Python for Machine Learning

Xiaojuan Zhu
Research Computing Support
517 Greve Hall
xzhu8@utk.edu

1

Supervised Machine Learning

- **Supervised:** you train the machine using data which is well "labeled." It means some data is already tagged with the correct answer. It can be compared to learning which takes place in the presence of a supervisor or a teacher.
- **Unsupervised learning:** a machine learning technique, where you do not need to supervise the model. Instead, you need to allow the model to work on its own to discover information. It mainly deals with the unlabelled data.
- Two categories of algorithms in supervised machine learning:
  - **Regression:** A regression problem is when the output variable is a real value, such as "dollars" or "weight".
  - **Classification:** A classification problem is when the output variable is a category, such as "red" or "blue" or "disease" and "no disease".

- https://www.youtube.com/watch?v=f_uwKZIAeM0
- https://www.youtube.com/watch?v=cfj6yaYE86U

Classification Methods-Logistic regression

- Logistic regression is a classification algorithm used to assign observations to a discrete set of classes. Unlike linear regression which outputs continuous number values, logistic regression transforms its output using the logistic sigmoid function to return a probability value which can then be mapped to two or more discrete classes.
- Reference: https://ml-cheatsheet.readthedocs.io/en/latest/logistic_regression.html

Decision Tree

- **Decision Tree:** a decision support tool that uses a tree-like graph or model of decisions and their possible consequences, including chance event outcomes, resource costs, and utility. It is one way to display an algorithm that only contains conditional control statements. Reference: https://medium.com/greyatom/decision-trees-a-simple-way-to-visualize-a-decision-dc506a403aeb

K-Nearest Neighbors

- **K-Nearest Neighbors:** The KNN algorithm assumes that similar things exist in close proximity. In other words, similar things are near to each other.

  KNN Algorithm:

  1. Initialize K to your chosen number of neighbors
  2. For each example in the data
     1. Calculate the distance between the query example and the current example from the data.
     2. Add the distance and the index of the example to an ordered collection
  3. Sort the ordered collection of distances and indices from smallest to largest (in ascending order) by the distances
  4. Pick the first K entries from the sorted collection
  5. Get the labels of the selected K entries

  Reference:https://towardsdatascience.com/machine-learning-basics-with-the-k-nearest-neighbors-algorithm-6a6e71d01761

Classification Methods

- **Linear Discriminant Analysis**: a dimensionality reduction technique used as a preprocessing step in Machine Learning and pattern classification applications. The main goal of dimensionality reduction techniques is to reduce the dimensions by removing the redundant and dependent features by transforming the features from higher dimensional space to a space with lower dimensions. Reference:https://medium.com/@srishtisawla/linear-discriminant-analysis-d38decf48105
- **Gaussian Naive Bayes**
- It is a classification technique based on Bayes' Theorem with an assumption of independence among predictors. In simple terms, a Naive Bayes classifier assumes that the presence of a particular feature in a class is unrelated to the presence of any other feature. Reference: https://machinelearningmastery.com/naive-bayes-for-machine-learning/

Random Forest

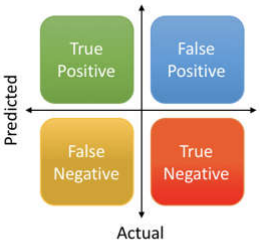
- **Random Forest Algorithm:** Fits a bunch of trees on "random samples from our sample" (called bootstrap samples) & they all vote on best class. The votes aggregated to choose the winning class. The concept of combining predictions like this is called "bagging" or Bootstrap Aggregation.
- Details:
  - Randomly select (with replacement) both N observations and a subset of predictors to create 100-500 subsets of data.
  - Fit a "bushy" tree to each sample e.g. no pruning so each WILL overfit!
  - At each split, variables are randomly sampled
  - Have each model make a prediction, then count (when classifying) or average (when regressing) their predictions (weights may be used based on model accuracy)

Neural Networks

- **Neural Network Algorithm**
  - Simulates human brain by weighting "neurons" stored in "hidden layers"
  - Each training observation adjusts the impact of each neuron, often through "back-propogation"
  - Deep Learning uses many layers and many neurons per layer
- **Pros:** 1) Excellent results for extreme complexity e.g. voice, image recognition 2) Though the model consists of a set of equations that is fairly small compared to many other methods 3) Model is quick to apply to new data
- **Cons:** 1) Computationally intensive to train the model 2) Performance on numerical data rarely much better than faster methods e.g. rf, gbm 3) Models are impossible to interpret 4) Extremely sensitive to multicollinearity (use PCA first or method = "pcaNNet")
- **Neural Network Tuning Parameters**
  - Depends on specific type but in general:
  - Number of hidden layers.
  - Number of neurons per layer
  - Type of feedback mechanism e.g. back-propogation

Model Evaluation

- **Confusion matrix:** a table to describe the performance of a classification model (or "classifier") on a set of test data for which the true values are known. It allows the visualization of the performance of an algorithm.
- **Accuracy:** the ratio of number of correct predictions to the total number of input samples.
- **Precision:** the number of correct positive results divided by the number of all predicted positive results (e.g. How many of the mushrooms we predicted would be edible actually were?).
- **Recall**: the number of correct positive results divided by the number of actual positive results that should have been returned (e.g. How many of the mushrooms that were poisonous did we accurately predict were poisonous?).
- The **F1** score is a measure of a test's accuracy. It considers both the precision and the recall of the test to compute the score. The F1 score can be interpreted as a weighted average of the precision and recall, where an F1 score reaches its best value at 1 and worst at 0.



Unsupervised Learning

- **Unsupervised Learning**: when dealing with real-world problems, most of the time, data will not come with predefined labels, so we will want to develop machine learning models that can classify correctly the data, by finding by themselves some commonality in the features, that will be used to predict the classes on new data.
- the goal is to study the intrinsic (and commonly hidden) structure of the data. This techniques can be condensed in two main types of problems that unsupervised learning tries to solve. This problems are:
  - **Clustering**
  - **Dimensionality Reduction**

Clustering Analysis

- Cluster analysis or clustering is the task of grouping a set of objects in such a way that objects in the same group (called a cluster) are more similar (in some sense) to each other than to those in other groups (clusters).
- Clustering, however, has many different names (with respect to the fields it is being applied):
  - **Cluster analysis**
  - **Automatic classification**
  - **Data segmentation**

K-means Cluster

K-means clustering aims to partition n observations into k clusters in which each observation belongs to the cluster with the nearest mean, serving as a prototype of the cluster (Definition from Wiki).

How the K-means algorithm works

To process the learning data, the K-means algorithm in data mining starts with a first group of randomly selected centroids, which are used as the beginning points for every cluster, and then performs iterative (repetitive) calculations to optimize the positions of the centroids It halts creating and optimizing clusters when either: The centroids have stabilized — there is no change in their values because the clustering has been successful. The defined number of iterations has been achieved.

reference: https://towardsdatascience.com/understanding-k-means-clustering-in-machine-learning-6a6e67336aa1

Hierarchical clustering

- **How Hierarchical Clustering Works**
- Hierarchical clustering starts by treating each observation as a separate cluster. Then, it repeatedly executes the following two steps:

  (1) identify the two clusters that are closest together, and

  (2) merge the two most similar clusters. This continues until all the clusters are merged together. This is illustrated in the dendrogram below.

Dendrogram

- A dendrogram is a diagram that shows the hierarchical relationship between objects. It is most commonly created as an output from hierarchical clustering. The main use of a dendrogram is to work out the best way to allocate objects to clusters.
- The key to interpreting a dendrogram is to focus on the height at which any two objects are joined together. In the example below, we can see that 4 and 22 are most similar, as the height of the link that joins them together is the smallest.
- Observations are allocated to clusters by drawing a horizontal line through the dendrogram. Observations that are joined together below the line are in clusters.
- Reference: https://www.displayr.com/what-is-dendrogram/

Density Based Scan Clustering (DBSCAN)

- The key idea is that for each point of a cluster, the neighborhood of a given radius has to contain at least a minimum number of points. Compared to centroid-based clustering like K-Means, density-based clustering works by identifying "dense" clusters of points, allowing it to learn clusters of arbitrary shape and identify outliers in the data.
- **DBSCAN algorithm requires two parameters:**
  - **eps** : It defines the neighborhood around a data point i.e. if the distance between two points is lower or equal to 'eps' then they are considered as neighbors. If the eps value is chosen too small then large part of the data will be considered as outliers. If it is chosen very large then the clusters will merge and majority of the data points will be in the same clusters. One way to find the eps value is based on the k-distance graph.
  - **MinPts**: Minimum number of neighbors (data points) within eps radius. Larger the dataset, the larger value of MinPts must be chosen. As a general rule, the minimum MinPts can be derived from the number of dimensions D in the dataset as, MinPts >= D+1. The minimum value of MinPts must be chosen at least 3. In this algorithm, we have 3 types of data points.
  - **Core Point**: A point is a core point if it has more than MinPts points within eps. **Border Point**: A point which has fewer than MinPts within eps but it is in the neighborhood of a core point. **Noise or outlier**: A point which is not a core point or border point.
- Reference: https://www.geeksforgeeks.org/dbscan-clustering-in-ml-density-based-clustering/

Gaussian Mixture Model

- A Gaussian mixture model is a probabilistic model that assumes all the data points are generated from a mixture of a finite number of Gaussian distributions with unknown parameters.
- The GaussianMixture object implements the expectation-maximization (EM) algorithm for fitting mixture-of-Gaussian models. It can also draw confidence ellipsoids for multivariate models, and compute the Bayesian Information Criterion to assess the number of clusters in the data.
- Reference: https://scikit-learn.org/stable/modules/mixture.html