Methodology:

1. overall processing Diagram
2. Dataset
3. Data Processing(Keras,Panda nia )
4. Result of knn(graph)
5. Result of rnn(graph)

Result:

(why use two algorithms)

Conclusion: benefit of using two algorithms and accuracy

Methodology:

The whole prediction of weather will be evaluated by using the K-nearest neighbor (KNN) model. In addition, Recurrent Neural Network (RNN) model will be implemented to individual parameters of weather (humidity, temperature, air pressure, and so on) to forecast the long-term trend of the parameters.

**Dataset:** The dataset consists of historical weather parameters (temperature, pressure, relative humidity) for major North American and other cities around the world over an extended time period of 2012 to 2017. Hourly data points are recorded, giving over 45000 data points, in total.

* By attempting to do a time-series prediction, we are implicitly assuming that the past weather pattern is a good indicator of the future.
* For this analysis, we focus only on the data for the city of Boston.
* The full dataset can be found here: <https://www.kaggle.com/selfishgene/historical-hourly-weather-data>

In our project, seven sets of data are used for predicting weather and all files are in CSV(comma-separated values) format. The first data set is used for cities and country details and there are also defined Latitude and Longitude country wise and city wise. And other datasets are humidity, pressure, and temperature are numbering formats that are recorded country and city-wise. “Weather description” is the main class data set to predict the weather. “Wind direction” and “wind speed” are also used for predicting the class and which are also numbering formats.

**Data Processing:** For data processing here used predefined libraries pandas, Keras, and sklearn and level encoding for preprocessing.

1. First of all here, we import the library

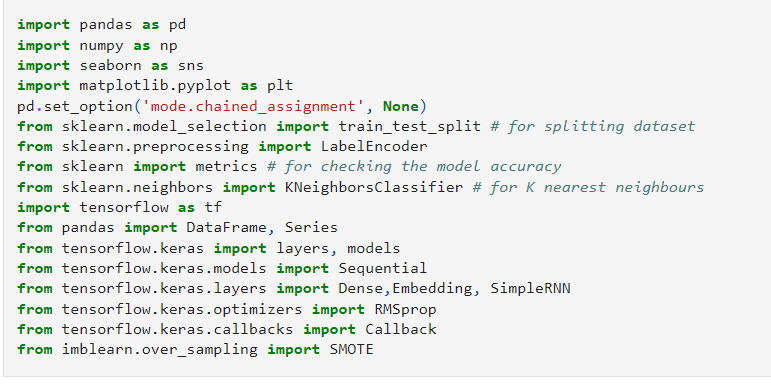


Fig: Import library

1. By using panadas library here read the whole data sets

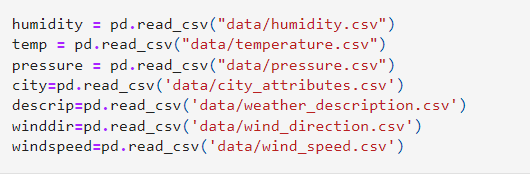


Fig: Read the data set

1. In our project Boston city is considered for predicting weather and merging all datasets as attributes only for Boston City.

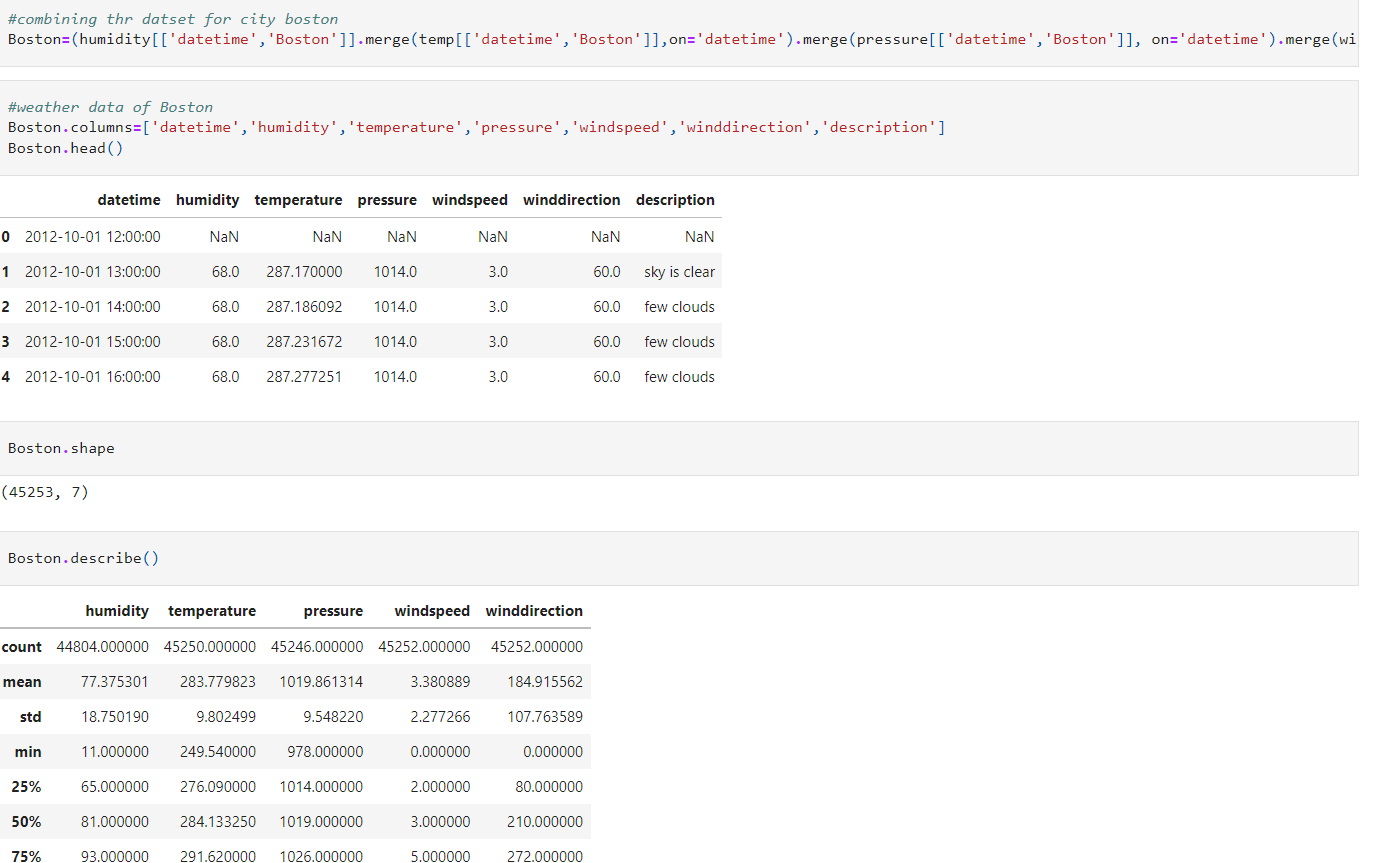


Fig : merging all attributes for Boston city

1. Then we format the date-time by using the date-time function which is imported from pandas only for Boston city.

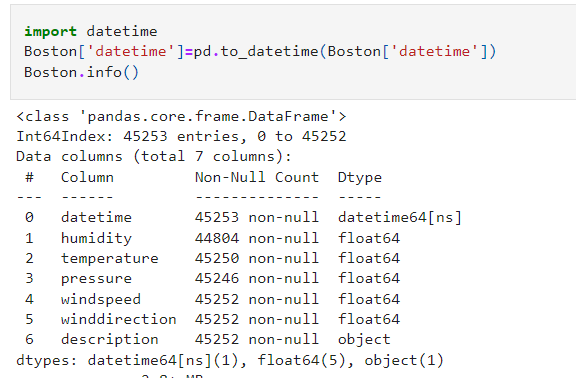


Fig: Date time formatting

1. Then check is there any null value or not and after checking use fillna function and the method is bfill to remove the null value. The fillna method replaces NULL values with a value that you specify. Unless the inplace parameter is set to True, the fillna method returns a new DataFrame object. If the inplace parameter is set to True, the fillna method replaces the original DataFrame instead.

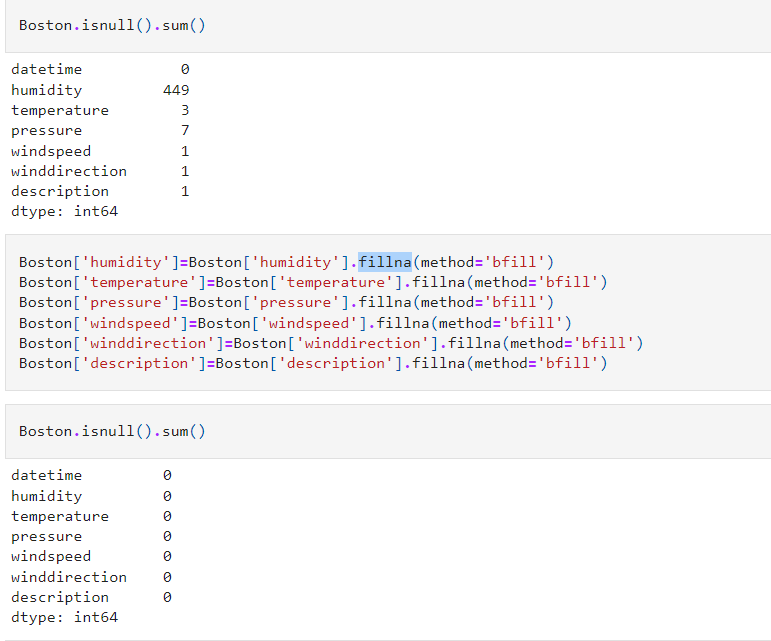


Fig: Null Value removing

1. Then we split the date-time by using the panadas library.

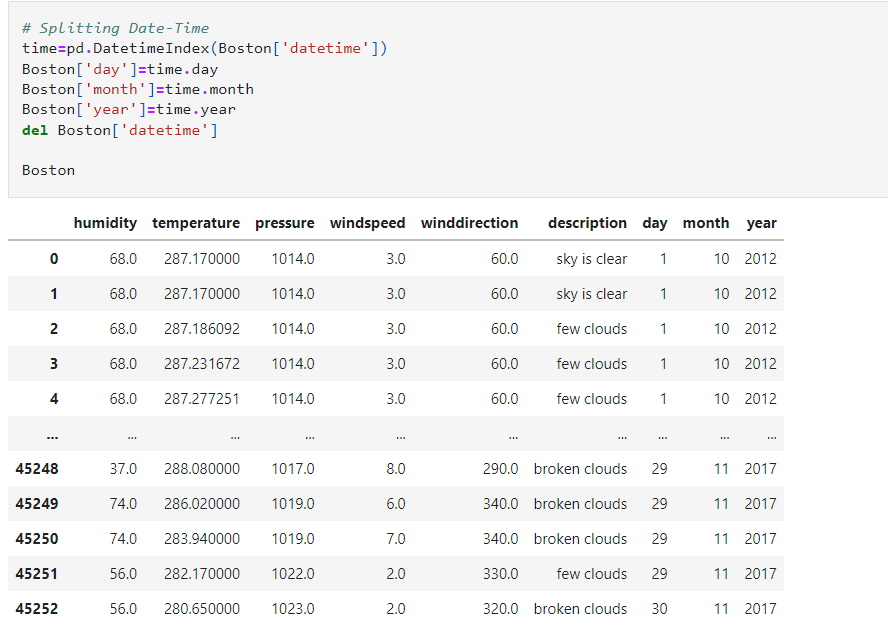
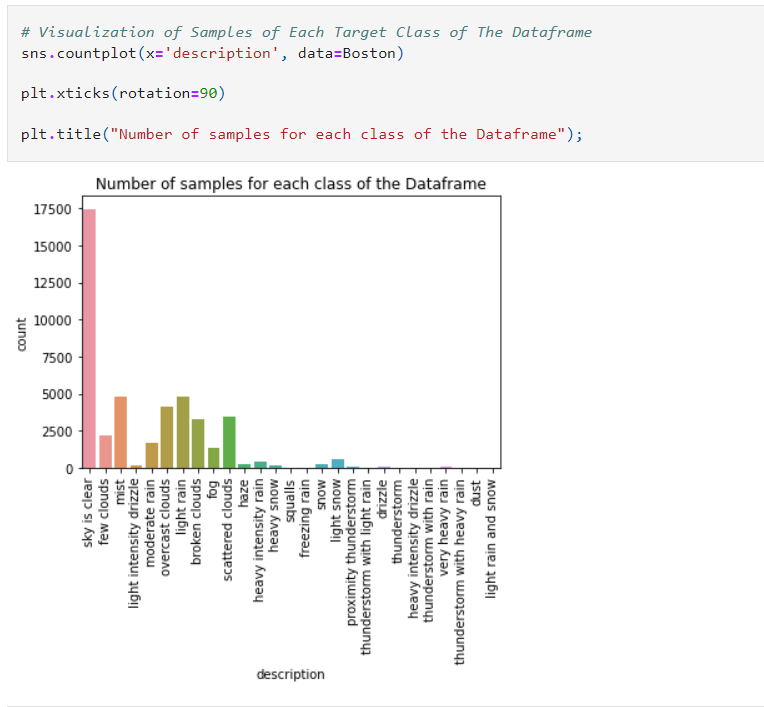


Fig: Splitting format.

1. Then we described class attributes and how many different instances are used to predict the weather numbering and graph format. And plot all instances.



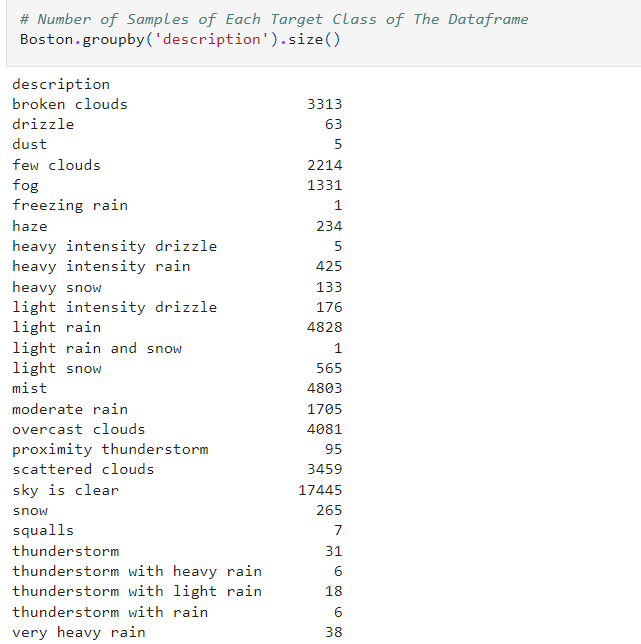
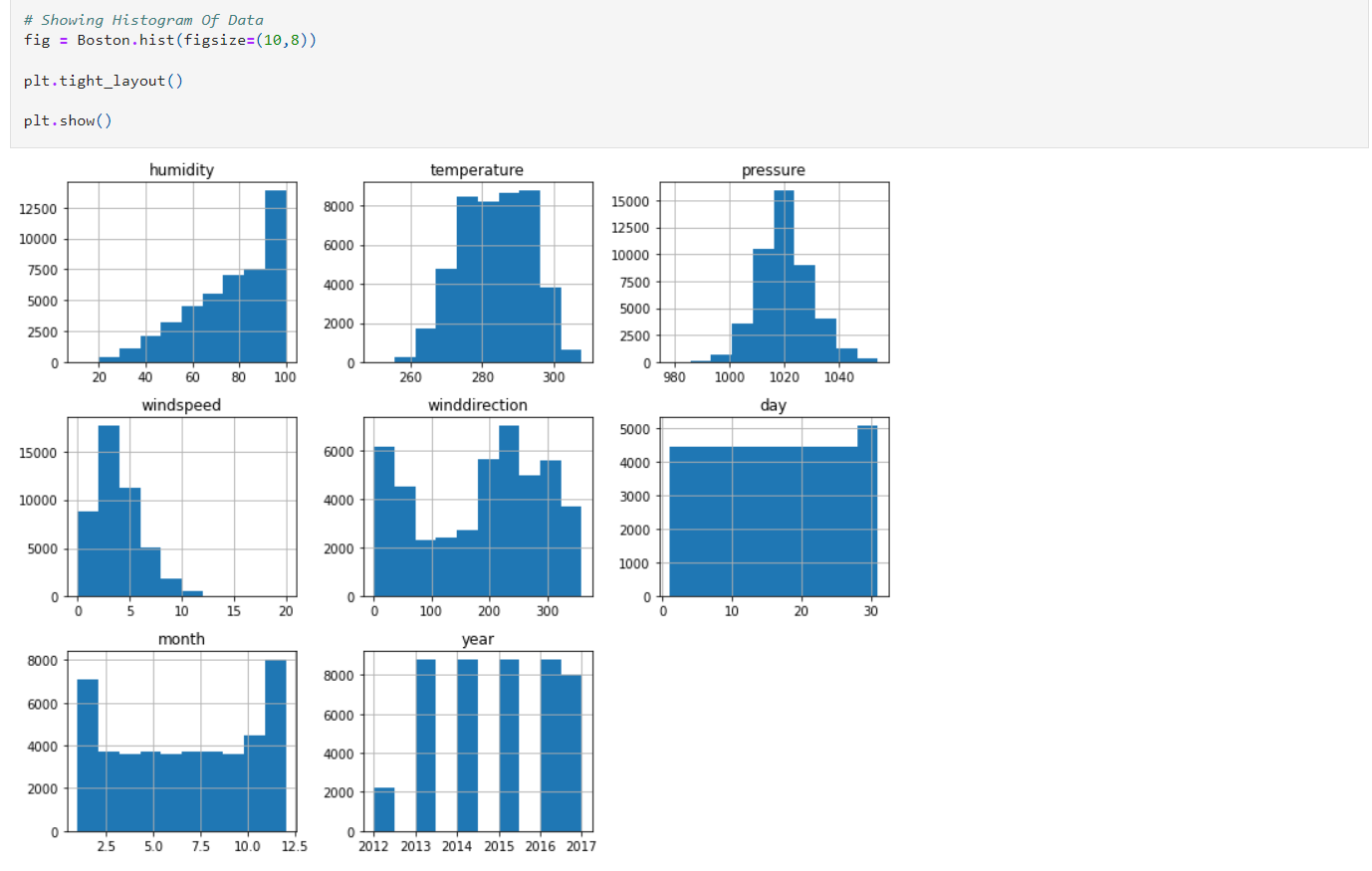


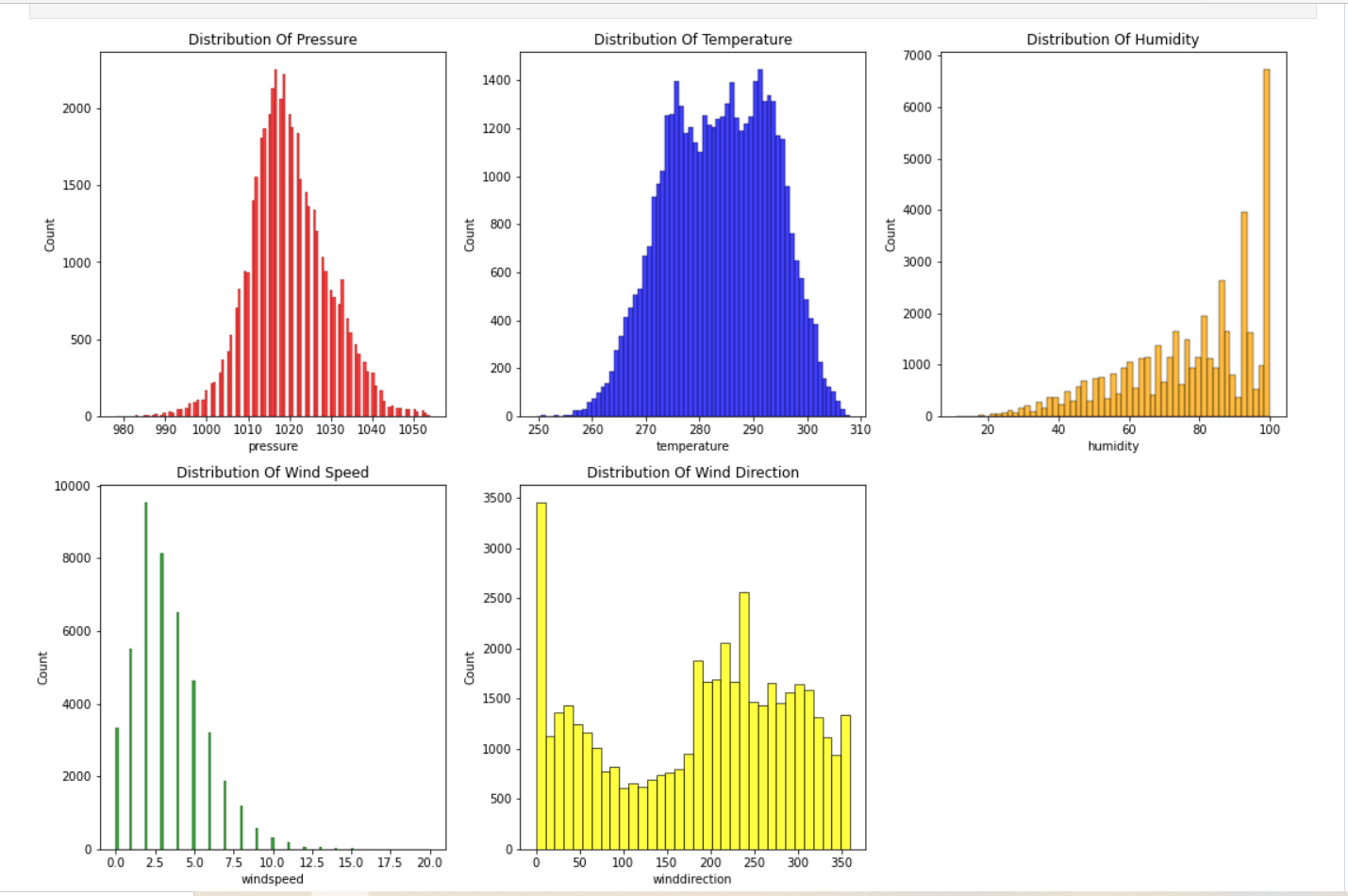
Fig: Number format



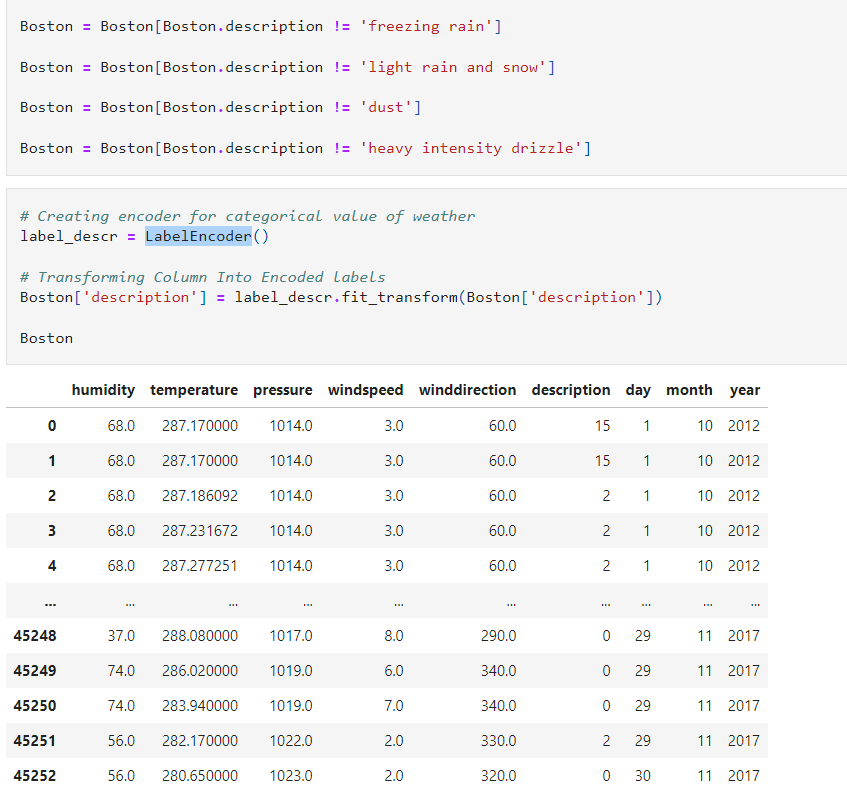
Fig: Graphically Format

1. Then Histogram represents all attributes and Visualization of all attributes without class attributes.

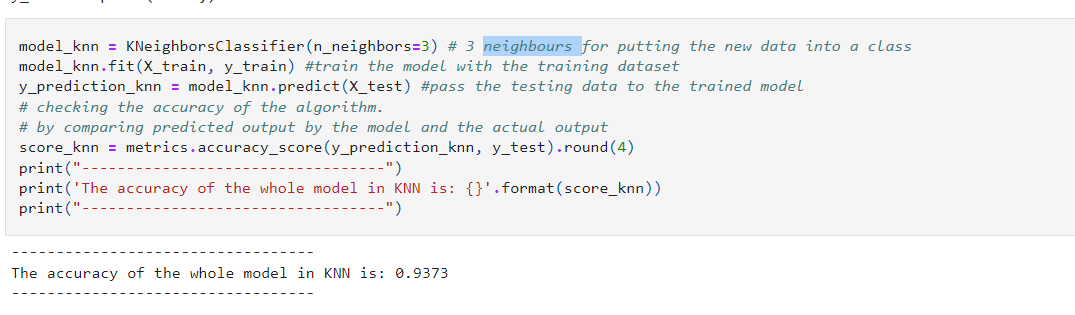




1. After that we get into number format only class attribute which name is a “description” using LabelEncoder.



**Result of KNN:** The 3- nearest neighbors algorithm used for weather forecasting to determine whole weather



Now plot all attributes for Boston city

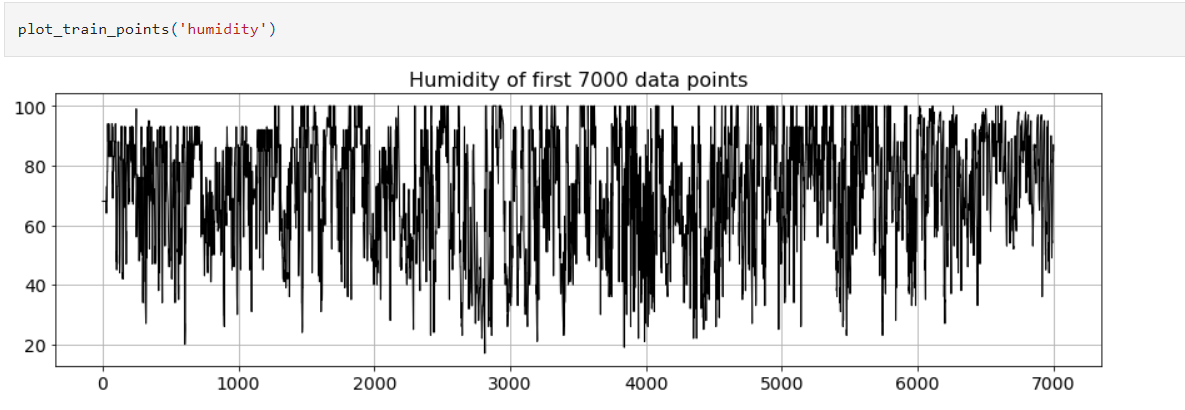


Fig: Humidity

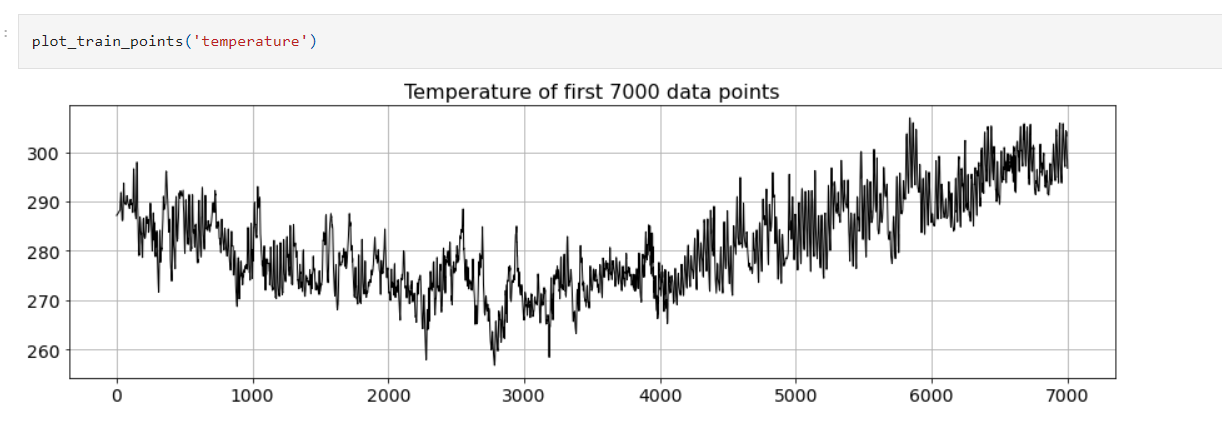
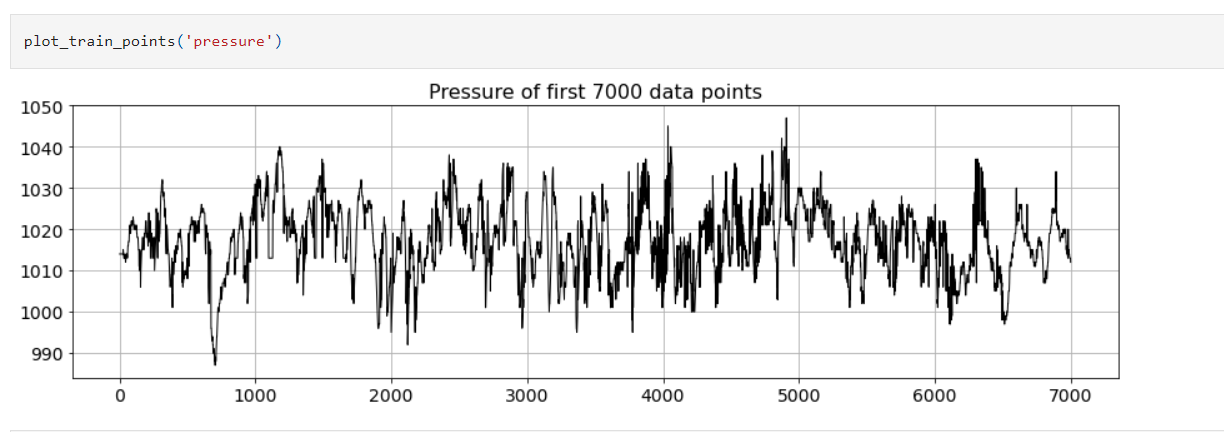


Fig: Temperature



Pressure

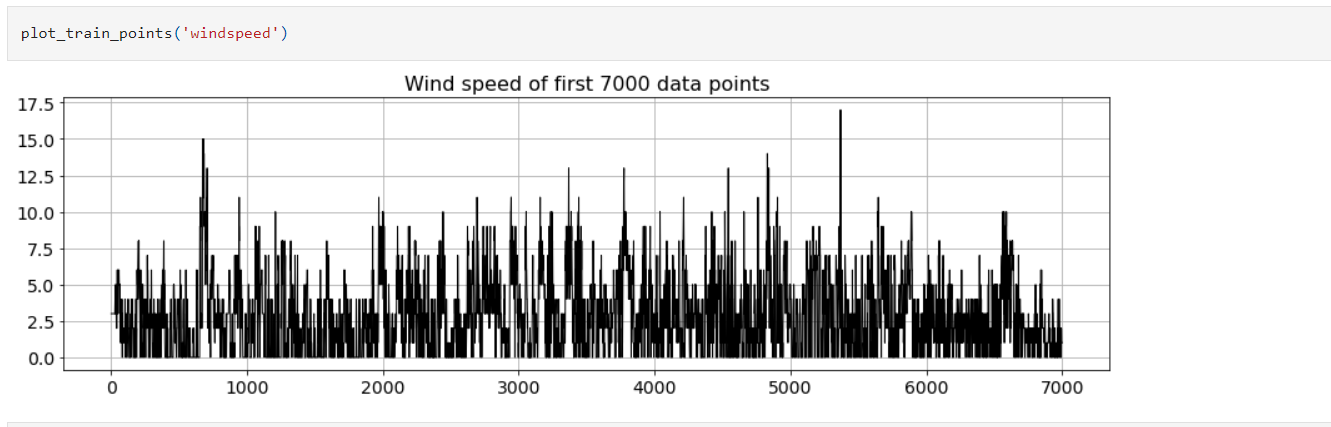


Fig: Wind Speed

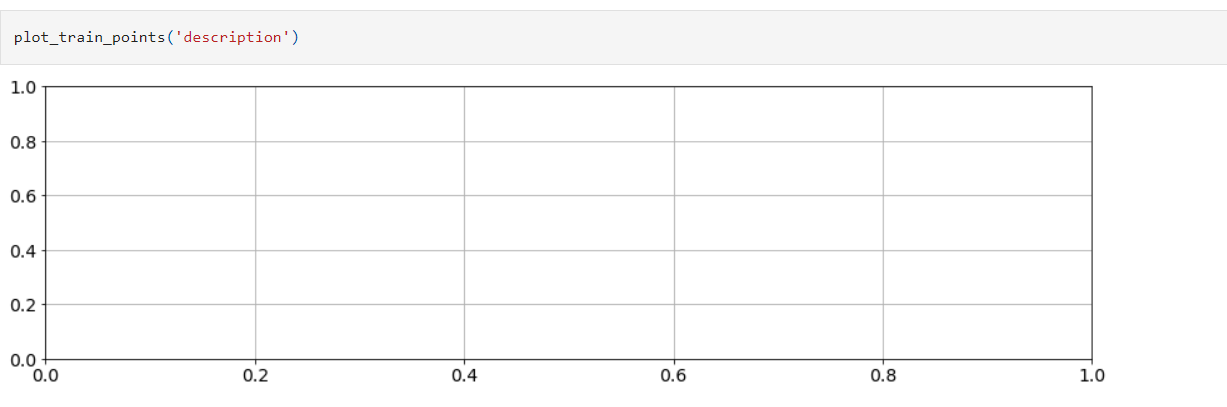


Fig: Description

Here consider 10000 data for the training set and 35241 for the test set. In the plot blue graph represents the train set and the yellow graph for the test set.

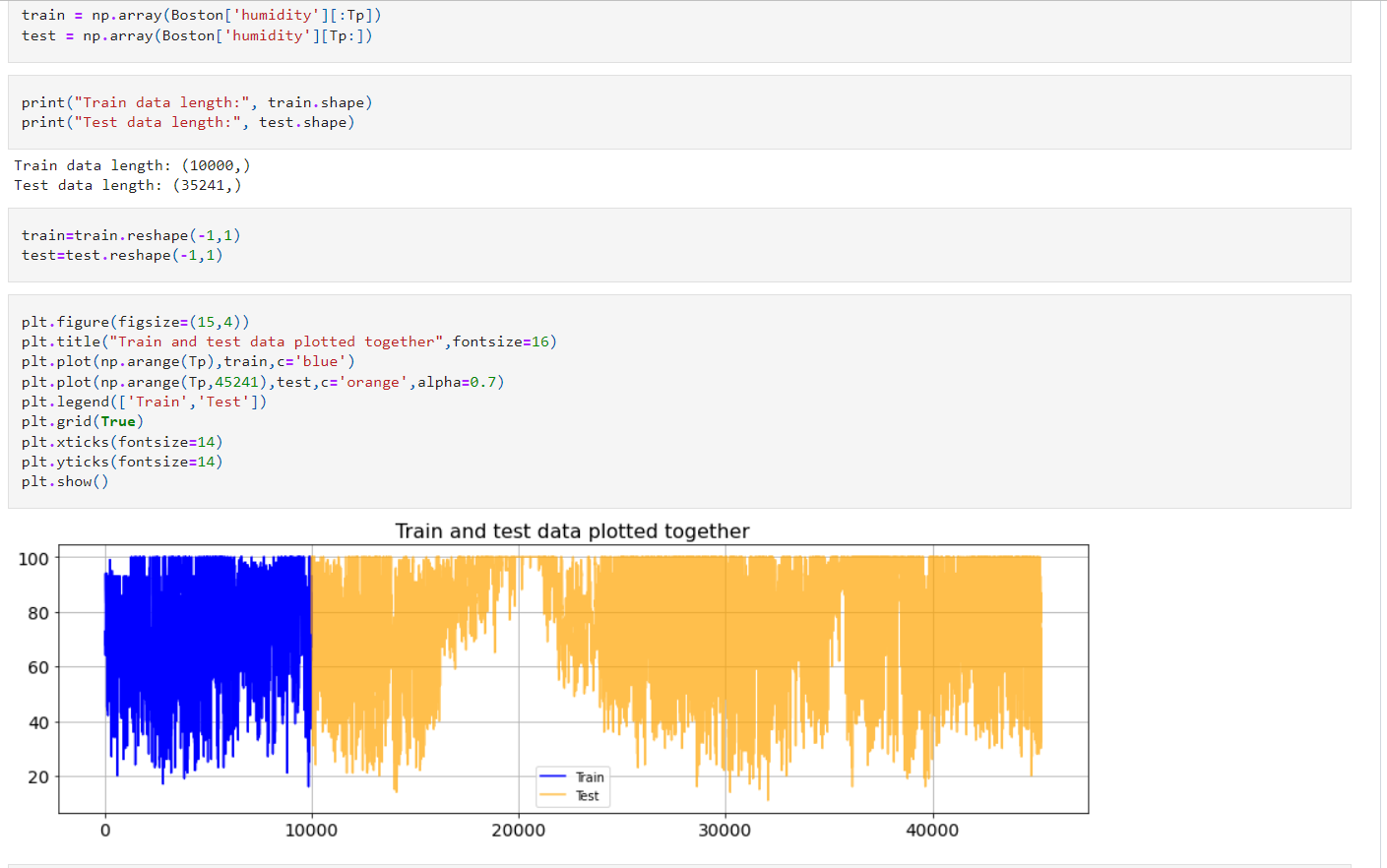


Fig: plot the train and test data

**Result of RNN:**

Train and Test sets are converted into the matrix. Then apply simple RNN for each attribute and compare test sets with the rain set in the plotting method. The root means square error (RMSE) is the residuals' standard deviation (prediction errors). Residuals are a measure of how far the data points are from the regression line; RMSE is a measure of how spread out the residuals are. To put it another way, it shows how closely the data is grouped around the line of greatest fit.

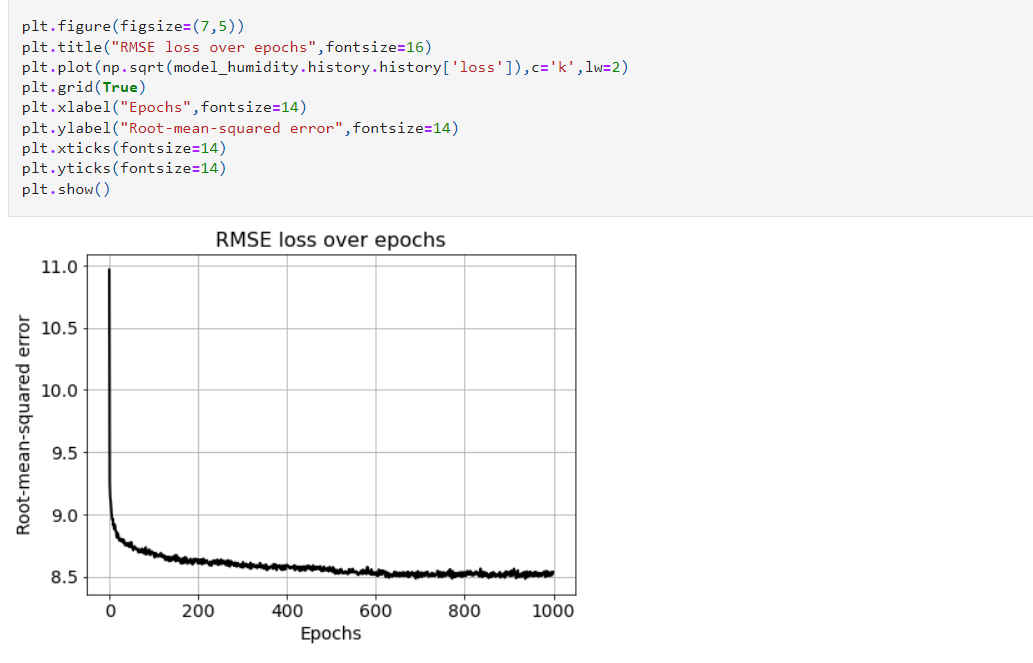
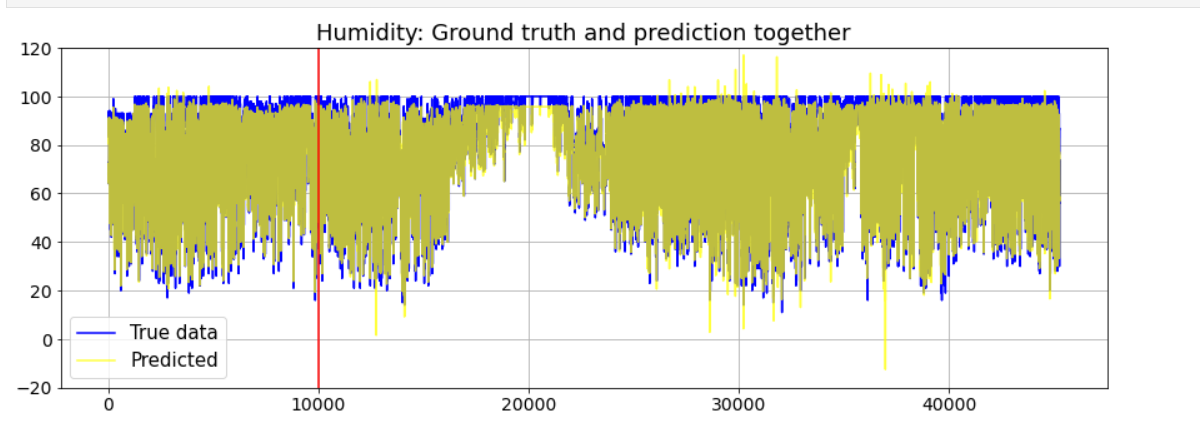
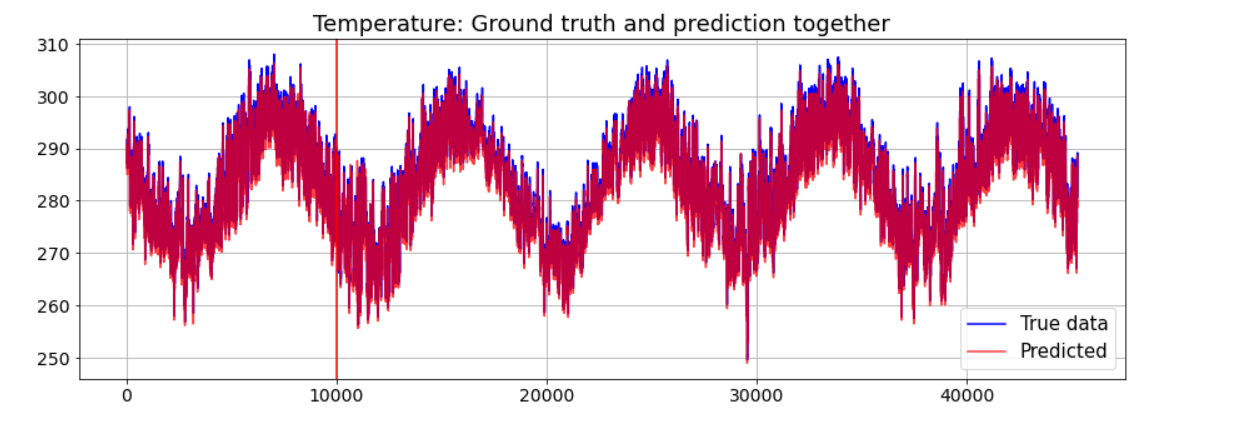
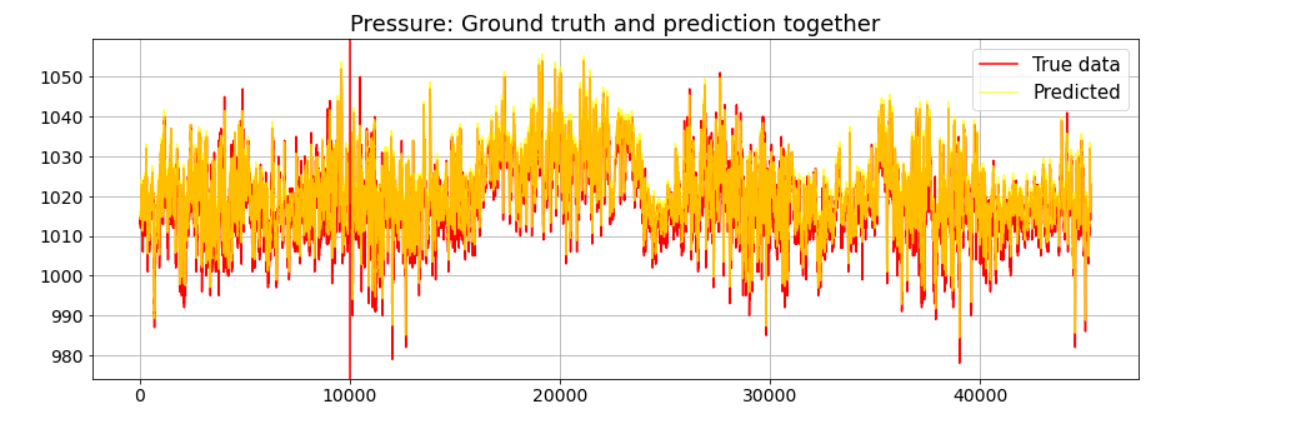


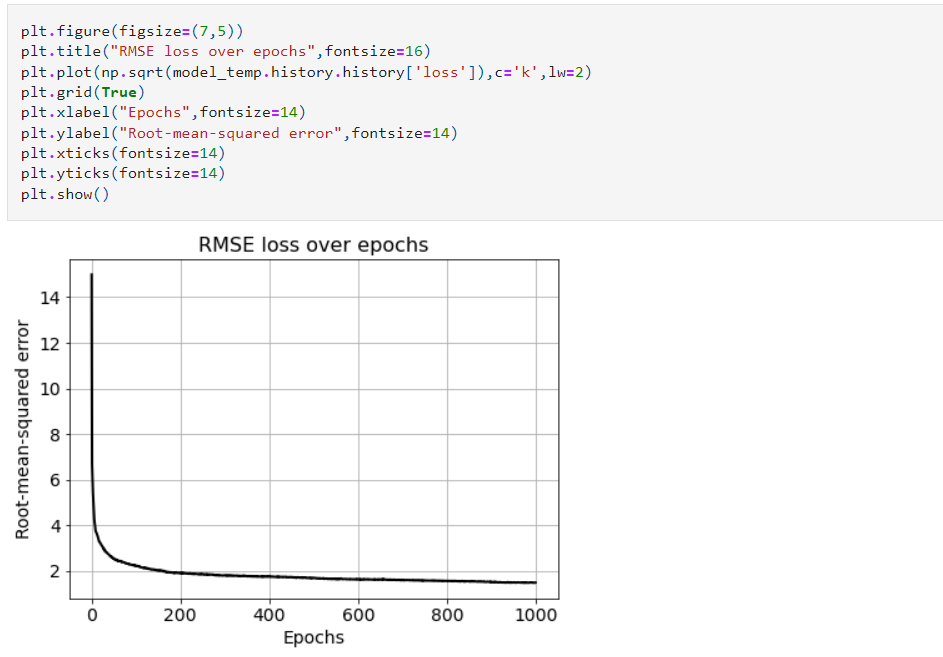
Fig:RMSE







Minimize Root mean square error



Weather forecasting is vital in our everyday life, particularly in agriculture and associated diligence. We created and applied a general computational architecture based on learning models in this study. Using the machine learning ideas of KNN and RNN, we developed a low-cost, reliable, and successful weather forecasting in Python. KNN was applied to determine the delicacy of weather prediction. For the prediction of a single parameter, an RNN is applied. We can also estimate long-term weather forecasting patterns and their numerous parameters, which may be employed in remote places