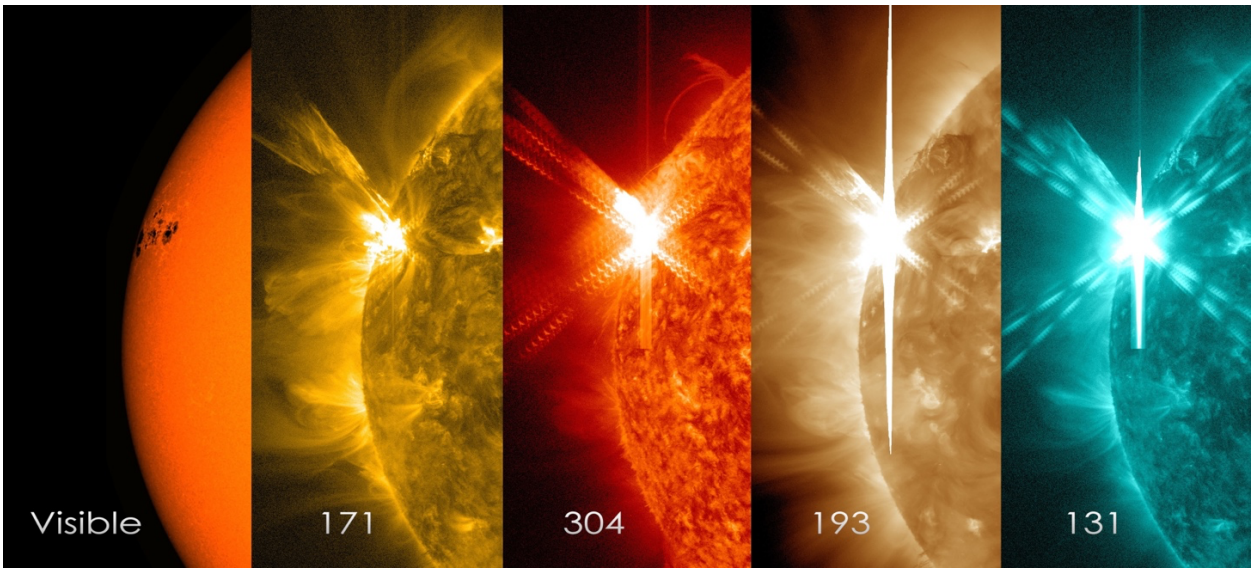
1. This assignment has only one part and will be graded out of 25 points.
2. Please work on this assignment individually.
3. Please submit all your code in one python (*.py) file and report in one pdf (*.pdf) file. Name the file as <*your_mac_id*>.py and <*your_mac_id*>_9.pdf
4. Submit these files on Avenue.
5. Please DO NOT upload ANY datasets. The datasets you upload will NOT be used for assignment evaluation.
6. Please import only the below libraries in your code. The assignment will NOT be graded if any other libraries are used besides the ones listed below. Notice that you may use the sci-kit learn library for this challenge.

```
import os
import sys
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn
import sklearn
```

7. The coding assignment will be evaluated on both **functionality** and **quality**.
    a. Functionality refers to the correctness of solution, sanity checks, stress tests, robustness, catching edge cases, comprehensiveness in code architecture, conciseness, and performance on the held-out test set (where applicable).
    b. Code quality refers to cleanliness/ readability, well-chosen data structures, program design for optimal efficiency, and clear and meaningful comments existing for *at least* 75% of the lines of code.
    c. Quality of report refers to clear, brief, and complete report with charts.

8. Please do not hardcode any aspects of your model (hardcoding model parameters is acceptable). The model performance will be evaluated by passing the test file to your code. This means, your code should be able to take any CSV file as an input with the same feature vector as your training data and produce a graph.

9. Please do not use AI to generate your code. For this class, it will count as academic integrity violation. While AI is good at generating text, it is still not good at producing quality code for building ML models. I have observed several students make fundamental mistakes on their ML models because they prompted AI to complete their task. *Most importantly, to know whether AI has given you the correct solution, you should be able to know what the correct solution is*. Using AI to learn how to build AI is a dystopian idea that will harm your learning and mastering AI.



Context: The sun's solar flares are large bursts of electromagnetic energy that can significantly impact life on Earth. Powerful solar flares can knock out power grids, satellite systems, and all communication systems on the planet. To monitor solar activity and study such events, the United States National Oceanic and Atmospheric Administration (NOAA) operates the Geostationary Operational Environmental Satellite System (GOES); a series of geosynchronous satellites with specialized measurement instruments onboard. The GOES system provides us with solar imagery, magnetometer data, solar X-ray data, and data on high-energy solar protons that hit Earth. This data is sent to Earth and regularly updated on the GEOS server which is open for public use.

Unlike Earth, the sun's regions are not divided by countries, states, or cities. Instead, various patches on the sun are numbered by NOAA for scientific investigation purposes, and the most active patches are frequently monitored for high-intensity bursts of electromagnetic radiation. Each patch is called a HARP (HMI Active Region Patch); an enduring, coherent magnetic structure that produces an electromagnetic field. The regions provide measurable features that characterize that patch. There are two classes of solar flare events of particular interest to scientists: the M-class and the X-class. These solar flares occur in various HARP regions, and the level of energy bursts is measured on a scale.

While monitoring these patches for solar flares is helpful, it is much more useful if we can predict the next powerful burst. Predicting an upcoming solar flare 24 hours in advance can give us a little time to prevent major disasters. However, this is very challenging because solar flares are rare events. Most importantly, we do not know which features directly indicate an upcoming solar flare.

Challenge: In this tutorial, you will build an ML-based binary classification model using the data from the Helioseismic and Magnetic Imager Instrument on NASA's Solar Dynamics Observatory (SDO) satellite that captures various solar events. It is the first instrument that

continuously maps the vector magnetic field of the sun. The magnetic activity recorded using these instruments will serve as the feature set for the ML model

A major solar event is defined as a burst of GEOS X-ray flux of peak magnitude above the M1.0 level. A *positive solar event* is defined as an active HARP region that flares with a peak magnitude above the M1.0 level, as reported in the GEOS database. A *negative solar event* is an active region that does not have such an event (a flare above M1.0 level) within 24 hours. The goal of the classification model should be to train on the given data and predict whether a major solar event will occur in the next 24 hours. This means that the classifier must predict whether a given solar event is positive (indicating a flaring active region) or negative (indicating a non-flaring active region).

For this challenge, you will implement a Support Vector Machine (SVM) classifier that can distinguish between a positive and negative solar flare event. Scientists M. G. Bobra and S. Couvidat used the SVM to study solar flares and published their findings in 2015. I have uploaded the paper on Avenue for reference. If you believe the above is insufficient to understand what you must implement, feel free to read the paper. However, reading the paper is *optional*. All the information you need to solve this challenge has been provided to you in lectures, slides, readings, and this document. Reading the paper may help you build a better model, though. After building the classifier, you will evaluate it using accuracy measures, k-fold cross-validation, and on 2-different datasets.

Dataset: The dataset is gathered from the GOES data server using SunPy - an open-source software for solar physics. The data is directly accessed from the server SunPy that provides a neat interface of data structures and methods to query custom data instances.
For this assignment, I have provided you with two datasets, one dataset records the solar activity from May 1st 2010 - May 1st 2015. This is exactly the dataset on which Bobra's experiments were conducted. The second dataset records solar activity from May 1st 2020 - May 1st 2024. Both datasets have same columns, just different data entries. The following table shows the features of the dataset and the files in which they are saved.

Feature Name	Description	Column Number and Filename
HARPNUM	HARP Number of the sun patch	Column: 1 – 2 all_harps_with _noaa_ars.txt
NOAA_ARS	Corresponding NOAA assignments	
USFLUX	Total unsigned flux	Column: 1 – 18 Main feature set (FS -I) pos_features_main_timechange.npy neg_features_main_timechange.npy
MEANGAM	Mean angle of field from radial	
MEANGBT	Mean gradient of total field	
MEANGBZ	Mean gradient of vertical field	
MEANGBH	Mean gradient of the horizontal field	
MEANJZD	Mean vertical current density	
TOTUSJZ	Total unsigned vertical current	
MEANALP	Mean characteristic twist parameter	
MEANJZH	Mean current helicity	
TOTUSJH	Total unsigned vertical current	
ABSNJZH	Absolute value of the net current helicity	
SAVNCPP	Sum of the modulus of the net current per polarity	

MEANPOT	Mean photospheric magnetic free energy	
TOTPOT	Total photospheric magnetic free energy density	
MEANSHR	Mean shear angle	
SHRGT45	Fraction of area with shear greater than 45 degrees	
R_VALUE	Sum of flux near polarity inversion line	
AREA_ACR	Area of strong field pixels in the active region	
Time-Change Features	Changes in the values of the above 18 properties between times 1) 48 hours prior to 24 hours prior to the peak time, 2) 42 hours prior to 24 hours prior, 3) 36 hours prior to 24 hours prior, and 4) 30 hours prior to 24 hours prior	Column : 19-90 Time Change Feature (FS-II) pos_features_main_timechange.npy neg_features_main_timechange.npy
Historical Activity Feature	Feature that characterizes the activity history of each active region calculated by adding the scores of the 'M1.0' class events between the 48 hours to 24 hours prior to the peak time.	Column: 1 Historical Activity Feature (FS-III) pos_features_historical.npy neg_features_historical.npy
MaxMin Feature	The difference between maximum and minimum values in the 48 hours to 24 hours prior to the peak time for each parameter.	Column (1-18) Max Min Feature (FS-IV) pos_features_maxmin.npy neg_features_maxmin.npy

In addition to the above, there are 4 additional files provided:

1. The geos_data.npy contains all GOES events between the start and end date.

```
{'event_date': '2021-04-19',
 'start_time': '2021.04.19_23:19:00_TAI',
 'peak_time': '2021.04.19_23:42:00_TAI',
 'end_time': '2021.04.19_23:59:00_TAI',
 'goes_class': 'M1.1',
 'goes_latitude': -25,
 'goes_longitude': -24,
 'noaa_active_region': 12816},
```

   I have applied various checks such as eliminating any events with no solar event for a given HARP and eliminating any out of range latitudes (errors that occur during real time data logging). In addition, the server logs the data every 12 minutes, therefore, the timestamp is rounded off to nearest 12 minutes before sending url request packet to the server.
2. The data_order.npy, which corresponds to the order of observations in which the model should be trained. Please use this observation order to train *at least one* model.
3. Pos_class.npy and neg_class.npy contain information about the all the positive and negative solar events based on HARP Num, Peak flare time and class of energy burst characterizing the strength of electromagnetic field.

Goals: The two datasets have been downloaded and shared with you in the Avenue. I have also created the time change, historical activity, and max-min features that you can use. Your goal is to create an input data array, provide appropriate labels, implement the SVM model, and evaluate it. Please implement functions that:

1. Preprocess the data to prepare it for the model by:
   a. Normalizing the features.
   b. Removing missing values, if any.

   c. Assigning appropriate labels to positive and negative observations.
2. Create an input data array by concatenating all features as a single 2-D array. Write this function such that you can input any combination of feature sets and create the input data. For instance, FS-I only, (FS-I, FS-II), (FS-I, FS-II, FS-IV) are all valid feature set combinations to try.
3. Perform classification using a Support Vector Machine. Use all *relevant* and *possible* feature set combinations to build different modes and select the best-performing model.
4. Perform k-Fold Cross Validation. The output of the function should be the mean and standard deviation for all folds. Report all accuracy output for SVMs corresponding to all feature set combinations. For instance, if SVM1 is trained using FS-I, and SVM2 is trained on (FS-I, and FS-II), the mean k-fold CV is reported for SVM1 and SVM2.
5. Perform accuracy computation. Using the measure of True Skill Score (TSS), compute the accuracy of the model using the equation,

$$TSS = \frac{TP}{TP + FN} - \frac{FP}{FP + TN}$$

   TSS is a good measure to predict rare events when we have a large class imbalance in the input data.
6. Visualize the performance of the k-fold cross-validation.
7. Visualize all accuracy scores using a confusion matrix.
8. Evaluate the performance of the best feature combination from point 3, on both datasets, 2010-15 and 2020-24.
9. Answer the following questions in a short brief report with charts.
   a. Which feature set combination worked best and which feature set was worst? Report the benchmark accuracy.
   b. Does adding additional FS-III and FS-IV features improve the TSS score?
   c. Which dataset led to a better TSS score (2010 or 2020)? Show the class distribution. Why do you think that happened?