CREATE A CHATBOT IN PYTHON

Phase 3 : Development part 1

Start building the chatbot model by loading and preprocessing the dataset

Dataset link: https://www.kaggle.com/datasets/grafstor/simple -dialogs-for-chatbot

Necessary Steps to follow :

We will begin by importing the necessary libraries we are going to use:

➢ numpy -fundamental for any kind of scientific computing with python

➢ pandas - must have tool for data analysis and manipulation

➢ matplotlib - the most complete package in python when it comes to data visualization

➢ seaborn – based on matplotlib as a higher level of virtualization tools

➢ Textvectorization – transform patch of string into indices

➢ tensorflow – python library for fast numerical computation

Importing the libraries:

```
import tensor flow as tf
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
 import seaborn as sns
from tensorflow.keras.layers import Textvectorization
import re,string
 from tensorflow.keras.layers import
LSTM,Dense,Embedding,Dropout,LayerNormalization
```

Load the dataset:

Load the csv dataset with variable name df.

To load the data, we can use pandas library

df.head()

Exploratory Data Analysis:

This includes checking for missing values, exploring the data statistics, and visualizing it.

Data Processing:

Data preprocessing is the process of transforming raw data into an understandable format. It is also an important step in data mining as we cannot work with raw data. The quality of the data should be checked before applying machine learning or data mining algorithms.

- **Steps in data preprocessing:**

➢Data Cleaning

➢Data Integration

➢Data Transformation

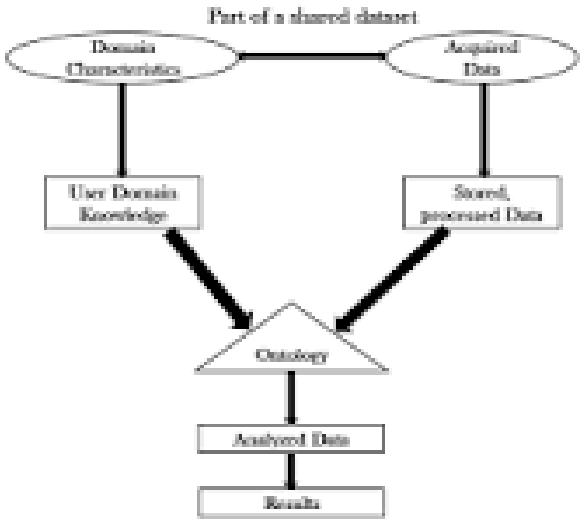
➢Data Reduction

➢Data Discretization

➢**Data Normalization**

Example for preprocessing:

data preprocessing include cleaning, instance selection, normalization, one-hot encoding, data transformation, feature extraction and feature selection.



Data Cleaning:

Data cleaning is the process of fixing or removing incorrect, corrupted, incorrectly formatted, duplicate, or incomplete data within a dataset. When combining multiple data sources, there are many opportunities for data to be duplicated or mislabeled.

Method for data cleaning:

1.Remove duplicates.

2.Remove irrelevant data.
3.Standardize capitalization.
4.Convert data type.
5.Clear formatting.
6.Fix errors.

7.Language translation.
8.Handle missing values.

PYTHON PROGRAM FOR DATA CLEANING USING THE PANDAS LIBRARY

```
Import pandas as od

 # load the dataset

df=pd.read_csv('name_dataset.csv')

 # handling missing values

df.dropna()

df.fillna(value)

 # Removing the duplicates

df.drop_duplicate()

 # Correcting the inconsistent data

 df['column_name'].replace(old value,new value,inplace=true)
```

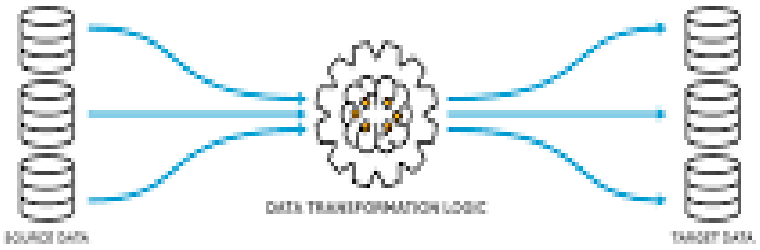
Data Integration:

Data integration refers to the process of bringing together data from multiple sources across an organization to provide a complete, accurate, and up-to-date dataset for BI, data analysis and other applications and business processes.

Purpose:

Data integration is the practice of consolidating data from disparate sources into a single dataset with the ultimate goal of providing users with consistent access and delivery of data across the spectrum of subjects and structure types, and to meet the information needs of all applications and business processes.

A common data integration technique is Extract Transform and Load (ETL) where data is physically extracted from multiple source systems, transformed into a different format, and loaded into a centralized data store.



 # merge the datasets vase dib a common column
Merge_df=pd.merge(data1,data2,on='common_column')

#perform additional transformation or manipulations as needed
Merged_data['new_column']=merged_data['column1']+merg ed_data['column2']

#save the integrated dataset

Merged_data.to_csv('integrated_dataset.csv',index=False

Data Transformation:

Data transformation is the process of converting data from one format to another, typically from the format of a source system into the required format of a destination system. Data transformation is a component of most data integration and data management tasks, such as data wrangling and data warehousing.

TYPES:

Constructive: The data transformation process adds, copies, or replicates data.

Destructive: The system deletes fields or records.

Aesthetic: The transformation standardizes the data to meet requirements or parameters.

Structural: The database is reorganized by renaming, moving, or combining columns

Encoding categorical variables

Encodedata=pd.get_dummies(data,columns=['categorical_col umn'])

normalize the data

Normalized_data=(data-data.mean())/data.std()

Data reduction:

Data reduction is a capacity optimization technique in which data is reduced to its simplest possible form to free up capacity on a storage device. There are many ways to reduce data, but the idea is very simple—squeeze as much data into physical storage as possible to maximize capacity.

STEPS:

1.editing

2.scaling

3.encoding

4. sorting

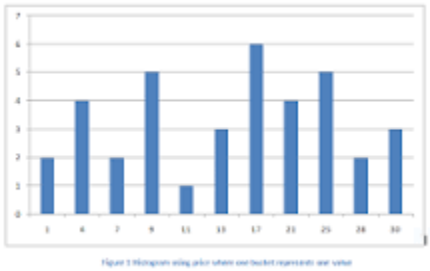
5.collating

6. producing tabular summaries.

Data discretization:

Data discretization is defined as a process of converting continuous data attribute values into a finite set of intervals and associating with each interval some specific data value.

Examples:



Discretization is one form of data transformation technique. It transforms numeric values to interval labels of conceptual labels. Ex. age can be transformed to (0-10,11-20....) or to conceptual labels like youth, adult, senior.

Use of data discretization:

Concept Hierarchy Generation

Data discretization is a method of converting attributes values of continuous data into a finite set of intervals with minimum data loss. In contrast, data binarization is used to transform the continuous and discrete attributes into binary attributes.

Data normalization:

Normalization is the process of organizing data in a database. It includes creating tables and establishing relationships between those tables according to rules designed both to protect the data and to make the database more flexible by eliminating redundancy and inconsistent dependency.

IMPORTANCE OF LOADING AND PREPROCESSING THE DATASET:

➢It ensures that the data is in a suitable format and ready for further analysis
➢Loading the dataset involves reading the data from a file or a database into memory
➢It helps to improve the accuracy, reliability and efficiency of the subsequent data analysis or machine learning tasks

CHALLENGES INVOLVED IN LOADING AND PREPROCESSING THE DATASET:

➢Inadequate or non existent data profiling
➢Missing or incomplete data
➢Invalid data values
➢Name and address standardization

➢**Inconsistent data across enterprise system**
➢**Data enrichment**
➢**Maintaining and expanding data prep process**

CONCLUSION:

We have traversed through essential steps, starting with importing the necessary libraries to facilitate data manipulation and analysis. Any potential issued through exploratory data analysis is essential for informed decision making.