Heuristics

Lab Section 8

Today's Lab

- We will explore:
- No assignment today
  - If you understand everything below you can leave
- Discuss Scikit Learn for machine learning
- Example.py for machine learning on the iris dataset
- Auction Project Description
- Auction Project Requirement

Introduction – Scikit Learn

- Scikit-learn (Sklearn) is the most useful and robust library for machine learning in Python. The primary library for machine learning.
- It provides a selection of efficient tools for machine learning and statistical modeling including classification, regression, clustering and dimensionality reduction via a consistence interface in Python.
- This library, which is largely written in Python, is built upon NumPy, SciPy and Matplotlib.
- To properly explore this library: You must have basic knowledge of Machine Learning. The best course is Andre Ng (pioneer in machine learning) course.

Installation - Scikit Learn

- If you already installed NumPy and Scipy, Following command can be used to install scikit-learn via pip –
- For windows: pip install -U scikit-learn
- For mac: pip3 install -U scikit-learn

- If you do not have Scipy or Numpy installed, use these commands to install them:
- pip install scipy
  - For mac, replace pip with pip3
- pip install numpy
  - For mac, replace pip with pip3

Scikit Learn – Features (Focus)

- Rather than focusing on loading, manipulating and summarising data, Scikit-learn library is focused on modeling the data.
- Some of the most popular groups of models provided by Sklearn are as follows –
- **Supervised Learning algorithms** – Almost all the popular supervised learning algorithms, like Linear Regression, Support Vector Machine (SVM), Decision Tree etc., are the part of scikit-learn.
- **Unsupervised Learning algorithms** – On the other hand, it also has all the popular unsupervised learning algorithms from clustering, factor analysis, PCA (Principal Component Analysis) to unsupervised neural networks.

Scikit Learn - Modelling Process

- **We will talk about different processes involved in Modeling the problem.**
  - **We will also different machine learning terminologies while exploring the modeling processes**
- Dataset Loading
- A collection of data is called a dataset. It is having the following two components
- **Features** – The variables of data are called its features. They are also known as predictors, inputs or attributes.
- **Feature matrix** – It is the collection of features, in case there is more than one.
- **Feature Names** – It is the list of all the names of the features.

Scikit Learn - Modelling Process - contd

- **Target** – It is the output variable that basically depends upon the feature variables. They are also known as reponse, label or output.
- **Target Vector** – It is used to represent the target column. Generally, we have just one target column.
- **Target Names** – These represents the possible values taken by a target vector.
- This ends the data loading process.
- Next process is: Splitting the dataset

Scikit Learn - Splitting the dataset - contd

- To check the accuracy of our model, we can split the dataset into two pieces- **a training set** and **a testing set**.
  - Use the training set to train the model and testing set to test the model. After that, we can evaluate how well our model did.
- To achieve splitting, we have **train_test_split()** function of scikit-learn to split the dataset. This function has the following main arguments –
- **X, y** – Here, **X** is the **feature matrix** and y is the **target vector**, which need to be split.
- **test_size** – This represents the ratio of test data to the total given data. For example, if we set **test_data = 0.3** for 150 rows of X. It will produce test data of 150*0.3 = 45 rows. How much training set: 105??
- **random_size** – It is used to guarantee that the split will always be the same. This is useful in situations where you want reproducible results.
- There are other attributes, you can explore here.

Scikit Learn - Modelling Process - contd

- Train the Model
- Next, we can use our dataset to train some prediction-model. As discussed, scikit-learn has a wide range of **Machine Learning (ML) algorithms** that have a consistent interface for fitting, predicting accuracy, recall etc.
- Model Persistence – why we need it, model traning takes a long time
- Once you train the model, it is desirable that the model should be persist for future use so that we do not need to retrain it again and again. It can be done with the help of dump and load features of joblib package.
- Consider the code below in which we will be saving the trained model (classifier_best) for future use
  - from sklearn.externals import joblib
  - joblib.dump(classifier_best, 'best_so_far.joblib')
- The above code will save the model into file named best_so_far.joblib. Now, the object can be reloaded from the file with the help of the following code –
  - joblib.load(best_so_far.joblib')

Scikit Learn - Modelling Process - contd

- Performance evaluation of the Model
- To evaluate the model, We have sklearn.metrics available, Which offers main metrics of :
  - Accuracy
    - Accuracy classification score. the set of labels predicted for a sample must *exactly* match the corresponding set of labels in y_true.
  - Precision
    - The precision is the ratio tp / (tp + fp). The precision is intuitively the ability of the classifier not to label as positive a sample that is negative.
  - Recall
    - The recall is the ratio tp / (tp + fn). The recall is intuitively the ability of the classifier to find all the positive samples.
  - The best value is 1 and the worst value is 0.
  - Classification report
    - Build a text report showing the main classification metrics.

Scikit Learn - Example

- Explore IRIS Dataset and run/evaluate example.py with two different models.
- The **Iris Datasets** is a multivariate data set that contains four features including the length and width of sepals and petals of 50 samples of three species of **Iris.**
  - These Three species are ***Iris setosa, Iris virginica, and Iris versicolor.***
- The **data set** is often used in **data** mining, classification, and clustering examples and to test algorithms.
  - It is one of the most common datasets that I came across in my journey into the data science world
- Open mydata.csv and check out the target.
- Now here you can see a lot of 0,1 and 2. Don't Panic if you don't know what all these zeros and one and two means…!
  - If the sample belongs from 'setosa' class then the target will be 0.
  - If the sample belongs from 'versicolor' class then the target will be 1.
  - If the sample belongs from 'virginica' class then the target will be 2.

Auction Project Description

- Inter-process communication game.
- You are given 100 units of budget (100 dollars/dirhams).
- There are four triplets of items.
  - 'Picasso', 'Van_Gogh', 'Rembrandt', 'Da_Vinci'
- They will come up in random order.
- You will bid for items.
- Whoever wins the bid gets the item
- If someone gets a triplet earlier than another item, then that person wins.
- So there will be a server (the auctioneer), two or more clients.
- Auctioneer receives bids and tells the clients who wins or loses.

Auction Project Requirement

- The game is based on client-server architecture with multiple clients as bidders and the server as auctioneer.
- You don’t need to worry about the client and server, the code will be provided with instructions on how to run it on Monday. (clientzmq.py and serverzmq.py on brightspace)
- All you need to worry about is about coming up with the best strategy to win.
  - Implement a function that when called gives bid.
  - The idea is that players are bidding for items of different types. The first player who obtains all needed items while staying within budget wins.
  - In each bidding round, players determine a bid and send it in.
  - (If two highest bids are the same, then the first one received gets it.)
  - Each player is told who received the item and for how much.
  - If you win the item in a round, then you pay that amount.
  - Otherwise, you pay nothing.
  - Using this information, determine what to do next.

Auction Project Information available

- At any round we have this information available:
- Items - the item being sold in a specific round.
- winner - the winner of the item sold in a specific round.
- winner amount, the amount of money paid for the item sold in a specific round.
- Players, the names of the current players.
- Artists, names of the artists (paintings) that are for sale in our auction.
- Standings, how many paintings and money the other players has.
- the current round.

End