Welcome to Supervised Learning - script

Welcome to Week 3&4 - Course 1

As an old saying goes, if you have a hammer, everything looks like a nail.
And this is exactly what you will get by the end of this week - you will get a Machine Learning Hammer!
This hummer is Neural Networks combined with TensorFlow - a powerful library for Machine Learning, open-sourced by Google in November 2015.
Chances are that you might have heard about, or used these tools before, but if not, then I think that the best time and place to learn about these tools would be NOW and HERE, that is, the second week of our course.

So, here is what you will learn in this week.
Let's start with our first lesson of this week.
First, you will get familiar with our main tool, TensorFlow.
We start with a review of how TensorFlow implements what is called a computational graph.
Then we will open our first TensorFlow Jupyter notebook, and see how these things work, by looking at very simple operations.
Next, you will see how a simple Linear Regression can be implemented in TensorFlow, and compare it with how it is done with other Python packages.
After that, you will learn how Neural Networks are organized, and how Linear Regression is nothing but a very special and simple Neural Network.
You will also see why Neural Networks can be seen as a Machine Learning Hammer.
After you learn about Neural Networks, you will learn about methods used for their training.
And here you will meet some of main powerhorses of modern Machine Learning, namely the Gradient Descent method, and it relatives, such as Stochastic Gradient Descent.
We will see their working both in theory and in practice, that is, in slides and in in Jupyter notebooks.
You will see how it works first for a Linear Regression, and then for a Neural Network regression implemented in TensorFlow.
All this will be done using some simple non-financial data, so that we will not get lost navigating in several new topics simultaneously.

All these topics will be presented in the first lesson.
Then, after we get such a powerful hammer, we will start to look for financial nails.
And the first nail you will see is a very classical financial problem, namely the problem of prediction of Earnings Per Share, or EPS for short.
Once we have our tools such as TensorFlow and other packages such as Scikit-Learn and Statsmodels, we will be able to try both Linear and Non-Linear, Neural Network Regression on this problem.
Your homework for this week will include assignments where you will extend Jupyter notebooks for the EPS analysis that I am going to show you.
Therefore, by the end of this week, you will be already using TensorFlow in practice!

This will be our second lesson in this week.

After that, we will turn to classification models.
We will start with looking at how Machine Learning deals with probabilistic models.
We will all about Maximum Likelihood Estimation and related methods such as a Maximum A-Posteriori method,
We will also talk about the notion of relative entropy, also known as a KL-divergence, which is very important for understanding of Machine Learning.
Then you will see how all this machinery applies to classification problems.
You will learn about one of the most popular approach to classification called Logistic Regression.
You will also learn how it can be implemented in Scikit-Learn and TensorFlow.
You will see how Logistic Regression is just another special case of our general hummer called Neural Networks.

And finally, our hummer will find its second financial nail.
This will be a problem of predicting bank failures.
As you will see, our hummer would be pretty good for this type of nails.

So, this is what you will master this week.
I hope that, by the end of this week, you will find yourself exponentially more knowledgeable, at least conceptually, on both the theory and practice of modern Machine Learning.
We will reconvene again at the end of this week to talk about what we will have learned and what we will be doing in the rest of the course and the Specialization, but for now, let's get started!