Date:- / /
Page:-

Project-1

Python & ML

House prediction project,
- Supervised, regression task, batch learning.
- Performance measure - RMSE

- Jupyter Notebook.
- installing libraries.
- python3 notebook
- Dragon Real state - Price Prediction notebook.
- download data set.
- jupyter nb
    - import pandas
    - housing data
    - looking data
    - data knowledge
        (describing, info, value_counts).

info - about data
value_count - count each value no.
describe - count, mean, std, min, 25%, 50%, 75%, max.

- %. matplotlib inline
    → go get graph within there.

- in python "plt.show" for histograph.

- Train-test-splitting.

Now we will divide the data test into 2 parts -
- training test (for studying).
- testing data (for testing).

- import numpy as np.
  # defining function.
- default random seed = 42
  - using shuffled to divide the data randomly.
- we will set ratio to divide data.
  # taking 0.2 a ratio for dividing.

- While diving data randomly, we will get diff. data each time we run the program.
  This will make it possible for machine to go through whole data.
  So, we have to fix data, so that we get fix set each time we run data.

- Now, while diving the data, we should make sure that both train set and test set get each type of record.
  If it's not taken care of, then machine will not get through each type of data resulting in incomplete training.

Date:- / /
Page:-

- We can check how data is divided b/w both sets,
  ∴ strat_train_set['—'].value_counts()
  — test —

- Looking for Correlation,

- Now, we have to know how any parameter is dependant on other parameters.
  ∴ corr_matrix = housing.corr()
  corr_matrix['—'].sort_values(ascending=False).

- Now, we will define it using graphs.
  ⇒ defining attributes b/w which we need graphs.

  ∴ 'pandas.plotting' defining graph.
  scatter_matrix(housing[attributes], figsize=(12,8))
  ↓ method. ↓ defining attributes ↓ size of graph

- Trying out Attribute Combination,

  simply define a new parameter
  ∴ housing['—']
  now simply,
  housing['—'] = housing['_'](/) housing['—']
  ↓ define operator.

  now, get new data set.
  ∴ housing.head()

- we can plot graph blw any two parameter defining type, and x,y also alpha.

- Missing Attributes,

To take care of missing attributes, we have 3 options.
- get rid of missing points.
- get rid of whole attribute.
- set the value to some value (0, mean or median).

⇒ option 1,
'dropna' for removing att. with null values.
null value.

'a.shape' ⇒ to get no. of data count. (row x columns)

- " a = housing.dropna(subset=['RM'])

⇒ option 2,
'drop' to remove whole attribute.

- " housing.drop("RM", axis=1)

⇒ option 3,
get median for the attribute with NA.

- ⇒ " median = housing["RM"].median()

Now fill NA with median.

- " housing["RM"].fillna(median)

Date:- / /
Page:-

- Imputer is used to store some value in the data project.
So, we may need median value in future.
we fit it in housing for future need.

sklearn now use SimpleImputer.

- Now we can check what has Imputer done.
⇒ imputer.statistics_

- Scikit-Learn Design,

Primarily, 3 types of objects.
• Estimators - it estimates some parameter based on a dataset.
Eg, imputer.
has a
it, fit method and transform method.
fit method - fits the dataset and calculates internal parameters
• Transformers - transform method takes input and return output based on the learnings for fit(). It also has a convenience fun. called fit_transform().
• Predictors - linear Regression model is an eg. of predictor. fit() and predict() are two common fun. It also gives score() fun. which will evaluate the predictions.

- Feature Scaling,
Primarily, two types of feature scaling method:
1.) Min-max scaling (normalization), (value-min)/(max-min)
Sklearn provides a class called MinMaxScaler for this.
2.) Standardization, (value-mean)/std.
sklearn provides a class called Standard Scaler for this.

Date:- / /
Page:-

- Creating a Pipeline,

it is to make it easy to make changes in future like models.

- Selecting a desired model for Indian Real State,