

End-to-End Machine Learning Project

1. Problem formulation

Formulate the problem by considering what ~~is~~ the expected output and the features available.

2. Get the data

* import os

- provides functions for interacting with operating system

* import urllib

- provides functions for fetching URLs

* import pandas as pd

- provides functions to work with data sets

* housing.head()

- returns the top 5 rows of the file

* housing.info()

- gives the count of non-null values in each attribute

* housing.describe()

- gives the statistical measures such as count, mean, std, min, 25%, 50%, 75%, max for each attribute/features.

* import matplotlib.pyplot as plt

- a collection of command style functions which works like MATLAB.

* housing.hist(bins=50, figsize=(20,15))

- histogram is plotted which is used for univariate analysis and bins are used to specify the width of each group.

* split_train_test()

- it divides the dataset into train set and test set. test-ratio can also be specified to divide the dataset.

* train_set.shape

- gives the shape of the train set (i.e no. of rows and columns)

* housing['rd'].value_counts()

- gives the count of values

3. Discover and visualize the data to gain insights
- * `strain_test.set, shape()`
 - * We plot scatterplot by using
`housing.plot(kind='scatter', x='long', y='lat')`
`plt.show()`
 - * Looking for correlation
`corr_matrix = housing.corr()`
 - * Using `scattermatrix` to plot correlations
`scatter_matrix(house = housing[attributes], figsize=(12,7))`
4. Prepare the data for machine learning algorithms.

→ Data cleaning

- * Get rid of the whole attribute
`housing.drop('total-bedrooms', axis=1)`
- * Set missing values to some value
zero, mean, median

→ Handling text & categorical attribute

- * Use 'OrdinalEncoder' to encode categorical features into ordinal integers.

→ Custom Transformers

- * We use `fit, transform, fit_transform, methods`

→ Feature Scaling

- * Min-Max Scaling: for each value, we subtract by min & divide by max-min
- * Standardization: for each value, we subtract mean & divide by std

5. Select and train a model

We use linear regression model

Decision tree Regressor - It is a machine learning algo. used for regression tasks, where the goal is to predict a continuous target variable.

6. Fine-Tune your model

→ `GridSearchCV` is a model provided by 'skikit-learn' library in Python for hyperparameter tuning of machine learning models.

→ `RandomizedSearchCV` can be used instead of above one.

→ Evaluate your system on the test set by using mean-squared error method.

7. Launch, Monitor & Maintain your System

We can automate this process by

- collecting fresh data regularly & labeling it
- writing script to train model & fine tune the hyper-parameter
- writing script to evaluate the model

1. Before training the model, we need to preprocess the data. This involves cleaning the data, handling missing values, and encoding categorical variables.

2. After preprocessing, we can split the data into training and testing sets. This is done using a random splitter. The training set is used to train the model, and the testing set is used to evaluate the model's performance.

3. Once the model is trained, we can use it to make predictions on new data. This is done by passing the new data to the model's predict method.

4. Finally, we can monitor the model's performance over time. This involves collecting new data, training the model, and evaluating its performance. If the performance drops, we can retrain the model with the new data.

5. In conclusion, launching, monitoring, and maintaining a machine learning system is a continuous process. It involves regularly updating the model with new data and retraining it to ensure it remains accurate and effective.

6. The final step in the process is to deploy the model into a production environment. This involves setting up the infrastructure needed to serve the model and ensuring it is accessible to the users. Once deployed, the model can be used to make predictions on new data, and its performance can be monitored and maintained over time.