# CREATING A CHATBOT USING PYTHON

## Phase 3: Development part 1

Creating a CSV file named “chatbot\_data.CSV”

The steps to create a CSV file using the given dataset

1.open a text editor or spreadsheet software: you can use software like Microsoft Excel , Google sheets or notepad.

2.Input your data: Enter the data into the rows and columns of the chosen software.

Each column should represent a different data category, and each row should represent a separate data entry.

3.save the file: In the text editor or spreadsheet software, go to the “file” menu and select “save as” or export and save the file with ”.csv” file extension.

# DATA PREPROCESSING:

Data preprocessing is required tasks for cleaning the data and making it suitable for a machine learning model which also increases the accuracy and efficiency of a machine learning model.

It involves below steps:

Importing libraries

Importing datasets

Finding Missing Data

Encoding Categorical Data

Splitting dataset into training and test set

1. Get the Dataset

To create a machine learning model, the first thing we required is a dataset as a machine learning model completely works on data. The collected data for a particular problem in a proper format is known as the dataset.

Dataset may be of different formats for different purposes, such as, if we want to create a machine learning model for business purpose, then dataset will be different with the dataset required for a liver patient. So each dataset is different from another dataset. To use the dataset in our code, we usually put it into a CSV file. However, sometimes, we may also need to use an HTML or xlsx file.

2) Importing Libraries

Numpy: Numpy Python library is used for including any type of mathematical operation in the code. It is the fundamental package for scientific calculation in Python.

Matplotlib: The second library is matplotlib, which is a Python 2D plotting library, and with this library, we need to import a sub-library pyplot.

Pandas: The last library is the Pandas library, which is one of the most famous Python libraries and used for importing and managing the datasets. It is an open-source data manipulation and analysis library.

3) Importing the Datasets

Now we need to import the datasets which we have collected for our machine learning project. But before importing a dataset, we need to set the current directory as a working directory. To set a working directory in Spyder IDE, we need to follow the below steps:

1)Save your Python file in the directory which contains dataset.

2)Go to File explorer option in Spyder IDE, and select the required directory.

3)Click on F5 button or run option to execute the file.

x= data\_set.iloc[:,:-1].values

output:

[['India' 38.0 68000.0]

['France' 43.0 45000.0]

['Germany' 30.0 54000.0]

['France' 48.0 65000.0]

['Germany' 40.0 nan]

['India' 35.0 58000.0]

['Germany' nan 53000.0]

['France' 49.0 79000.0]

['India' 50.0 88000.0]

['France' 37.0 77000.0]]

To extract dependent variables, again, we will use Pandas .iloc[] method.

y= data\_set.iloc[:,3].values

Output:

array(['No', 'Yes', 'No', 'No', 'Yes', 'Yes', 'No', 'Yes', 'No', 'Yes'],

dtype=object)

4) Handling Missing data:

The next step of data preprocessing is to handle missing data in the datasets. If our dataset contains some missing data, then it may create a huge problem for our machine learning model

To handle missing values, we will use Scikit-learn library in our code, which contains various libraries for building machine learning models. Here we will use Imputer class of sklearn.preprocessing library. Below is the code for it:

from sklearn.preprocessing import Imputer

imputer= Imputer(missing\_values ='NaN', strategy='mean', axis = 0)

imputerimputer= imputer.fit(x[:, 1:3])

x[:, 1:3]= imputer.transform(x[:, 1:3])

Output:

array([['India', 38.0, 68000.0],

['France', 43.0, 45000.0],

['Germany', 30.0, 54000.0],

['France', 48.0, 65000.0],

['Germany', 40.0, 65222.22222222222],

['India', 35.0, 58000.0],

['Germany', 41.111111111111114, 53000.0],

['France', 49.0, 79000.0],

['India', 50.0, 88000.0],

['France', 37.0, 77000.0]], dtype=object

5) Encoding Categorical data:

Categorical data is data which has some categories such as, in our dataset there are two categorical variable, Country, and Purchased.

Since machine learning model completely works on mathematics and numbers, but if our dataset would have a categorical variable, then it may create trouble while building the model. So it is necessary to encode these categorical variables into numbers.

For Country variable:

Firstly, we will convert the country variables into categorical data. So to do this, we will use LabelEncoder() class from preprocessing library.

from sklearn.preprocessing import LabelEncoder

label\_encoder\_x= LabelEncoder()

x[:, 0]= label\_encoder\_x.fit\_transform(x[:, 0]) = label\_encoder\_x.fit\_transform(x[:, 0])

Output:

array([[2, 38.0, 68000.0],

[0, 43.0, 45000.0],

[1, 30.0, 54000.0],

[0, 48.0, 65000.0],

[1, 40.0, 65222.22222222222],

[2, 35.0, 58000.0],

[1, 41.111111111111114, 53000.0],

[0, 49.0, 79000.0],

[2, 50.0, 88000.0],

[0, 37.0, 77000.0]], dtype=object)

Dummy Variables:

Dummy variables are those variables which have values 0 or 1. The 1 value gives the presence of that variable in a particular column, and rest variables become 0. With dummy encoding, we will have a number of columns equal to the number of categories.

In our dataset, we have 3 categories so it will produce three columns having 0 and 1 values. For Dummy Encoding, we will use OneHotEncoder class of preprocessing library.

Output:

array([[0.00000000e+00, 0.00000000e+00, 1.00000000e+00, 3.80000000e+01,

6.80000000e+04],

[1.00000000e+00, 0.00000000e+00, 0.00000000e+00, 4.30000000e+01,

4.50000000e+04],

[0.00000000e+00, 1.00000000e+00, 0.00000000e+00, 3.00000000e+01,

5.40000000e+04],

[1.00000000e+00, 0.00000000e+00, 0.00000000e+00, 4.80000000e+01,

6.50000000e+04],

[0.00000000e+ 00, 1.00000000e+00, 0.00000000e+00, 4.00000000e+01,

6.52222222e+04],

[0.00000000e+00, 0.00000000e+00, 1.00000000e+00, 3.50000000e+01,

5.80000000e+04],

[0.00000000e+00, 1.00000000e+00, 0.00000000e+00, 4.11111111e+01,

5.30000000e+04],

[1.00000000e+00, 0.00000000e+00, 0.00000000e+00, 4.90000000e+01,

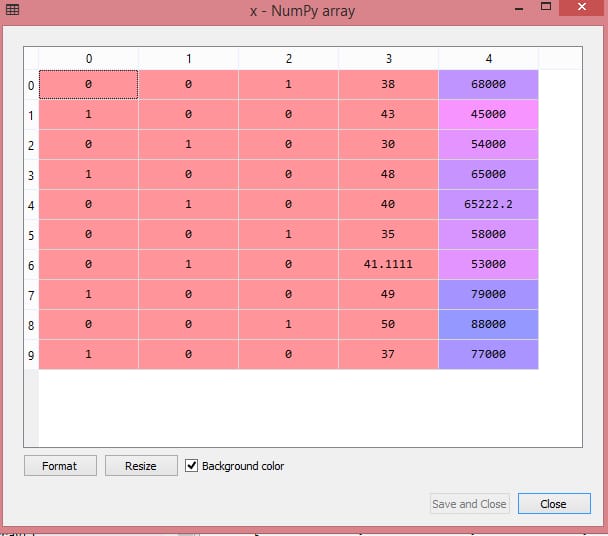
7.90000000e+04],

[0.00000000e+00, 0.00000000e+00, 1.00000000e+00, 5.00000000e+01,

8.80000000e+04],

[1.00000000e+00, 0.00000000e+00, 0.00000000e+00, 3.70000000e+01,

7.70000000e+04]]



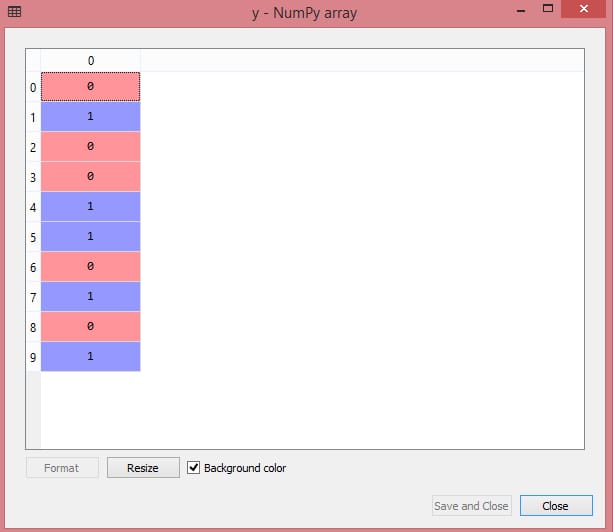
For Purchased Variable:

labelencoder\_y= LabelEncoder()

y= labelencoder\_y.fit\_transform(y)

Output

array([0, 1, 0, 0, 1, 1, 0, 1, 0, 1])



6) Splitting the Dataset into the Training set and Test set

In machine learning data preprocessing, we divide our dataset into a training set and test set. This is one of the crucial steps of data preprocessing as by doing this, we can enhance the performance of our machine learning model.

Suppose, if we have given training to our machine learning model by a dataset and we test it by a completely different dataset. Then, it will create difficulties for our model to understand the correlations between the models.

If we train our model very well and its training accuracy is also very high, but we provide a new dataset to it, then it will decrease the performance. So we always try to make a machine learning model which performs well with the training set and also with the test dataset. Here, we can define these datasets as:

Data Preprocessing in Machine learning

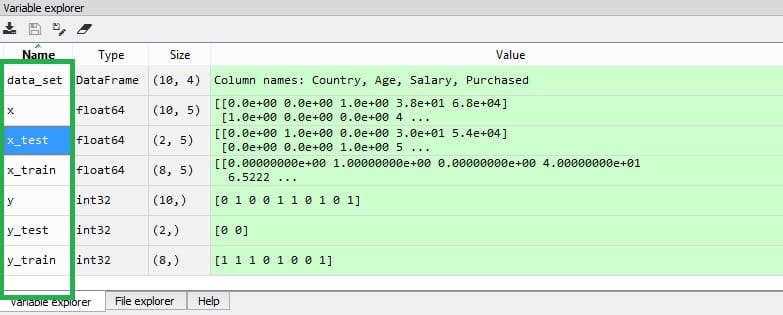
Training Set: A subset of dataset to train the machine learning model, and we already know the output.

Test set: A subset of dataset to test the machine learning model, and by using the test set, model predicts the output.

For splitting the dataset, we will use the below lines of code:

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=0)



Challenges:

Datasets often have missing values, which can adversely affect the performance of

machine learning models.

Outliers can significantly impact the mean and standard deviation, leading to skewed

results and inaccurate predictions.

Machine learning algorithms typically work with numerical data, so categorical variables need to be transformed into a numerical format.