

Cheatsheet

Here is a concise cheat sheet for machine learning (ML) in Python:

****Key Concepts****

- * **Supervised Learning**: Training a model on labeled data to make predictions on new data.
- * **Unsupervised Learning**: Training a model on unlabeled data to discover patterns or structure.
- * **Regression**: Predicting continuous values (e.g., prices, temperatures).
- * **Classification**: Predicting categorical values (e.g., spam/not spam, dog/cat).
- * **Overfitting**: When a model is too complex and performs well on training data but poorly on new data.
- * **Underfitting**: When a model is too simple and performs poorly on both training and new data.

****Libraries****

- * **Scikit-learn**: A popular library for machine learning in Python.
- * **TensorFlow**: A popular library for deep learning in Python.
- * **Keras**: A high-level library for deep learning in Python, running on top of TensorFlow or Theano.

****Code Snippets****

****Scikit-learn****

- * **Importing**: ``import sklearn``
- * **Loading datasets**: ``from sklearn.datasets import load_iris; iris = load_iris()``
- * **Splitting data**: ``from sklearn.model_selection import train_test_split; X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)``
- * **Training a model**: ``from sklearn.linear_model import LinearRegression; model = LinearRegression(); model.fit(X_train, y_train)``
- * **Making predictions**: ``y_pred = model.predict(X_test)``
- * **Evaluating a model**: ``from sklearn.metrics import mean_squared_error; mse = mean_squared_error(y_test, y_pred)``

****TensorFlow****

- * **Importing**: ``import tensorflow as tf``
- * **Creating a model**: ``model = tf.keras.models.Sequential([...])``
- * **Compiling a model**: ``model.compile(optimizer='adam', loss='mean_squared_error', metrics=['accuracy'])``
- * **Training a model**: ``model.fit(X_train, y_train, epochs=10, batch_size=32)``
- * **Making predictions**: ``y_pred = model.predict(X_test)``

****Keras****

- * **Importing**: ``import keras``
- * **Creating a model**: ``model = keras.Sequential([...])``

* **Compiling a model**: `model.compile(optimizer='adam', loss='mean_squared_error', metrics=['accuracy'])`

* **Training a model**: `model.fit(X_train, y_train, epochs=10, batch_size=32)`

* **Making predictions**: `y_pred = model.predict(X_test)`

Additional Tips

* **Data Preprocessing**: Always preprocess your data (e.g., scaling, encoding categorical variables) before training a model.

* **Feature Engineering**: Select and engineer relevant features to improve model performance.

* **Hyperparameter Tuning**: Tune hyperparameters (e.g., learning rate, batch size) to improve model performance.

* **Model Selection**: Choose the best model for your problem based on performance metrics (e.g., accuracy, F1 score).

This cheat sheet covers the basics of machine learning in Python, including key concepts, popular libraries, and code snippets to get you started.