29/03/2024

Week-2 & Week-3

End-to-End Machine Learning Project

1. Problem formulation

Formulate the problem by considering what ~~are~~ 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 ploted 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['id'].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(frame = housing[attributes], figsize=(12,8))

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 treeRegressor - 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 scikit-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 models & fine tune the hyper-parameter
- writing script to evaluate the model