Machine Learning Summary

Notebook:	4th Year Project		
Created:	2018-09-29 8:27 PM	**Updated:**	2018-11-11 9:29 PM
Author:	moflanker@gmail.com		
Tags:	Machine Learning		
URL:	https://www.udemy.com/machinelearning/learn/v4/t/lecture/6403742?start=0		

Data Preprocessing

1. Importing Data
2. Missing Data
    - How to deal with missing data, different techniques
        - mean
        - median
        - most_frequent
        - Python library (from sklearn.preprocessing import Imputer)
3. Categorical Data
    - How to deal with categorial data and convert them to numbers
        - Label Encoding
        - One Hot Encoding
        - Python library (from sklearn.preprocessing import LabelEncoder, OneHotEncoder)
4. Splitting the dataset into Training set and Test set
    - Split data into train set to train model and test set to test model efficiency
    - This idea is implemented more efficiently later on K-Fold cross validation
    - from sklearn.cross_validation import train_test_split
5. Feature Scaling
    - To have a better predictive model, feature scaling is implemented where data is scaled down between -1, 1 with respect to standard deviation and mean or maximum and minimum
    - Standardization and Normalization
    - Usually implemented within ML libraries

Regression

Regression models (both linear and non-linear) are used for predicting a real value, like salary for example. If your independent variable is time, then you are forecasting future values, otherwise your model is predicting present

but unknown values. Regression technique vary from Linear Regression to SVR and Random Forests Regression.

In this part, you will understand and learn how to implement the following Machine Learning Regression models:

1. Simple Linear Regression
    - from sklearn.linear_model import LinearRegression
2. Multiple Linear Regression
    - Optimization: backward elimination technique based on variable statistics contribution to the predicted value
    - from sklearn.linear_model import LinearRegression
3. Polynomial Regression
    - from sklearn.preprocessing import PolynomialFeatures
4. Support Vector for Regression (SVR)
    - Library doesn't support feature scaling
    - from sklearn.svm import SVR
5. Decision Tree Regression
    - from sklearn.tree import DecisionTreeRegressor
6. Random Forest Regression
    - from sklearn.ensemble import RandomForestRegressor

1. What are the pros and cons of each model ?

Regression

Regression Model	Pros	Cons
Linear Regression	Works on any size of dataset, gives informations about relevance of features	The Linear Regression Assumptions
Polynomial Regression	Works on any size of dataset, works very well on non linear problems	Need to choose the right polynomial degree for a good bias/variance tradeoff
SVR	Easily adaptable, works very well on non linear problems, not biased by outliers	Compulsory to apply feature scaling, not well known, more difficult to understand
Decision Tree Regression	Interpretability, no need for feature scaling, works on both linear / nonlinear problems	Poor results on too small datasets, overfitting can easily occur
Random Forest Regression	Powerful and accurate, good performance on many problems, including non linear	No interpretability, overfitting can easily occur, need to choose the number of trees

Machine Learning A-Z © SuperDataScience

2. How do I know which model to choose for my problem ?

First, you need to figure out whether your problem is **linear or non linear.** You will learn how to do that in Part 10 - Model Selection. Then:

If your problem is linear, you should go for **Simple Linear Regression** if you only have **one feature**, and **Multiple Linear Regression** if you have **several**

features.

If your problem is **non linear**, you should go for **Polynomial Regression, SVR, Decision Tree or Random Forest.** Then which one should you choose among these four ? That you will learn in Part 10 **- Model Selection.** The method consists of using a very relevant technique that **evaluates your models performance**, called **k-Fold Cross Validation**, and then picking the model that shows the **best results**. Feel free to jump directly to Part 10 if you already want to learn how to do that.

3. How can I improve each of these models ?

In Part 10 - Model Selection, you will find the second section dedicated to Parameter **Tuning**, that will allow you to improve the performance of your models, by tuning them. You probably already noticed that each model is composed of two types of parameters:

- the parameters that are learnt, for example the coefficients in Linear Regression,
- the hyperparameters.

The **hyperparameters** are the parameters that are **not learnt** and that **are fixed values** inside the model equations. For example, the regularization parameter lambda or the penalty parameter C are hyperparameters. So far we used the default value of these hyperparameters, and we haven't searched for their optimal value so that your model reaches even higher performance. **Finding their optimal value is exactly what Parameter Tuning is about.** So for those of you already interested in improving your model performance and doing some parameter tuning, feel free to jump directly to Part 10 - Model Selection.

Model Selection

Type of Problems:
No dependent value - **Cluster** problem
Dependent value

- continuous - **regression** problem
- non-continuous, category - **classification** problem

K-Fold Cross Validation

- iterate through data and split them into n sections (n folds)
- each n is used as a test set based on all other n training set

Grid Search

- hyperparameters selection
- run a list of different values of hyperparameters
- grid search uses cross validation in its computations

Next step to be learned

Classification

Unlike regression where you predict a continuous number, you use classification to predict a category. There is a wide variety of classification applications from medicine to marketing. Classification models include linear models like Logistic Regression, SVM, and nonlinear ones like K-NN, Kernel SVM and Random Forests.

In this part, you will understand and learn how to implement the following Machine Learning Classification models:

1. Logistic Regression
2. K-Nearest Neighbors (K-NN)
3. Support Vector Machine (SVM)
4. Kernel SVM
5. Naive Bayes
6. Decision Tree Classification
7. Random Forest Classification

Follow Classification I am learning:

- Clustering
  - K-Means Clustering
  - Hierarchical Clustering
- Association Rule Learning
  - Apriori
  - Eclat
- Reinforcement Learning
  - Upper Confidence Bound
  - Thompson Sampling
- Natural Language Processing
- Deep Learning
  - Artificial Neural Networks
  - Conventional Neural Networks
- Dimensionality Reduction
  - Principal Component Analysis PCA
  - Linear Discriminant Analysis LDA
  - Kernel PCA