Advertisement Sale prediction from an existing customer

Find The Problem-Application — To Predict — Predicting Whether this New customer, will buy this Product or not

Collecting Dataset — Input: AGE & Salary Output: PURCHASE STATUS — Base on Age, Salary of existing customer Status(Purchased & Not Purchased

Load & Summarize Dataset — Load Dataset from the directory & Summarize the details such as no. of rows and columns & content
- Pandas-Load CSV Format Dataset
  dataset=pandas.read_csv("Dataset.csv")
- No.of Rows & Rows & Columns
  dataset.shape
- Display 1st 5 rows of dataset
  dataset.head(5)

Segregating Dataset into X & Y — iloc - It helps us select a value that belongs to a particular row or column
- SYNTAX: dataset.iloc[:,start_col:end_col]
  X = dataset.iloc[:, :-1].values
- Y = dataset.iloc[:, -1].values

Splitting Dataset to Train & Test — Useful for validation — train_test_split(X, Y, test_size = 0.25, random_state = 0)

Feature Scaling
- Since both the features have different scales, there is a chance that higher weightage is given to features with higher magnitude. This will impact the performance of the machine learning algorithm and obviously, we do not want our algorithm to be biassed towards one feature.
- we scale our data to make all the features contribute equally to the result
- Types
  - Standardization — Here the values are centered around the mean with a unit standard deviation. This means that the mean of the attribute becomes zero and the resultant distribution has a unit standard deviation.
  - Normalization — Normalization is a scaling technique in which values are shifted and rescaled so that they end up ranging between 0 and 1. It is also known as Min-Max scaling

Algorithm — Logistic Regression — Uses 1 or More Independent Variable (X) to determine dependent variable (Y)

Training — Training our Model for Pre-processed Dataset — model.fit(X_train, y_train)

Validation — Obtaining the accuracy of the Model — Confusion Matrix

Prediction — Observing how our model is classifying our new data — result = model.predict(sc.transform(newCust))