**Extreme Gradient Boosting with XGBoost**

Do you know the basics of supervised learning and want to use state-of-the-art models on real-world datasets? Gradient boosting is currently one of the most popular techniques for efficient modeling of tabular datasets of all sizes. XGboost is a very fast, scalable implementation of gradient boosting, with models using XGBoost regularly winning online data science competitions and being used at scale across different industries. In this course, you'll learn how to use this powerful library alongside pandas and scikit-learn to build and tune supervised learning models. You'll work with real-world datasets to solve classification and regression problems.

#### 01 - Classification with XGBoost

**1. Welcome to the course!**

Hi, my name is Sergey Fogelson and I'm the instructor for Datacamp's course on Gradient Boosted Trees With XGBoost. I'm a data scientist working in the media industry and have used XGBoost extensively on a variety of machine learning problems. I've created this course with DataCamp to help others quickly understand how to use this very popular implementation of gradient boosting. Let's get started.

**2. Before we get to XGBoost...**

In order to understand XGBoost, we need to have some handle on the broader topics of supervised classification, decision trees, and boosting, which we will cover throughout this chapter. To begin, let's briefly review what

**3. Supervised learning**

supervised learning is and the kinds of problems its methods can be applied to. At its core, supervised learning, which is the kind of learning problems that XGBoost can be applied to, relies on labeled data. That is, you have some understanding of the past behavior of the problem you're trying to solve or what you're trying to predict.

**4. Supervised learning example**

For example, assessing whether a specific image contains a person's face, is a classification problem. Here the training data are images converted into vectors of pixel values, and the labels are either 1 when the image contains a face or 0 when the image doesn't contain a face. Given this, there are two kinds of supervised learning problems that account for the vast majority of use-cases: classification problems and regression problems. We will only talk about classification problems here and leave regression to chapter 2.

**5. Supervised learning: Classification**

Classification problems involve predicting either binary or multi-class outcomes.

**6. Binary classification example**

For example, predicting whether a person will purchase an insurance package given some quote is a binary supervised learning problem,

**7. Multi-class classification example**

and predicting whether a picture contains one of several species of birds is a multi-class supervised learning problem. When dealing with binary supervised learning problems,

**8. AUC: Metric for binary classification models**

the AUC, or Area Under the Receiver Operating Characteristic Curve, is the most versatile and common evaluation metric used to judge the quality of a binary classification model. It is simply the probability that a randomly chosen positive data point will have a higher rank than a randomly chosen negative data point for your learning problem. So, a higher AUC means a more sensitive, better performing model. When dealing with multi-class classification problems,

**9. Accuracy score and confusion matrix**

it is common to use the accuracy score (higher is better) and to look at the overall confusion matrix to evaluate the quality of a model.

**10. Review**

Some common algorithms for classification problems include logistic regression and decision trees. If you want a deeper review, check out DataCamp's introductory course on supervised learning.

**11. Other supervised learning considerations**

All supervised learning problems, including classification problems, require that the data is structured as a table of feature vectors, where the features themselves (also called attributes or predictors) are either numeric or categorical. Furthermore, it is usually the case that numeric features are scaled to aid in either feature interpretation or to ensure that the model can be trained properly (for example, numerical feature scaling is essential to ensure properly trained support vector machine models). Categorical features are also almost always encoded before applying supervised learning algorithms, most commonly using one-hot encoding. Finally, other kinds of supervised learning problems exist, so I'll mention them here briefly.

**12. Ranking**

Ranking problems involve predicting an ordering on a set of choices (like google search suggestions),

**13. Recommendation**

and recommendation problems involve recommending an item or set of items to a user based on his/her consumption history and profile (like Netflix).

**14. Let's practice!**

Now that you've been reminded about the basics of classification problems, let's get to work!

#### 02 - Regression with XGBoost

#### 03 - Fine-tuning your XGBoost model

#### 04 - Using XGBoost in pipelines