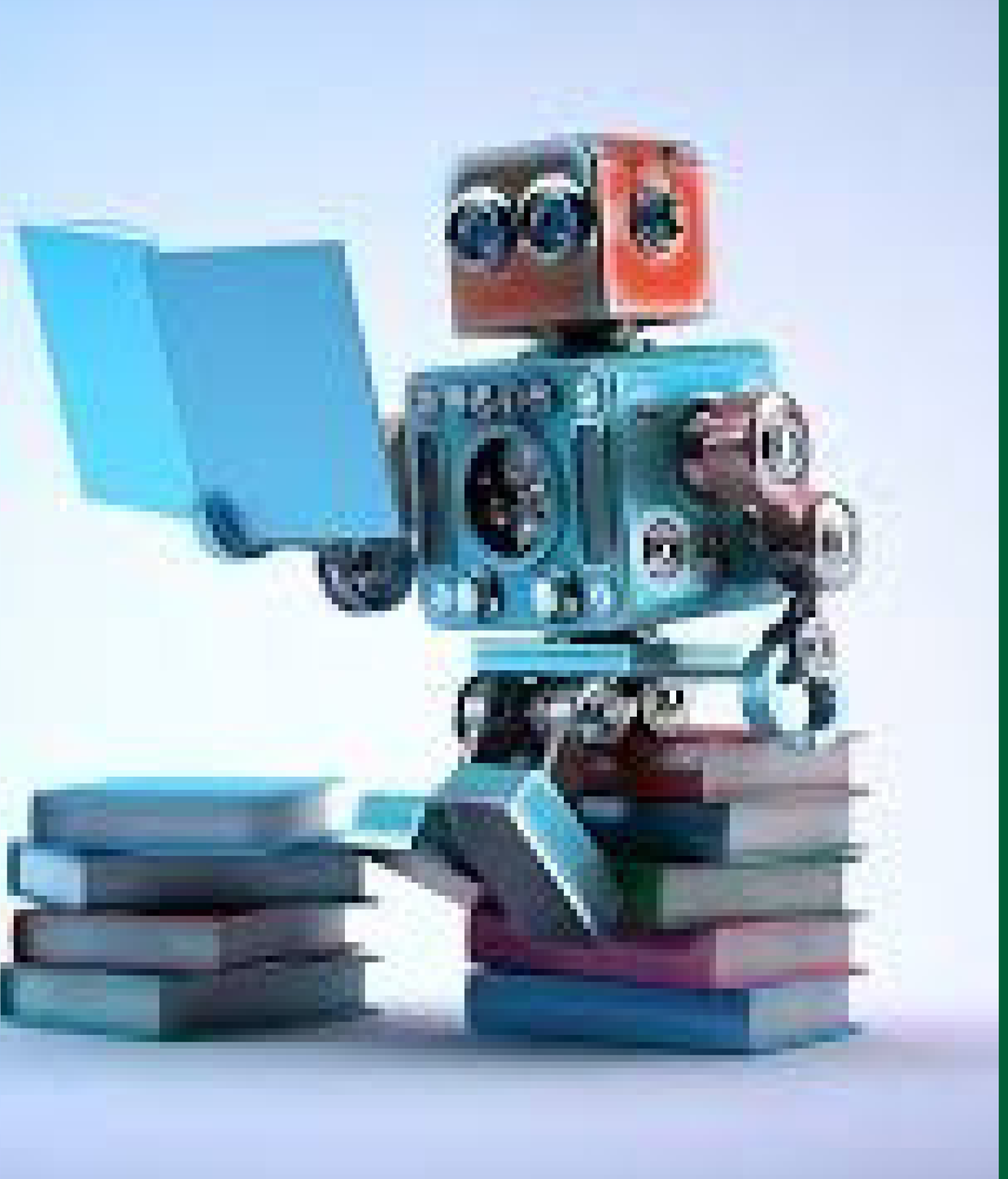
GATE WAY TO ML

MACHINE LEARNING PROJECT: PREDICTING BOSTON HOUSE PRICES WITH REGRESSION



OUTLINE OF TOPICS

WHAT WE'LL DISCUSS

Raw Data
Data Exploration
Descriptive Statistics
Feature Observation
Algorithm modelling
Accuracy measure

THE BOSTON HOUSE PRICE DATA

ABOUT DATA

The dataset of this model comes from the UCI Machine Learning Repository. This data was collected in 1978 and each of the 506 entries represents aggregate data about 14 features for homes from various suburbs in Boston.

ATTRIBUTES IN DATA

THE FEATURES CAN BE SUMMARIZE AS FOLLOWS:

CRIM: This is the per capita crime rate by town

ZN: This is the proportion of residential land zoned for lots larger than 25,000 sq.ft.

INDUS: This is the proportion of non-retail business acres per town.

CHAS: This is the Charles River dummy variable (this is equal to 1 if tract bounds river; 0 otherwise)

NOX: This is the nitric oxides concentration (parts per 10 million)

RM: This is the average number of rooms per dwelling

AGE: This is the proportion of owner-occupied units built prior to 1940

ATTRIBUTES IN DATA

THE FEATURES CAN BE SUMMARIZE AS FOLLOWS:

DIS: This is the weighted distances to five Boston employment centers

RAD: This is the index of accessibility to radial highways

TAX: This is the full-value property-tax rate per $10,000

PTRATIO: This is the pupil-teacher ratio by town

B: This is calculated as $1000(Bk—0.63)^2$, where Bk is the proportion of people of African American descent by town

LSTAT: This is the percentage lower status of the population

MEDV: This is the median value of owner-occupied homes in $1000s

LOADING THE DATA SET

WE WILL LOAD THE HOUSING DATA FROM THE SCIKIT-LEARN LIBRARY AND UNDERSTAND IT.

```
FROM SKLEARN.DATASETS IMPORT LOAD_BOSTON
BOSTON_DATASET = LOAD_BOSTON()
```

DATA PRE PROCESSING

AFTER LOADING THE DATA, IT'S A GOOD PRACTICE TO SEE IF THERE ARE ANY MISSING VALUES IN THE DATA. WE COUNT THE NUMBER OF MISSING VALUES FOR EACH FEATURE USING ISNULL()

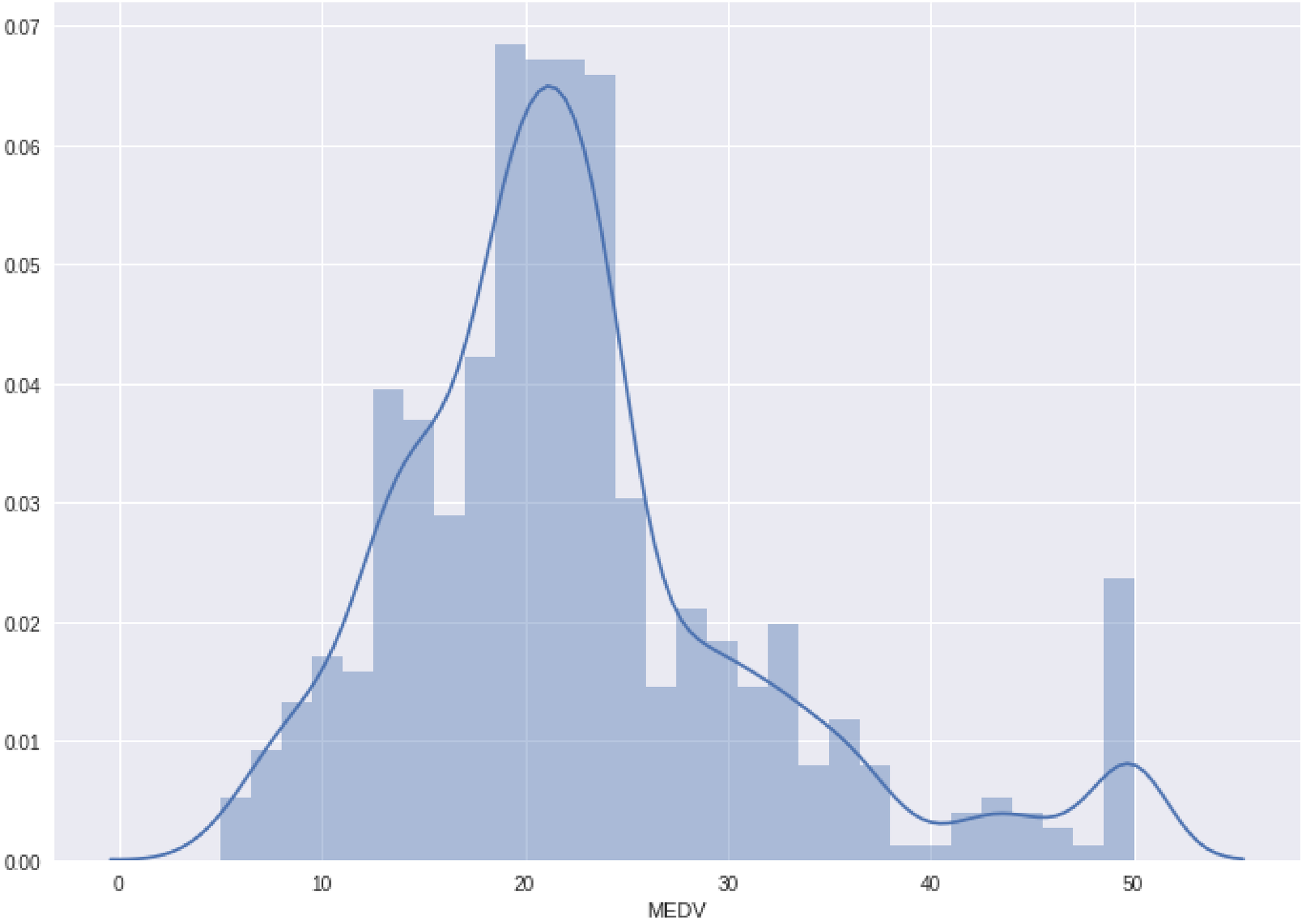
BOSTON.ISNULL().SUM()

EXPLORATORY DATA ANALYSIS

EXPLORATORY DATA ANALYSIS IS A VERY IMPORTANT STEP BEFORE TRAINING THE MODEL. IN THIS SECTION, WE WILL USE SOME VISUALIZATIONS TO UNDERSTAND THE RELATIONSHIP OF THE TARGET VARIABLE WITH OTHER FEATURES.

LET'S FIRST PLOT THE DISTRIBUTION OF THE TARGET VARIABLE MEDV. WE WILL USE THE DISTPLOT FUNCTION FROM THE SEABORN LIBRARY.

```
SNS.SET(RC={'FIGURE.FIGSIZE':(11.7,8.27)})
SNS.DISTPLOT(BOSTON['MEDV'], BINS=30)
PLT.SHOW()
```


0.07
0.06
0.05
0.04
0.03
0.02
0.01
0.00
0
10
20
30
40
50
MEDV

OBSERVATIONS:

TO FIT A LINEAR REGRESSION MODEL, WE SELECT THOSE FEATURES WHICH HAVE A HIGH CORRELATION WITH OUR TARGET VARIABLE MEDV. BY LOOKING AT THE CORRELATION MATRIX WE CAN SEE THAT RM HAS A STRONG POSITIVE CORRELATION WITH MEDV (0.7) WHERE AS LSTAT HAS A HIGH NEGATIVE CORRELATION WITH MEDV(-0.74).

SPLITTING THE DATA INTO TRAINING AND TESTING SETS

NEXT, WE SPLIT THE DATA INTO TRAINING AND TESTING SETS. WE TRAIN THE MODEL WITH 80% OF THE SAMPLES AND TEST WITH THE REMAINING 20%. WE DO THIS TO ASSESS THE MODEL'S PERFORMANCE ON UNSEEN DATA. TO SPLIT THE DATA WE USE TRAIN_TEST_SPLIT FUNCTION PROVIDED BY SCIKIT-LEARN LIBRARY. WE FINALLY PRINT THE SIZES OF OUR TRAINING AND TEST SET TO VERIFY IF THE SPLITTING HAS OCCURRED PROPERLY.

SPLITTING THE DATA INTO TRAINING AND TESTING SETS

```
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=5)
PRINT(X_TRAIN.SHAPE)
PRINT(X_TEST.SHAPE)
PRINT(Y_TRAIN.SHAPE)
PRINT(Y_TEST.SHAPE)
```

TRAINING AND TESTING THE MODEL

WE USE SCIKIT-LEARN'S LINEAR REGRESSION TO TRAIN OUR MODEL ON BOTH THE TRAINING AND TEST SETS.

```
FROM SKLEARN.LINEAR_MODEL IMPORT LINEARREGRESSION
FROM SKLEARN.METRICS IMPORT MEAN_SQUARED_ERROR
LIN_MODEL = LINEARREGRESSION ( )
LIN_MODEL.FIT(X_TRAIN, Y_TRAIN)
```

MODEL EVALUATION

```
# MODEL EVALUATION FOR TRAINING SET
Y_TRAIN_PREDICT = LIN_MODEL.PREDICT(X_TRAIN)
RMSE = (NP.SQRT(MEAN_SQUARED_ERROR(Y_TRAIN, Y_TRAIN_PREDICT)))
R2 = R2_SCORE(Y_TRAIN, Y_TRAIN_PREDICT)
PRINT("THE MODEL PERFORMANCE FOR TRAINING SET")
PRINT("--------------------------------------")
PRINT('RMSE IS {}'.FORMAT(RMSE))
PRINT('R2 SCORE IS {}'.FORMAT(R2))
PRINT("\N")
# MODEL EVALUATION FOR TESTING SET
Y_TEST_PREDICT = LIN_MODEL.PREDICT(X_TEST)
RMSE = (NP.SQRT(MEAN_SQUARED_ERROR(Y_TEST, Y_TEST_PREDICT)))
R2 = R2_SCORE(Y_TEST, Y_TEST_PREDICT)
PRINT("THE MODEL PERFORMANCE FOR TESTING SET")
PRINT("--------------------------------------")
PRINT('RMSE IS {}'.FORMAT(RMSE))
PRINT('R2 SCORE IS {}'.FORMAT(R2))
```

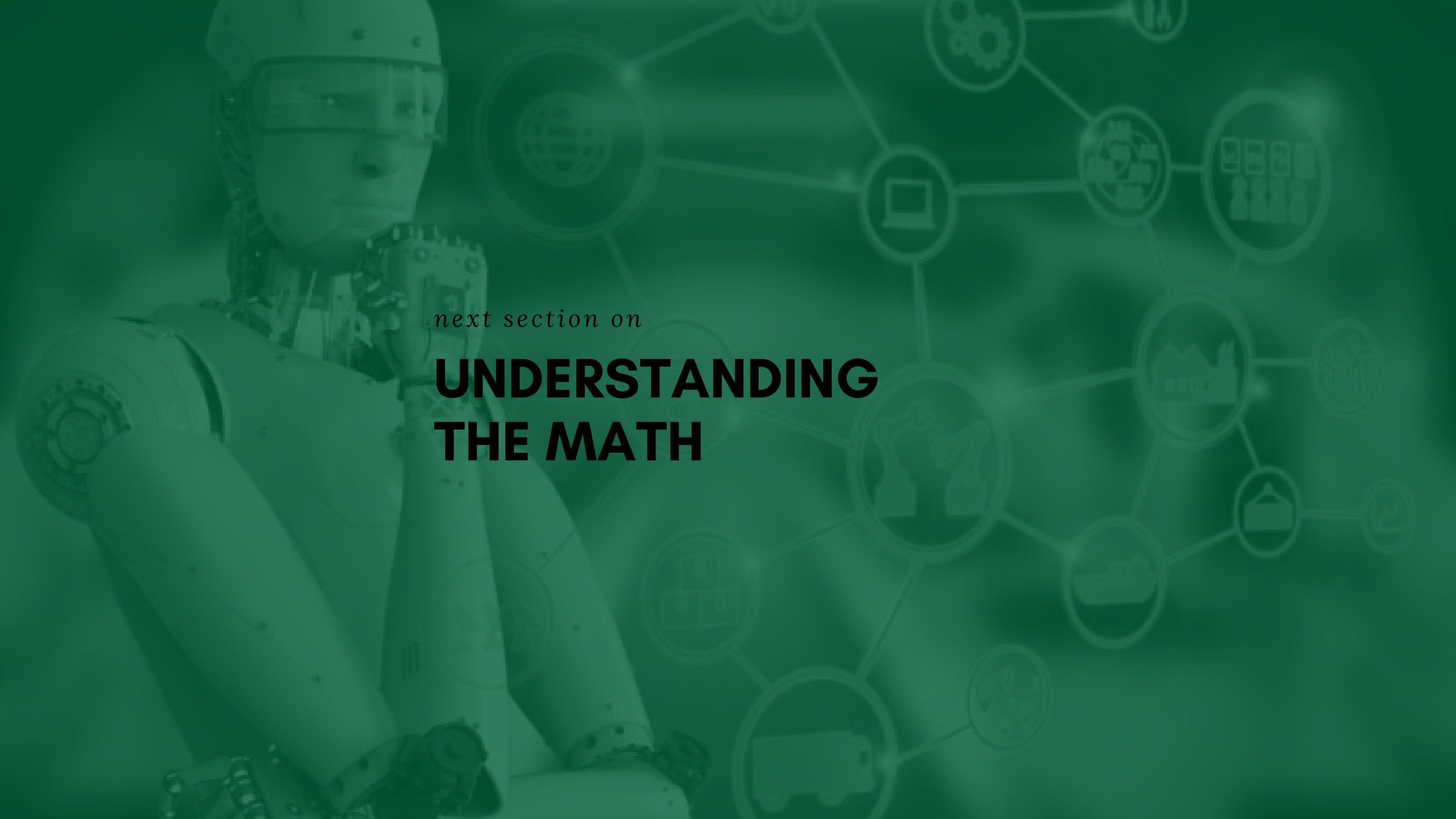
SUM UP

LOADED DATA
PRE PROCESSED DATA
ANALYSIS
FEATURE SELECTION
MODEL SELECTION
ACCURACY MEASURE

LET'S CODE

USE THE CODE SNIPPETS TO
MAKE
WORKING ML MODEL

LET'S CODE



next section on

UNDERSTANDING THE MATH