Advanced Machine Learning
Practical Lecture

Part I. Exercise

1. A student wants to know how to do well for ML exam. He collects those old statistics and decides to use decision trees to get his model. He now gets 9 data points, and two features: "whether stay up late before exam" (S) and "whether attending all the classes" (A). We already know the statistics is as below:
Set (all) = [5+, 4−]
Set (S+) = [3+, 2−], Set (S−) = [2+, 2−]
Set (A+) = [5+, 1−], Set (A−) = [0+, 3−]

Suppose we are going to use the feature to gain most information at first split, which feature should we choose? How much is the information gain?
You may use the following approximations:

N	3	5	7
$\log_2(N)$	1.58	2.32	2.81

2. A French Region wants to understand the investment differences in several towns and villages. To study this problem, several criteria was taking into account:

- Existence of a Train Station: yes/no
- Status: Town/Village
- Distance to Paris: close/far
- Investment level: small/medium/high

2.1 Build a Decision Tree using like Class the attribute Investment, using the following data:

Station	**Status**	**Distance**	**Investment**
No	Village	Close	Medium
Yes	Town	Close	High
No	Village	Far	Small
Yes	Village	Close	High
No	Town	Far	Small
Yes	Town	Far	Medium
No	Town	Close	High

2.2 If a Village has a train station and it is situated far from Paris, predict the investment level.

3. What linear function is used by a SVM for classification? How is an input vector x_i (instance) assigned to the positive or negative class?
4. If the training examples are linearly separable, how many decision boundaries can separate positive from negative data points? Which decision boundary does the SVM algorithm calculates? Why?

5. What is the margin? Which are the equations of the two margin hyperplans H_+ and H_- ?

Part II. Practical Lecture (graded)

1. Training and Visualizing a Decision Tree

1.1. First, let's load the iris dataset from sci-kit learn library. Plot the data set.

1.2. Use the function train_test_split(X,y,test_size=0.3, random_state=42) and DecisionTreeClassifier() to train the dataset using the Decision Tree Model.

Compute the accuracy score by writing your own function.

2. Decision Trees parameters

There are many hyperparameters that a decision tree classifier has. Analyze the meaning of theses parameters.

Will you use ***gini*** index or ***entropy*** index. Try the both and explain the results.

3. Visualize and save the Tree

We can also save the decision tree which our model has built. You need to have Graph Viz and pydotplus installed in your system.

```
from sklearn.externals.six import StringIO
from sklearn.tree import export_graphviz
import pydotplus
dot_data = StringIO()
export_graphviz(model, out_file=dot_data)
graph = pydotplus.graph_from_dot_data(dot_data.getvalue())
graph.write_pdf("iris.pdf")

from IPython.display import Image
Image(filename="./images/tree.png")
```

Another way to visualize decision trees is to install the **dtreeviz** package :

```
!pip install dtreeviz
```

Use the function dtreeviz to plot the tree.

You can plot the decision boundary on the scatter of points.

Create the function plot_decision_boundary(clf, x, y) where **clf** is the Decision Tree classifier, **X** is the dataset and **y** is the target.

4. Estimating Class Probabilities

To estimate the probability of an instance belongs to a class, you can use predict_proba, to determine the class that an instance will be assigned to use predict.

5. Regression

Decision trees can be used for regression tasks too. Instead of predicting a class, in regression tasks, the aim is to predict a numeric value.

- Load the diabetes dataset and use the DecisionTreeRegressor function to predict the diabetes evolution (target variable).
- Compute the MSE error and compare it if using the Linear Regression. Conclude.

7. Support Vector Machine

On the iris dataset use the SVC function and train_test_split to classify and predict the class on test data. Compute the accuracy and compare it to those obtained using Decision Tree.

Normalize the data using StandardScaler() and apply again the SVC. How the accuracy changes?

8. Kernel trick.

The kernel parameter of the SVC function allows to define different kernels functions as : Radial Basis Function Kernel, Polynomial Kernel, Sigmoid kernel, etc.

Try several of them by comparing the obtained accuracy score.