CLASSIFICATION AND REGRESSION USING SUPERVISED LEARNING

TIM AJAR KECERDASAN BUATAN

AGENDA

SUPERVISED VS UNSUPERVISED

CLASSIFICATION METHODS

DATA PREPROCESSING METHODS

LABEL ENCODING

LOGISTIC REGRESSION CLASSIFIER

NAÏVE BAYES

SVM

SUPERVISED VS UNSUPERVISED LEARNING

Supervised learning refers to the process of building a machine learning model that is based on labeled training data.

- In supervised learning, each example or row is a tuple consisting of input variables and a desired target variable

Unsupervised learning refers to the process of building a machine learning model without relying on labeled training data.

- In some sense, it is the opposite of supervised learning. Since there are no labels available, you need to extract insights based on just the data given to you. With unsupervised learning, we are training a system where separate datapoints will potentially be separated into multiple clusters or groups
- an unsupervised learning algorithm needs to separate the given dataset into several groups in the best way possible

SUPERVISED LEARNING

- a common dataset used in machine learning is the "Titanic" dataset. This dataset contains features to describe the passengers of the famous ship RMS Titanic.
- Some of the input features are:
  - Passenger name
  - Cabin class
  - Age
  - Place of embarkment
- And the target variable in this case would be whether the passenger survived or not

UNSUPERVISED LEARNING

- Assume you have a set of portraits of people. The people in this set are a very diverse group of men and women and you have all kinds of nationalities, ages, body weights, and so on.
- Initially, you put the dataset through an unsupervised learning algorithm. In this case, without any a priori knowledge, the unsupervised algorithm will start classifying these photographs depending on some feature that it recognizes as similar.
- For example, on its own, it might start recognizing that men and women are different, and it might start clustering the men in one group and the women in another.
- But there is no guarantee that it will find that pattern. It might cluster the images because some portraits have a dark background and others have a light background, which would likely be a useless inference.

MAIN TASK

SUPERVISED LEARNING

- Classification
- Regression

UNSUPERVISED LEARNING

- Clustering
- Association
- Dimentionality reduction

WHAT IS CLASSIFICATION?

- we will discuss supervised classification techniques
- The classification process is a technique used to arrange data into a fixed number of categories so that it can be used effectively and efficiently.
- In machine learning, classification is used to identify the category to which a new datapoint belongs.
- A classification model is built based on the training dataset containing datapoints and the corresponding labels.
- For example, let's say that we want to determine whether a given image contains a person's face or not.
    - We would build a training dataset containing classes corresponding to two classes: face and no-face. A model would then be trained based on the available training samples. The trained model can then be used for inference.

PREPROCESSING DATA

- machine learning algorithms expect data to be formatted in a certain way before the training process can begin
- In order to prepare the data for ingestion by machine learning algorithms, the data must be preprocessed and converted into the right format.
- preprocessing techniques example:
  - Binarization
  - Mean removal
  - Scaling
  - Normalization

BINARIZATION

```
import numpy as np
from sklearn import preprocessing

input_data = np.array([[5.1, -2.9, 3.3],[-1.2, 7.8, -6.1],
 [3.9, 0.4, 2.1],[7.3, -9.9, -4.5]])
# Binarize data
data_binarized = preprocessing.Binarizer(threshold=2.1).transform(input_data)
print("\nBinarized data:\n", data_binarized)
```

Output:

```
Binarized data:
 [[1. 0. 1.]
 [0. 1. 0.]
 [1. 0. 0.]
 [1. 0. 0.]]
```

MEAN REMOVAL

```
# Print mean and standard deviation
print("\nBEFORE:")
print("Mean =", input_data.mean(axis=0))
print("Std deviation =", input_data.std(axis=0))

# Remove mean
data_scaled = preprocessing.scale(input_data)
print("\nAFTER:")
print("Mean =", data_scaled.mean(axis=0))
print("Std deviation =", data_scaled.std(axis=0))
```

- Output

```
BEFORE:
Mean = [ 3.775 -1.15  -1.3  ]
Std deviation = [3.12039661 6.36651396 4.0620192 ]

AFTER:
Mean = [1.11022302e-16 0.00000000e+00 2.77555756e-17]
Std deviation = [1. 1. 1.]
```

SCALING

- The MinMaxScaler algorithm:

$$\frac{x_i - \min(x)}{\max(x) - \min(x)}$$

```
# Min max scaling
data_scaler_minmax = preprocessing.MinMaxScaler(feature_range=(0, 1))
data_scaled_minmax = data_scaler_minmax.fit_transform(input_data)
print("\nMin max scaled data:\n", data_scaled_minmax)
```

- Output:

```
Min max scaled data:
 [[0.74117647 0.39548023 1.        ]
 [0.         1.         0.        ]
 [0.6        0.5819209  0.87234043]
 [1.         0.         0.17021277]]
```

NORMALIZATION

- We use the process of normalization to modify the values in the feature vector so that we can measure them on a common scale. In machine learning, we use many different forms of normalization.
- Some of the most common forms of normalization aim to modify the values so that they sum up to 1.
- L1 normalization, which refers to Least Absolute Deviations, works by making sure that the sum of absolute values is 1 in each row.
- L2 normalization, which refers to least squares, works by making sure that the sum of squares is 1.

NORMALIZATION (CONT.)

```
# Normalize data
data_normalized_l1 = preprocessing.normalize(input_data, norm='l1')
data_normalized_l2 = preprocessing.normalize(input_data, norm='l2')
print("\nL1 normalized data:\n", data_normalized_l1)
print("\nL2 normalized data:\n", data_normalized_l2)
```

- Output

```
L1 normalized data:
 [[ 0.45132743 -0.25663717  0.2920354 ]
 [-0.0794702   0.51655629 -0.40397351]
 [ 0.609375    0.0625      0.328125  ]
 [ 0.33640553 -0.4562212  -0.20737327]]

L2 normalized data:
 [[ 0.75765788 -0.43082507  0.49024922]
 [-0.12030718  0.78199664 -0.61156148]
 [ 0.87690281  0.08993875  0.47217844]
 [ 0.55734935 -0.75585734 -0.34357152]]
```

LABEL ENCODING

- When performing classification, we usually deal with lots of labels. These labels can be in the form of words, numbers, or something else.
- Many machine learning algorithms require numbers as input. So, if they are already numbers, they can be directly used for training. But this is not always the case.
- Labels are normally words, because words can be understood by humans.
- Training data is labeled with words so that the mapping can be tracked.
- To convert word labels into numbers, a label encoder can be used. Label encoding refers to the process of transforming word labels into numbers.

WHAT IS REGRESSION?

- Regression is the process of estimating the relationship between input and output variables.
- One item to note is that output variables are continuous-valued real numbers. Hence, there are an infinite number of possibilities.
- This is in contrast with classification, where the number of output classes is fixed.
- The classes belong to a finite set of possibilities. In regression, it is assumed that the output variables depend on the input variables, so we want to see how they are related.
- Consequently, the input variables are called independent variables, also known as predictors, and output variables are called dependent variables, also known as criterion variables.
- It is not necessary that the input variables are independent of one another; indeed, there are a lot of situations where there are correlations between input variables.

LOGISTIC REGRESSION CLASSIFIER

- Logistic regression is a technique that is used to explain the relationship between input variables and output variables.
- Regression can be used to make predictions on continuous values, but it can also be useful to make discrete predictions where the result is True or False, for example, or Red, Green, or Yellow as another example.
- The input variables are assumed to be independent and the output variable is referred to as the dependent variable.
- The dependent variable can take only a fixed set of values. These values correspond to the classes of the classification problem.

NAÏVE BAYES CLASSIFIER

- Naïve Bayes is also known as a probabilistic classifier since it is based on Bayes' Theorem.

$$P(Y|X) = \frac{P(X \text{ and } Y)}{P(X)}$$

- These probabilities are denoted as the prior probability and the posterior probability.
- The prior probability is the initial probability of an event before it is contextualized under a certain condition, or the marginal probability.
- The posterior probability is the probability of an event after observing a piece of data.

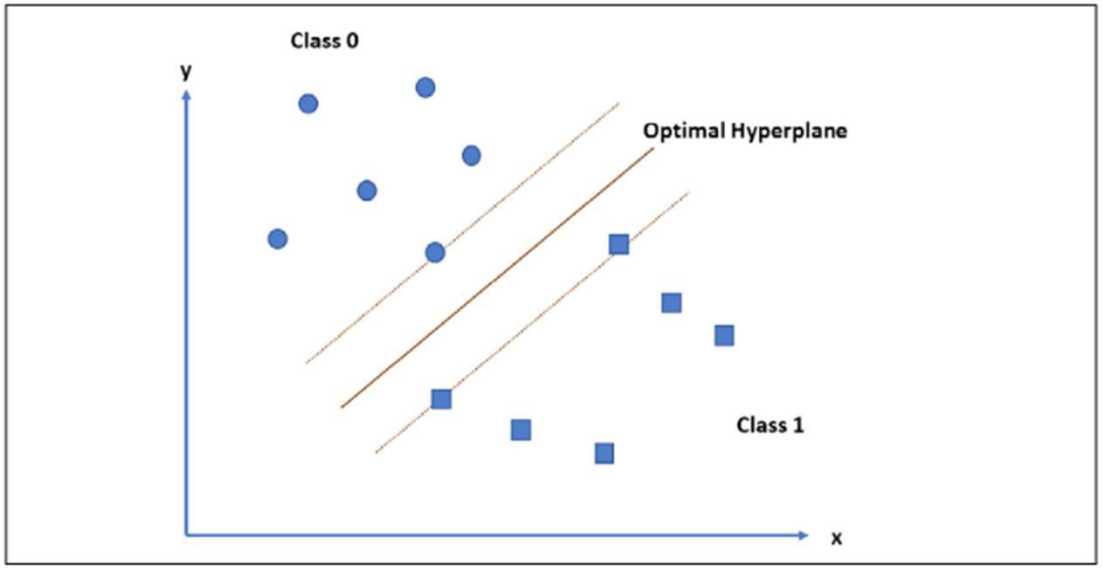
EXAMPLE

Nomor	OUTLOOK	TEMPERATURE	HUMIDITY	WINDY	PLAY
1	sunny	hot	high	false	no
2	sunny	hot	high	true	no
3	cloudy	hot	high	false	yes
4	rainy	mild	high	false	yes
5	rainy	cool	normal	false	yes
6	rainy	cool	normal	true	yes
7	cloudy	cool	high	true	yes
8	sunny	mild	high	false	no
9	sunny	cool	normal	false	yes
10	rainy	mild	normal	false	yes
11	sunny	mild	normal	true	yes
12	cloudy	mild	high	true	yes
13	cloudy	hot	normal	false	yes
14	rainy	mild	high	true	no

OUTLOOK: rainy
TEMPERATURE: hot
HUMIDITY: high
WINDY: true
PLAY: ?

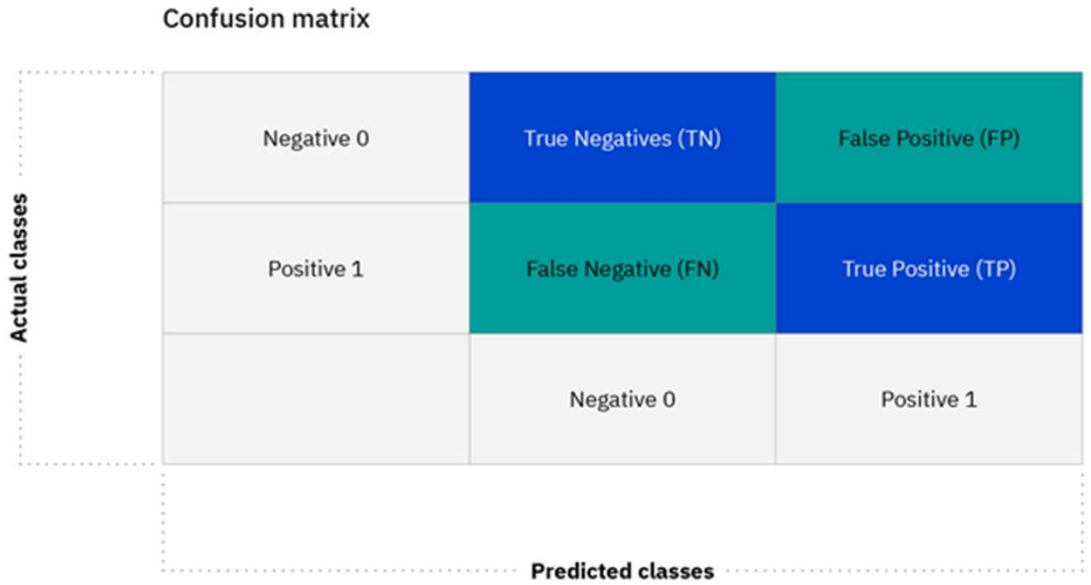
SUPPORT VECTOR MACHINE

- A Support Vector Machine (SVM) is a classifier that is defined using a separating hyperplane between the classes.
- This hyperplane is the N-dimensional version of a line. Given labeled training data and a binary classification problem, the SVM finds the optimal hyperplane that separates the training data into two classes.

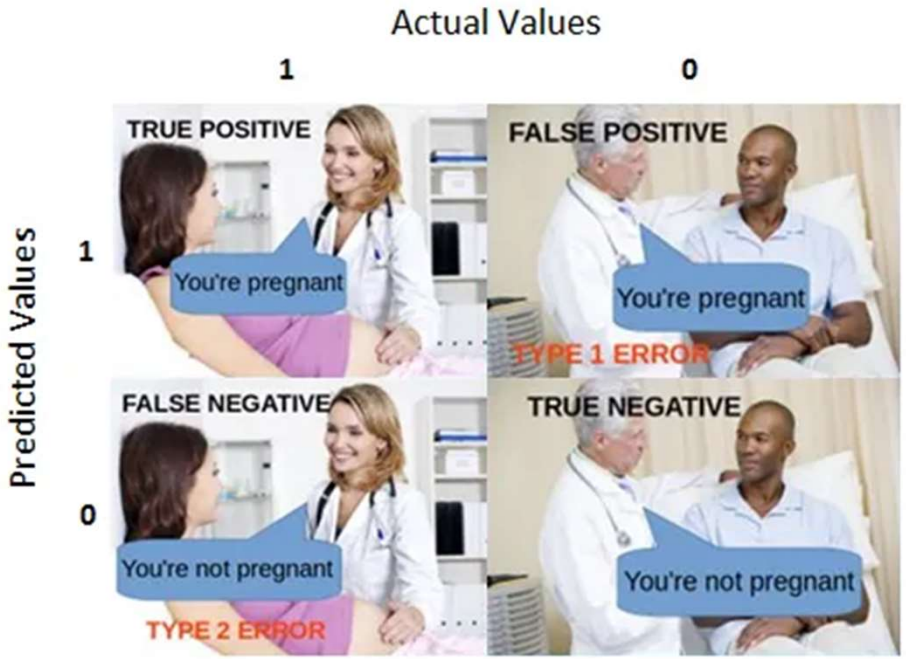


CONFUSION MATRIXES

- A confusion matrix is a figure or a table that is used to describe the performance of a classifier. Each row in the matrix represents the instances in a predicted class and each column represents the instances in an actual class.



CONFUSSION MATRIXES (CONT.)



Confusion Matrix [Image 3] (Image courtesy: My Photoshopped Collection)

https://towardsdatascience.com/understanding-confusion-matrix-a9ad42dcfd62

CLASSIFICATION MEASURE

- Accuracy

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

Accuracy simply measures how often the classifier makes the correct prediction. It's the ratio between the number of correct predictions and the total number of predictions.

- Precision

$$\text{Precision} = \frac{TP}{TP + FP}$$

It is a measure of correctness that is achieved in true prediction. In simple words, it tells us how many predictions are *actually positive* out of all the *total positive predicted*.

CLASSIFICATION MEASURE (CONT.)

- Recall $$\text{Recall} = \frac{TP}{TP + FN}$$

- It is a measure of **actual observations** which are predicted **correctly**, i.e. how many observations of positive class are actually predicted as positive. It is also known as **Sensitivity**. *Recall* is a valid choice of evaluation metric when we want to capture *as many positives* as possible.

- F-measure/F1-score $$\text{F1-Score} = 2* \frac{(\text{Recall}*\text{Precision})}{(\text{Recall} + \text{Precision})}$$

The **F1 score** is a number between **0 and 1** and is the *harmonic mean of precision and recall*. We use harmonic mean because it is not sensitive to extremely large values, unlike simple averages.

THANK YOU