



Reading: Course 6 resources and citations

Module 1: The different types of machine learning

Resources

Case study: The Woobles: The power of recommendation systems to drive sales

- [The Woobles](#)
- [Google Analytics](#)
- [Triple Whale](#)

Find solutions online

- [Seaborn documentation](#)
- [Stack Overflow](#)
- [Cross Validated](#)
- [Data Science](#)
- [Mathematics](#)
- [Kaggle](#)
- [competitions](#)
- [Gemini Code Assist](#)
- [Stack Exchange site list](#)
- [Asking questions on Stack Overflow](#)
- [Kaggle educational content](#)

Citations

Reference guide: Python for machine learning

- [Project Jupyter](#). (n.d.). Jupyter.
- [Python.org](#). (n.d.). Python.

Python libraries and packages

- [Matplotlib — Visualization with Python](#). (n.d.). Matplotlib.
- [NumPy](#). (n.d.). NumPy.
- [Pandas - Python data analysis library](#). (n.d.). Pandas.
- [Plotly: Low-code data app development](#). (n.d.). Plotly.
- [Scikit-learn: Machine learning in Python](#). (n.d.). Scikit-learn.
- [Seaborn: Statistical data visualization](#). (n.d.). Seaborn.pydata.

Module 2: Workflow for building complex models

Resources

Explore feature engineering

- [MinMaxScaler documentation](#)
- [StandardScaler documentation](#)

More about imbalanced datasets

- [Google Developers](#)
- [imbalanced-learn](#)
- [RandomOverSampler](#)
- [Upsampling methods](#)
- [Downsampling methods](#)

Naive Bayes classifiers

- Scikit-learn: [BernoulliNB](#)
- Scikit-learn: [CategoricalNB](#)
- Scikit-learn: [ComplementNB](#)
- Scikit-learn: [GaussianNB](#)
- Scikit-learn: [MultinomialNB](#)

More about evaluation metrics for classification models

- [Evaluation metrics for classification models in scikit-learn](#)

Citations

Feature engineering and class balancing

- [Python 3.13.5 documentation](#). (n.d.). Python.

Construct a Naive Bayes model with Python

- [Python 3.13.5 documentation](#). (n.d.). Python.
- [Scikit-learn: machine learning in Python](#). (n.d.). Scikit-learn

Activity: Build a K-means model

- Gorman, K. B., Williams, T. D., & Fraser, W. R. (2014, March 5). [Ecological sexual dimorphism and environmental variability within a community of Antarctic penguins](#) (Genus Pygoscelis). PLoS ONE, 9(3), e90081.
- [Scikit-learn: Machine learning in Python](#). (n.d.). Scikit-learn.

Module 3: Unsupervised learning techniques

Resources

More about K-means

- [KMeans documentation](#)
- [Arthur, D.; Vassilvitskii, S. \(2007\). "k-means++: the advantages of careful seeding" \(PDF\).](#)
[Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms. Society for Industrial and Applied Mathematics Philadelphia, PA, USA. pp. 1027–1035](#)

Clustering beyond K-means

- [DBSCAN addendum transcript](#)
- [DBSCAN](#)
- [Agglomerative clustering](#)
- [Agglomerative clustering transcript](#)

More about inertia and silhouette coefficient metrics

- [scikit-learn documentation for silhouette_score](#)
- [Academic paper](#)

Citations

Use K-means for color compression with Python

- Brownlee, J. (2014, July 17). [What is the difference between test and validation datasets?](#)
Machine Learning Mastery.
- [Exsilio Solutions]. (2016, September 9). Accuracy, precision, recall and F1 score: Interpretation of performance measures. [Exsilio Solutions](#)
- Hans. (2012, May 9). [Tulip yellow](#). Pixabay.
- Liberman, N. (2020, May 22). [Decision trees and random forests - Towards data science](#). Medium.
- [Python 3.13.5 documentation](#). (n.d.). Python.

Apply inertia and silhouette score with Python

- [Python 3.13.5 documentation](#). (n.d.). Python.

Module 4: Tree-based modeling

Resources

Hyperparameter tuning

- [scikit-learn documentation for DecisionTreeClassifier](#)

Bagging: How it works and why to use it

- [Kaggle lesson on bagging](#)
- [Academic paper](#)
- [Bagging classifier documentation](#)

- [Bagging regressor documentation](#)

More about random forests

- [Random forest classifier](#)
- [Random forest regressor](#)

Reference guide: Random forest tuning

- [Model metrics](#)
- [Random forest classifier](#)
- [Random forest regressor](#)
- [accuracy_score](#)
- [average_precision_score](#)
- [confusion_matrix](#)
- [f1_score](#)
- [fbeta_score](#)
- [metrics.log_loss](#)
- [multilabel_confusion_matrix](#)
- [precision_recall_curve](#)
- [precision_score](#)
- [recall_score](#)
- [roc_auc_score](#)
- [mean_absolute_error](#)
- [mean_squared_error](#)
- [mean_squared_log_error](#)
- [median_absolute_error](#)
- [mean_absolute_percentage_error](#)
- [r2_score](#)

Reference guide: Validation and cross-validation

- [PredefinedSplit](#)
- [GridSearchCV](#)
- [train_test_split](#)
- [scikit-learn cross-validation documentation](#)
- [developers.google.com - Validation Sets](#)

Case Study: Machine learning model unearths resourcing insights for Booz Allen Hamilton

- [Booz Allen Hamilton](#)

More about gradient boosting

- [Gradient boosting with scikit-learn](#)
- [Gradient boosted decision trees in developers.google.com](#)

Reference guide: XGBoost tuning

- [scikit-learn model metrics](#)
- [XGBoost classifier](#)
- [XGBoost Regressor](#)
- [Notes on parameter tuning from XGBoost](#)
- [XGBoost parameters](#)
- [accuracy_score](#)
- [confusion_matrix](#)
- [f1_score](#)
- [fbeta_score](#)
- [average_precision_score](#)
- [metrics.log_loss](#)
- [multilabel_confusion_matrix](#)
- [precision_recall_curve](#)
- [precision_score](#)
- [recall_score](#)
- [roc_auc_score](#)
- [mean_absolute_error](#)
- [mean_squared_error](#)
- [mean_squared_log_error](#)
- [median_absolute_error](#)
- [mean_absolute_percentage_error](#)
- [r2_score](#)

Citations

Identify: Parts of the decision tree

- [What is a boolean?](#) (2021, February 1). Computer Hope.
- Verma, R.. (2009). [Decision trees](#). Hypertext Bookshop.

Build a decision tree with Python

- [Python 3.13.5 documentation](#). (n.d.). Python.
- [Scikit-learn: Machine learning in Python](#). (n.d.). Scikit-learn.

Tune a decision tree

- [Python 3.13.5 documentation](#). (n.d.). Python.

Tune and validate decision trees with Python

- [Python 3.13.5 documentation](#). (n.d.). Python.

Build and cross-validate a random forest model with Python

- [Python 3.13.5 documentation](#). (n.d.). Python.
- [Scikit-learn: Machine learning in Python](#). (n.d.). Scikit-learn.

Build and validate a random forest model using a validation dataset

- [Python 3.13.5 documentation](#). (n.d.). Python.

Build an XGBoost model with Python

- [Python 3.13.5 documentation](#). (n.d.). Python.
- [Scikit-learn: Machine learning in Python](#). (n.d.). Scikit-learn.
- [XGBoost documentation](#). (n.d.). XGBoost.

Module 5: Course 6 end-of-course project

Resources

Activity: Create your Course 6 Automatidata project

- [TLC's accessible vehicle initiatives](#)

Citations

Course 6 end-of-course project overview

- [About TLC](#). (n.d.). NYC Taxi & Limousine Commission.
- [FHV trip record data](#). (n.d.). NYC Taxi & Limousine Commission.

Activity: Create your Course 6 New York City TLC project

- [2017 Yellow taxi trip data - catalog](#). (2022, May 12). [Data.Gov](#).
-