Statistical inference and machine learning (MGT-448)
Midterm Project

- Deadline: Tuesday, November 21, 2023, at 12:00 PM. Late submissions will not be accepted.
- Please complete the midterm project on your own. Any discussion of the questions with peers is not allowed.
- Upload a single .zip file on Moodle, including the implementations, necessary output files, and the report as a pdf file.

1 Logistic Regression and Naive Bayes [100 points]

This exercise pertains to both the theoretical aspects of Naive Bayes and Logistic Regression classifiers as well as their implementation and comparison.

Part 1: Logistic Regression [30 points]

In this question, we will derive the "multi-class logistic regression" algorithm and demonstrate some of its properties. We assume the dataset D is d-dimensional (with d features) with n entries.

Given a training set $\{(x_i, y_i)|i = 1, \ldots, n\}$ where $x_i \in \mathbb{R}^{d+1}$ is a feature vector and $y_i \in \{0, 1\}^k$ is a binary (one-hot) vector with k entries (classes). Note that in a one-hot vector, the corresponding class label is 1 and all other entries are 0.

Note that x_i is a vector of length $d + 1$ because we pad the d features by 1 to vectorize computing the bias, that is $x_i = [1, x'_i]$ where x'_i is the actual feature vector from D.

We want to find the parameters $w \in \mathbb{R}^{k\times(d+1)}$ that maximize the likelihood for the training set assuming a parametric model of the form:

$$p(y_i^c = 1|x_i; w) = \frac{\exp(w_c^\top x_i)}{\sum_{c'=1}^{k} \exp(w_{c'}^\top x_i)}, \quad \forall c \in \{1, 2, \ldots, k\}, \tag{1}$$

where y_i^c and w_c denote the c-th entry of y_i and c-th row of w, respectively.

As the total probability of all classes equals 1, there is no necessity to make predictions for the last class. Rather, we can conveniently calculate $p(y_i^k = 1|x_i; w)$ with the following formula:

$$p(y_i^k = 1|x_i; w) = 1 - \sum_{c'=1}^{k-1} p(y_i^{c'} = 1|x_i; w)$$

1. Rewrite the parametric model in Equation (1) using $k - 1$ weight vectors instead of k.
2. Use your suggested parametric model and derive the gradient of the log-likelihood with respect to the c-th class weight, i.e., $\frac{\partial l(w)}{\partial w_c}$, where $l(w)$ denotes the log-likelihood.

Part 2: Naive Bayes [25 points]

1. Explain why the independence assumption in Naive Bayes is crucial for simplifying the computation of the posterior probabilities.

2. We want to classify documents as being spam or not. Each document is associated with a pair (x, y) where x is a feature vector of word counts of the document and y is the label for whether it is spam ($y = 1$ if yes, $y = 0$ if no). The vocabulary is size 3 so feature vectors look like $(3, 0, 1)$, $(0, 1, 1)$, etc.

   Consider a naive Bayes model with the following conditional probability table:

word type	$P(w \mid y = 1)$	$P(w \mid y = 0)$
1	2/10	4/10
2	1/10	2/10
3	7/10	4/10

   and the following prior probabilities over classes:

$P(y = 1)$	$P(y = 0)$
3/10	7/10

   (a) Consider a document $x = (1, 0, 1)$. Which class has the highest posterior probability? Show mathematically.

   (b) Now suppose that we have a new document whose label we do not know. What is the probability that a word in this document is word type 1?

Part 3: Implementation and Comparison [45 points]

1. Explain how to formulate a Naive Bayes classifier for multi-class classification where the features are continuous. Implement it from scratch.

2. Implement your multi-class logistic regression. You are allowed to use any library for that (sklearn is recommended).

3. Use the provided Jupyter file (Question1.ipynb) and load the dataset to train both models. Split the data into training and testing sets with an 80-20 ratio respectively and compare their performance in terms of accuracy, precision, and recall for both training and testing sets.

2 Decision Trees in Medical Diagnostics [100 Points]

You are given a dataset called *synthetic_medical_data.csv* containing synthetic medical records of patients. The dataset includes the following features:

- **Age**: Represents the age of the patient. (Numeric value ranging from 20 to 90 years)
- **Pain_Level**: A subjective score indicating the level of pain experienced by the patient. (Numeric value on a scale of 0 - no pain to 10 - extreme pain)
- **Cholesterol**: Indicates the cholesterol level in the patient's blood. It is an important measure as higher levels might indicate potential health risks. (Numeric value ranging from 100 to 300 mg/dL)
- **Marker_Presence**: A binary value representing the presence (1) or absence (0) of a certain medical marker in the patient.
- **Blood_Pressure**: Represents the systolic blood pressure measurement of the patient. Elevated blood pressure can be a sign of potential cardiovascular issues. (Numeric value ranging from 90 to 180 mmHg)
- **Diagnosis**: The target variable. Indicates whether the patient was diagnosed with a particular disease (1) or not (0) based on the aforementioned features.

The goal is to predict the diagnosis of a patient based on the provided data.

Part 1: Data Exploration [15 Points]

1. [**5 points**] How many patients have been diagnosed with the disease, and how many have not?
2. [**10 points**] Compute mean, median, and standard deviation for age, pain level, and cholesterol.

Part 2: Theoretical Understanding [35 Points]

1. [**15 points**] Briefly explain the concept of entropy and information gain along their mathematical formulation.
2. [**20 points**] Gini impurity is another criterion for splitting a decision tree. Describe its formulation and the difference between the Gini impurity and entropy as criteria for splitting in a decision tree.

Part 3: Model Implementation and Evaluation [50 Points]

1. [**25 points**] Split the dataset into a training set and a testing set using an 80-20 split. Train a decision tree classifier using the Gini impurity as the criterion. Experiment with different values for the maximum depth of the tree: for instance, try using *max_depth* values of 3, 5, and unrestricted growth. Visualize the learned decision tree for *max_depth* = 3. You also need to upload your implementation.
2. [**15 points**] Report the accuracy of your model on both the training and testing datasets for each of the *max_depth* values you experimented with.
3. [**10 Points**] Based on the accuracy scores from the training and testing datasets, can you identify any signs of overfitting? Specifically, for which (if any) *max_depth* values do you believe overfitting is occurring?

You are allowed to use packages for this question. Recommended packages are:

```
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeClassifier, plot_tree
from sklearn.metrics import accuracy_score
```

3 Support Vector Machine [100 points]

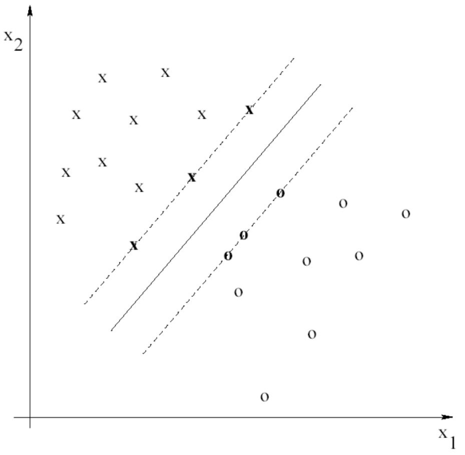
Part 1: LOOCV [15 points]

Leave-one-out cross-validation (LOOCV) is a method of model validation which involves training a model N times for a dataset containing N observations. In each iteration, $N-1$ observations are used as the training set and a single observation is held out as the test set. The process is repeated such that each observation is used once as the test set. The LOOCV estimate of model performance is the average error over all N trials, mathematically given by:

$$LOOCV = \frac{1}{N}\sum_{i=1}^{N} E_i, \tag{2}$$

where E_i is the prediction error on the **test set** in the i-th iteration.

1. [**5 Points**] What is the LOOCV error estimate for maximum margin SVM in Figure 1? In this figure, data samples are distributed throughout the feature space, with each class denoted by 'x' and 'o' symbols, respectively.



2. [**10 Points**] Construct an example with two classes such that data points are linearly separable but LOOCV is not zero.

Part 2: Duality [40 points]

Consider the optimization problem for an SVM with a 2-norm margin as follows:

$$\begin{aligned} \underset{w,b,\zeta}{\text{minimize}} \quad & \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{N}\zeta_i^2 \\ \text{subject to} \quad & y^{(i)}(w^T x^{(i)} + b) \geq 1 - \zeta_i,\ i = 1, \ldots, N \\ & \zeta_i \geq 0 \end{aligned} \tag{3}$$

1. [**15 Points**] Show that conditions $\zeta_i \geq 0$ are redundant and do not change the solution of the optimization problem (3).

2. [**10 Points**] Write the Lagrangian form and take its derivatives with respect to parameters w, b, and ζ_i.

3. [**15 Points**] Drive the KKT conditions and write the Dual problem.

Part 3: SVM Implementation [45 points]

Complete *SVM.ipynb* Jupyter notebook.