RADAR Packet Analyzer

By: Visharad Vashist

The project is implemented in 

Python is an interpreted, high-level, general-purpose programming language.

Python is dynamically typed and garbage-collected. It supports multiple programming paradigms, including procedural, object-oriented, and functional programming

Aim is to decode messages in packets received from RADAR and make predictions about trajectories of aircrafts.

Data Packets stored in .pcap files are read using pypcapfile, converted to readable form and stored in a .csv file using pandas

Pypcapfile

pypcapfile is a pure Python library for handling libpcap savefiles. The core functionality is implemented in pcap.savefile. It currently has very basic support for Ethernet frames and IPv4 packet parsing.

Pandas

pandas is a Python package providing fast, flexible, and expressive data structures designed to make working with structured (tabular, multidimensional, potentially heterogeneous) and time series data both easy and intuitive

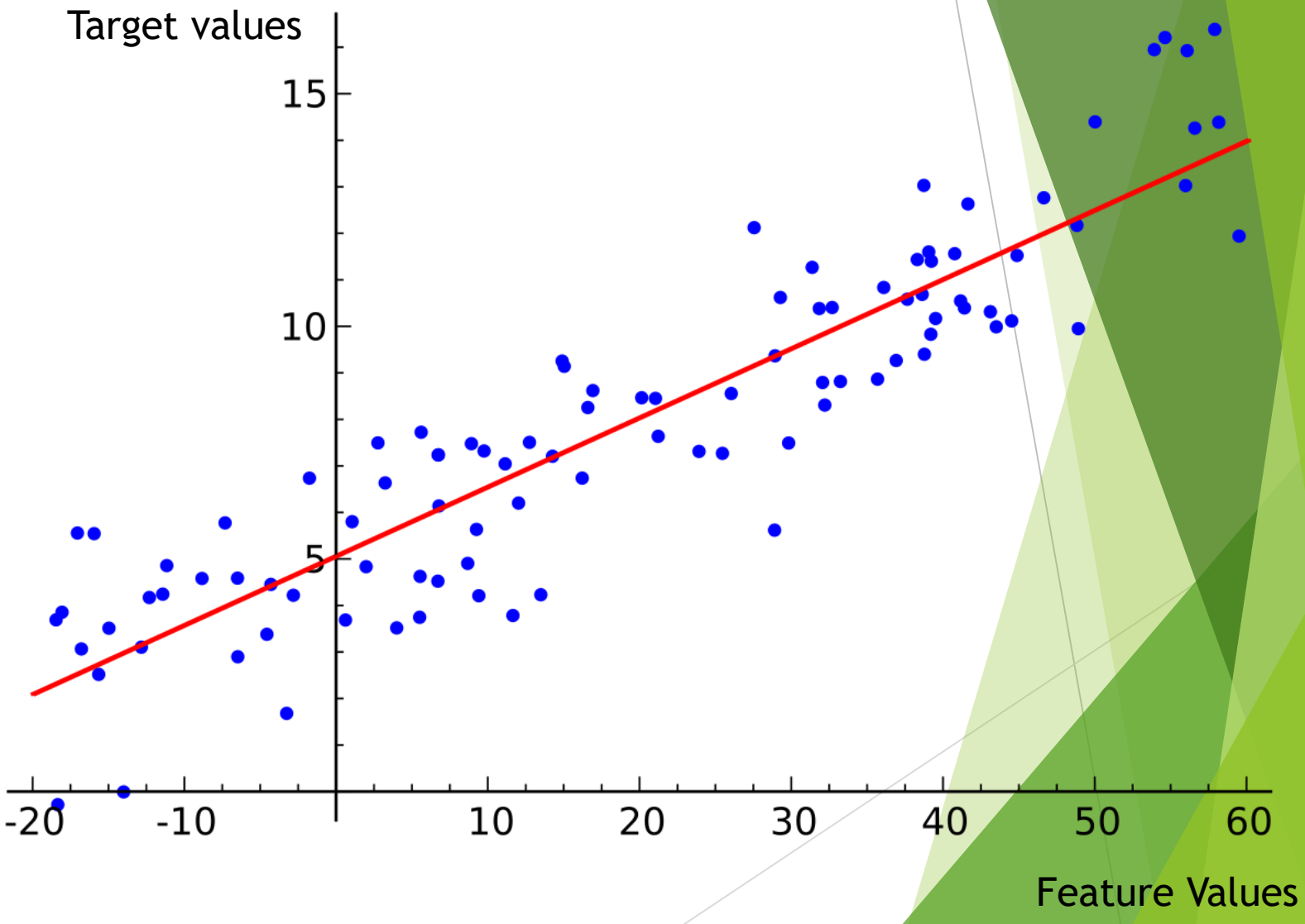
Records of aircrafts are grouped using aircrafts' unique code.

Values of their polar coordinates are used to train a

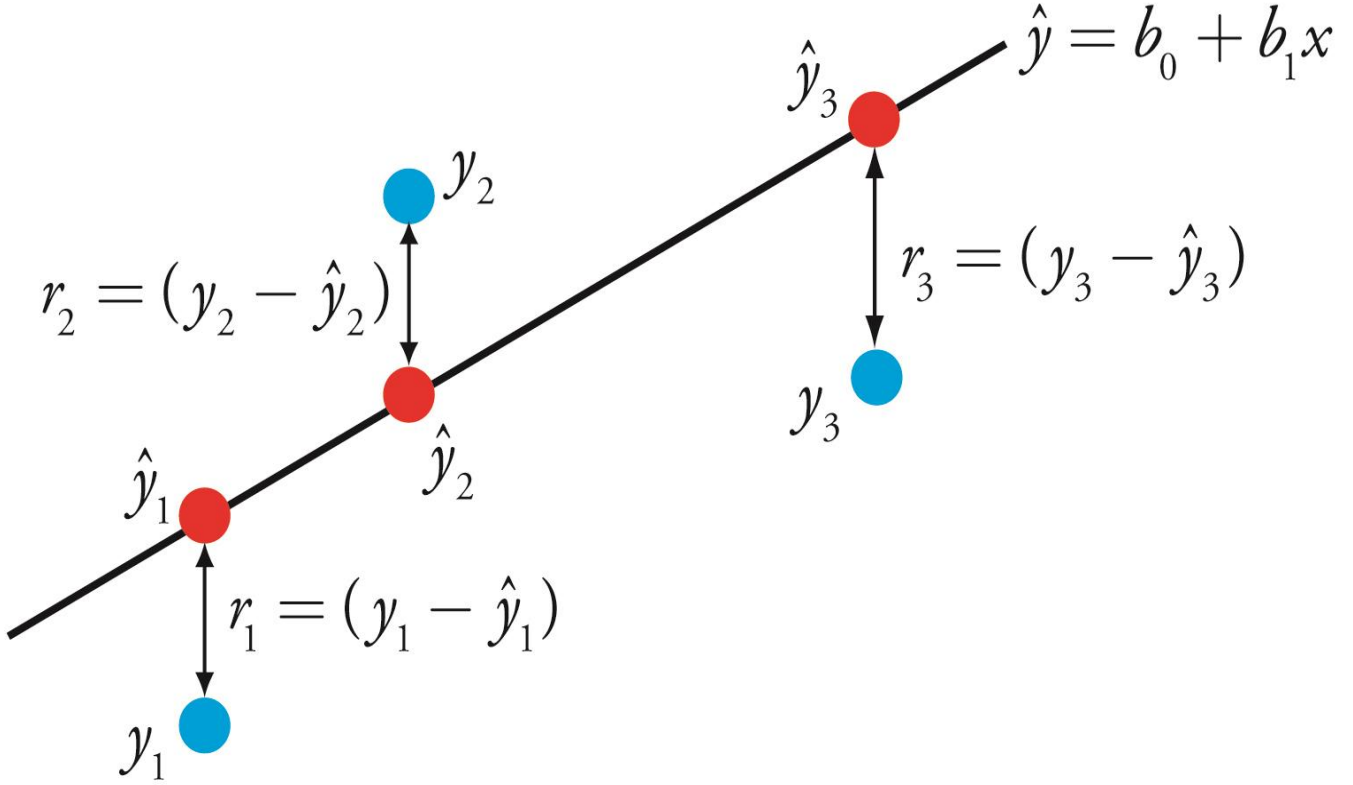
Linear Regression

model and predict further values recursively (using predicted values to predict next)

It simply means, finding a linear relationship between the features and target values and using the same to predict further target values.



It starts with assuming a slope and intercept of the line that would be nearest to all data-points.



Then, finding the distance (error) between the data-points and the line.

Individual Error for one point = $y_i - (mx_i + c)$

Total Square Error = $\sum (y_i - (mx_i + c))^2$

Mean Squared Error= $\frac{1}{N} \sum (y_i - (mx_i + c))^2$

And finally, trying to reduce that error in every iteration.

$$c\_gradient = \frac{\partial MSE}{\partial c} = \frac{\Sigma\ 2}{N}\left(y_i - (mx_i + c) \right)$$

$$m\_gradient = \frac{\partial MSE}{\partial m} = \frac{\Sigma\ 2}{N} m\left(y_i - (mx_i + c) \right)$$

$$new\_c = current\_c - learning\_rate * c\_gradient$$

$$new\_m = current\_m - learning\_rate * m\_gradient$$