Data Import

1)First import the dataset into notebook

* use pandas library



* visualize data in a data frame

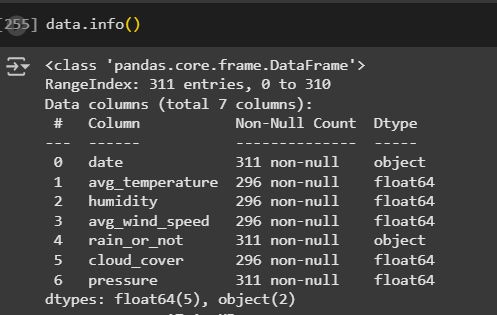
A screenshot of a computer

AI-generated content may be incorrect.

Data Preprocess

2)Next step was preprocess the data before training the models

* get the information of each column



* identify percentage of null values in each column

A screenshot of a computer

AI-generated content may be incorrect.

* print rows which contain null values
  + we can see total 15 number of columns

A screenshot of a computer

AI-generated content may be incorrect.

* Drop the rows because they doesn’t add much information to the data because 4 out of 5 features in each row contain null values.

A screenshot of a computer

AI-generated content may be incorrect.

* Identify duplicated rows
  + There is no duplicated rows

A screen shot of a computer

AI-generated content may be incorrect.

* Identify garbage values
  + First print the data type of each column
  + So no garbage values

A screenshot of a computer

AI-generated content may be incorrect.

* Convert rain\_or\_not categorical field to numerical type

A screen shot of a computer program

AI-generated content may be incorrect.

* Grouping data as features and labels

A screen shot of a computer code

AI-generated content may be incorrect.

* Check the size of data

A black screen with green and white text

AI-generated content may be incorrect.

* Scaling the data using minMax scaler

A screenshot of a computer

AI-generated content may be incorrect.

* Measure correlation with each feature and target
  + As we can see avg\_wind\_speed has lowesr correlation

With target

A screenshot of a computer program

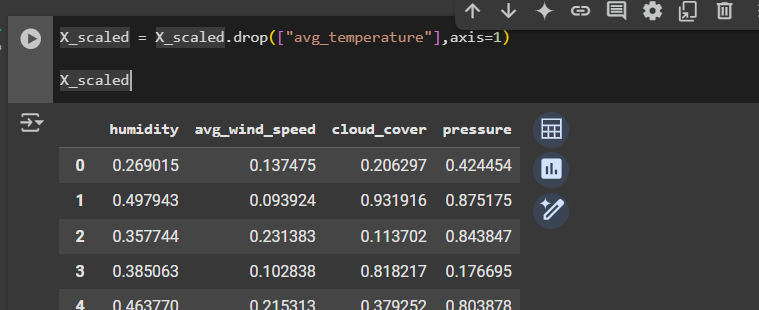
AI-generated content may be incorrect.

* Measure Correlation between each feature
  + As we can see humadity and avg temperature has minimum correlation

A screenshot of a computer screen

AI-generated content may be incorrect.

* Drop the avg\_temperature column
  + Because it has low correlation with target while has strong correlation with humidity



* Drawing Box plot

A screen shot of a computer code

AI-generated content may be incorrect.

A screenshot of a computer screen

AI-generated content may be incorrect.

* Removing outliers

A computer screen shot of a program

AI-generated content may be incorrect.

* Box plots after removing outliers

A screenshot of a graph

AI-generated content may be incorrect.

* Drawing scatter plots

A graph of weather conditions

AI-generated content may be incorrect.

* Drawing histograms

A computer screen shot of a program code

AI-generated content may be incorrect.

A graph of different levels of humidity

AI-generated content may be incorrect.

* Split the data set to training and testing

A screen shot of a computer

AI-generated content may be incorrect.

* Split the data set training and testing



Training

1)Logistic regression model

* First I’ve used GridSearchCV
* To find best parameters
* Best Parameters: {'C': 1,

'penalty': 'l2',

'solver': 'liblinear'}

* From that after model evaluation
* Accuracy: 0.6949
* Log Loss: 0.5875
* Next parameters inserted manually
  + Parameters were
    - random\_state=0,
    - penalty='l2';
    - C=0.1,
    - solver='liblinear',
* From that after model evaluation
  + - * Accuracy: 0.7288
      * Log Loss: 0.5944

2)Decision tree

* From GridSearchCV
  + Best Parameters: {'
    - criterion': 'gini',
    - 'max\_depth': 6,
    - 'min\_samples\_split': 7}
  + After evaluation
    - Accuracy: 0.7119
    - Log Loss: 3.5250
* Next parameters inserted manually
  + Parameters were
* random\_state=0,
* max\_depth=5
* min\_samples\_split=4
* criterion= 'gini'

from that after model evaluation

* Accuracy: 0.6949
* Log Loss: 1.2606

3)Random forest

* From GridSearchCV
  + Best Parameters: {
    - 'max\_depth': 1,
    - 'max\_features': None, 'min\_samples\_split': 2,
    - 'n\_estimators': 200}
  + After evaluation

Accuracy: 0.6271Log Loss: 0.5914

* Next parameters inserted manually
  + Parameters were
* random\_state=0,
* n\_estimators= 200,
* max\_depth=1,
* min\_samples\_split= 6,
* max\_features = 'log2'

from that after model evaluation

* Accuracy: 0.7288
* Log Loss: 0.5834

3)Gradient boosting

* From GridSearchCV
  + Best Parameters: {
    - 'learning\_rate': 0.01,
    - 'max\_depth': 3,
    - 'max\_features': 'sqrt',
    - 'min\_samples\_split': 8,
    - 'n\_estimators': 75After evaluation

Accuracy: 06949 Log Loss: 0.5903

* Next parameters inserted manually
  + Parameters were
* random\_state=42,
* n\_estimators=50,
* learning\_rate=0.01,
* max\_depth=7,
* min\_samples\_split=10,
* max\_features='sqrt'from

that after model evaluation

* Accuracy: 0.7458
* Log Loss: 0.5938

Gradient boosting perform very well so far, we can enhance model by adjusting parameters further