Course Learning Objectives

Print to PDF

Contents

By the end of this course, you will be able to:

- Describe supervised learning and its suitability for various tasks.
- Explain key machine learning concepts such as classification, regression, overfitting, and the trade-off in model complexity.
- Identify appropriate data preprocessing techniques for specific scenarios, provide reasons for their selection, and integrate them into machine learning pipelines.
- Develop an intuitive understanding of common machine learning algorithms.
- Build end-to-end supervised machine learning pipelines using Python and scikit-learn on real-world datasets.

Lecture Learning Objectives

Below are specific lecture learning objectives.

Introduction and Course Information

- be able to explain the motivation to study machine learning;
- be able to differentiate between supervised and unsupervised learning;
- know how to navigate through the course material.

Lecture 1: Terminology, Baselines, Decision Trees

- differentiate between classification and regression problems;

- explain machine learning terminology such as features, targets, predictions, training, and error;
- use `DummyClassifier` and `DummyRegressor` as baselines for machine learning problems;
- explain the `fit` and `predict` paradigm and use `score` method of ML models;
- broadly describe how decision tree prediction works;
- use `DecisionTreeClassifier` and `DecisionTreeRegressor` to build decision trees using `scikit-learn`;
- explain the difference between parameters and hyperparameters;
- explain the concept of decision boundaries.

Lecture 2: Machine Learning Fundamentals

- explain how decision boundaries change with the `max_depth` hyperparameter;
- explain the concept of generalization;
- split a dataset into train and test sets using `train_test_split` function;
- explain the difference between train, validation, test, and "deployment" data;
- identify the difference between training error, validation error, and test error;
- explain cross-validation and use `cross_val_score` and `cross_validate` to calculate cross-validation error;
- explain overfitting, underfitting, and the fundamental tradeoff;
- state the golden rule.

Lecture 3: k-nearest neighbours (k-NNs), support vector machines (SVMs) with RBF kernel

- explain the notion of similarity-based algorithms;
- broadly describe how k-NNs use distances;
- discuss the effect of using a small/large value of the hyperparameter k when using the k-NN algorithm;
- describe the problem of curse of dimensionality;
- explain the general idea of SVMs with RBF kernel;
- explain the differences between k-NNs and SVM RBFs;

- broadly describe the relation of `gamma` and `C` hyperparameters with the fundamental tradeoff.

Lecture 4: Preprocessing and pipelines

- identify when to implement feature transformations such as imputation, scaling, and one-hot encoding in a machine learning model development pipeline;
- use `sklearn` for applying feature transformations on your dataset;
- discuss golden rule in the context of feature transformations;
- use `sklearn.pipeline.Pipeline` to build a preliminary machine learning pipeline;
- use `ColumnTransformer` to build all our transformations together into one object and use it with `sklearn` pipelines.

Lecture 5: More on categorical features and encoding text data

- explain `handle_unknown="ignore"` hyperparameter of `scikit-learn`’s `OneHotEncoder`;
- identify when it’s appropriate to apply ordinal encoding vs one-hot encoding;
- explain strategies to deal with categorical variables with too many categories;
- explain why text data needs a different treatment than categorical variables;
- use `scikit-learn`’s `CountVectorizer` to encode text data;
- explain different hyperparameters of `CountVectorizer`.

Lecture 6: Hyperparameter optimization and optimization bias

- explain the need for hyperparameter optimization
- carry out hyperparameter optimization using `sklearn`’s `GridSearchCV` and `RandomizedSearchCV`
- explain optimization bias
- identify and reason when to trust and not trust reported accuracies

Lecture 7: Naive Bayes

- Explain the naive assumption of naive Bayes.
- Predict targets by hand on toy examples using naive Bayes.
- Use `scikit-learn`'s `MultiNomialNB`, `BernoulliNB`, and `GaussianNB`.
- Use `predict_proba` for different classifiers and explain its usefulness.
- Explain the need of smoothing in naive Bayes.
- Explain how `alpha` controls the fundamental tradeoff.
- Use naive Bayes for multi-class classification.
- Name advantages and disadvantages of naive Bayes.

Lecture 8: Linear models and multi-class, meta-strategies

- Explain the general intuition behind linear models
- Explain the predict paradigm of linear models
- Use `scikit-learn`'s `LogisticRegression` classifier
    - Use `fit`, `predict`, `predict_proba`
    - Use `coef_` to interpret the model weights
- Compare logistic regression with naive Bayes
- Explain the advantages and limitations of linear classifiers
- Carry out multi-class classification using OVR and OVO strategies.