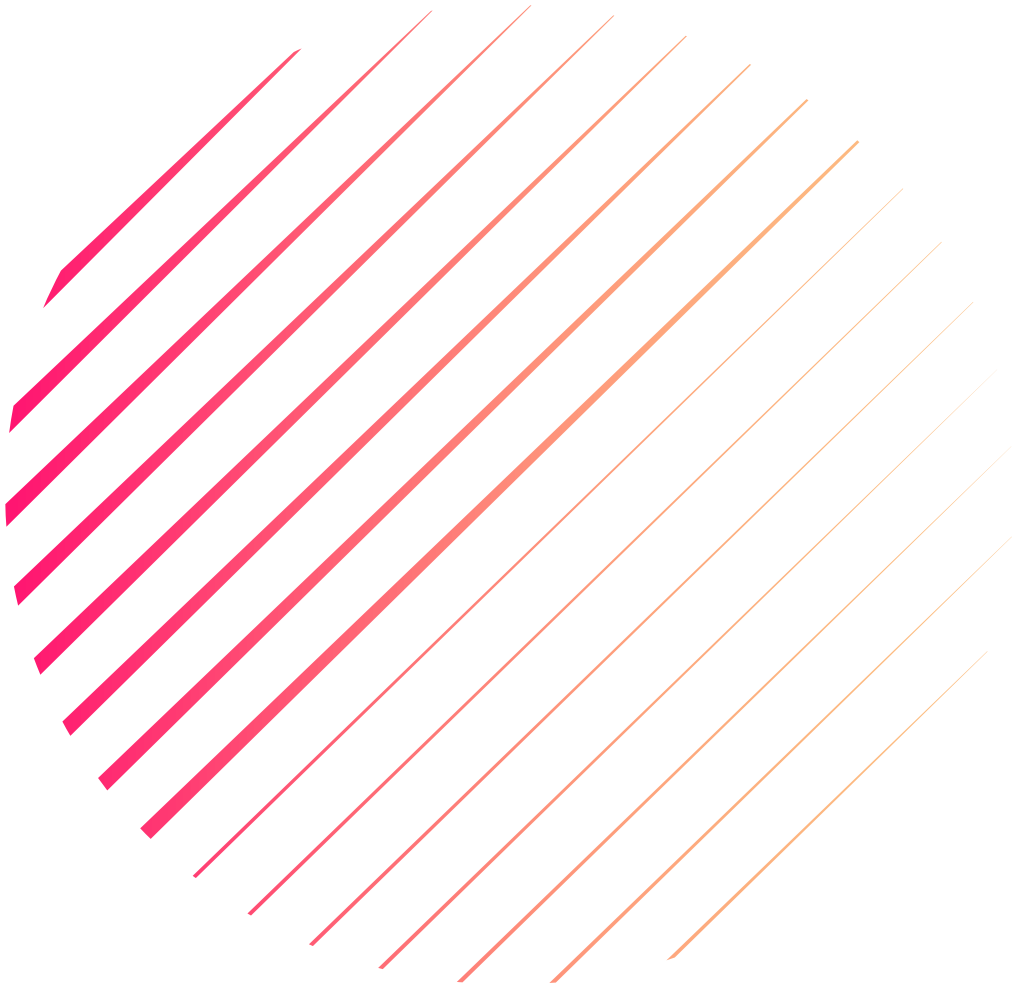
[Decision tree – Logistic regression] classifiers



Anas Hamed 4989

Decision Tree

- ## How it works?

The basic idea behind any decision tree algorithm is as follows:

1. Select the best attribute using Attribute Selection Measures (ASM) to split the records.
2. Make that attribute a decision node and breaks the dataset into smaller subsets.
3. Starts tree building by repeating this process recursively for each child until one of the conditions will match:
    - All the tuples belong to the same attribute value.
    - There are no more remaining attributes.
    - There are no more instances.

- ## Attribute selection measures

Attribute selection measure is a heuristic for selecting the splitting criterion that partition data into the best possible manner. It is also known as splitting rules because it helps us to determine breakpoints for tuples on a given node. ASM provides a rank to each feature (or attribute) by explaining the given dataset.

Best score attribute will be selected as a splitting attribute (Source). In the case of a continuous-valued attribute, split points for branches also need to define. Most popular selection measures are Information Gain, Gain Ratio, and Gini Index.

We used Gini index as our splitting criterion. The Gini Index considers a binary split for each attribute. You can compute a weighted sum of the impurity of each partition. If a binary split on attribute A partitions data D into D1 and D2, the Gini index of D is:

$$\text{Gini}(D) = 1 - \sum_{i=1}^{m} \text{Pi}^2$$

$$\text{Gini}_A(D) = \frac{|D1|}{|D|}\text{Gini}(D_1) + \frac{|D2|}{|D|}\text{Gini}(D_2)$$

- Decision tree classifier

Importing Required Libraries

```
from sklearn.tree import DecisionTreeClassifier
```

Loading data and feature selection

Data is loaded and divided in the dataReader() class and provides train_df and test_df. Feature selection is CountVectorizer which is used as a pre-processing step.

Building decision model

```
model = DecisionTreeClassifier()
model.fit(train_table, train_df['label'])
predictions = model.predict(test_table)
```

Logistic Regression

- What is logistic regression and how it works?

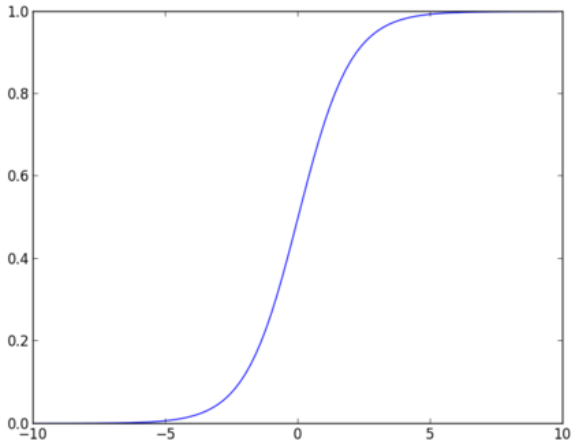
Logistic regression is a statistical method for predicting binary classes. The outcome or target variable is dichotomous in nature. Dichotomous means there are only two possible classes. For example, it can be used for cancer detection problems. It computes the probability of an event occurrence.

It is a special case of linear regression where the target variable is categorical in nature. It uses a log of odds as the dependent variable. Logistic Regression predicts the probability of occurrence of a binary event utilizing a logit function.

Sigmoid Function

The sigmoid function, also called logistic function gives an ‘S’ shaped curve that can take any real-valued number and map it into a value between 0 and 1. If the curve goes to positive infinity, y predicted will become 1, and if the curve goes to

negative infinity, y predicted will become 0. If the output of the sigmoid function is more than 0.5, we can classify the outcome as 1 or YES, and if it is less than 0.5, we can classify it as 0 or NO. The outputcannotFor example: If the output is 0.75, we can say in terms of probability as: There is a 75 percent chance that patient will suffer from cancer.



- Logistic regression classifier

Importing Required Libraries

```
import sklearn.linear_model as lm
model = lm.LogisticRegression()
```

Loading data and feature selection

Data is also loaded and divided in the dataReader() class and provides train_df and test_df. Feature selection is CountVectorizer which is used as a pre-processing step.

Building decision model

```
model.fit(train_table, train_df['label'])
predictions = model.predict(test_table)
```

Decision Tree Vs. Logistic Regression

CRITERIA	LOGISTIC REGRESSION	DECISION TREE CLASSIFICATION
Interpretability	Less interpretable	More interpretable
Decision Boundaries	Linear and single decision boundary	Bisects the space into smaller spaces
Ease of Decision Making	A decision threshold has to be set	Automatically handles decision making
Overfitting	Not prone to overfitting	Prone to overfitting
Robustness to noise	Robust to noise	Majorly affected by noise
Scalability	Requires a large enough training set	Can be trained on a small training set