Artificial intelligence

Fake News Detection Using NLP

OUR TEAM MEMBERS:

AKILA J -951321106001
DAYANA S -951321106007
DIVYA DHARSHINI R -951321106008
JERLIN PRABHA J -951321106018
RATHINA MALA B -951321106039
MENTOR:
Dr.M.RUBAN GLADWIN,M.E,Ph.d,(AP/ECE)

Introduction

Intelligence:”The capacity to learn and solve problems’.

Artificial intelligence is the simulation of humans intelligence by machines.

INTRODUCTION OF FAKE NEWS DETECTION :

Fake news is the intentional broadcasting of false or misleading claims as news, where the statements are purposely deceitful.

Problem definition:

The problem is to develop a fake news detection model using a Kaggle dataset. The goal is to distinguish between genuine and fake news
articles based on their titles and text. This project involves using natural language processing (NLP) techniques to preprocess the text data,
building a machine learning model for classification, and evaluating the model's performance.

- Input: News items, social contexts and associated side information
- Output: One of two labels: ‘fake’ or ‘real’.

Design thinking:

1. Data Source: Choose the fake news dataset available on Kaggle, containing articles titles and text, along with their labels (genuine or fake).
2. Data Preprocessing: Clean and preprocess the textual data to prepare it for analysis.
3. Feature Extraction: Utilize techniques like TF-IDF (Term Frequency-Inverse Document Frequency) or word embeddings to convert text into numerical features.
4. Model Selection: Select a suitable classification algorithm (e.g., Logistic Regression, Random Forest, or Neural Networks) for the fake news detection task.
5. Model Training: Train the selected model using the preprocessed data.
6. Evaluation: Evaluate the model's performance using metrics like accuracy, precision, recall, F1-score, and ROC-AUC.

Source code:

```
This Python 3 environment comes with many helpful analytics libraries installed
# It is defined by the kaggle/python Docker image: https://github.com/kaggle/docker-python
# For example, here's several helpful packages to load
import warnings
warnings.filterwarnings('ignore')
import numpy as np # linear algebra
import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)
import nltk
from nltk.sentiment import SentimentIntensityAnalyzer
import warnings
warnings.filterwarnings("ignore")
from sklearn.model_selection import train_test_split
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.svm import SVC
from sklearn.metrics import accuracy_score
```

Source code: ☝

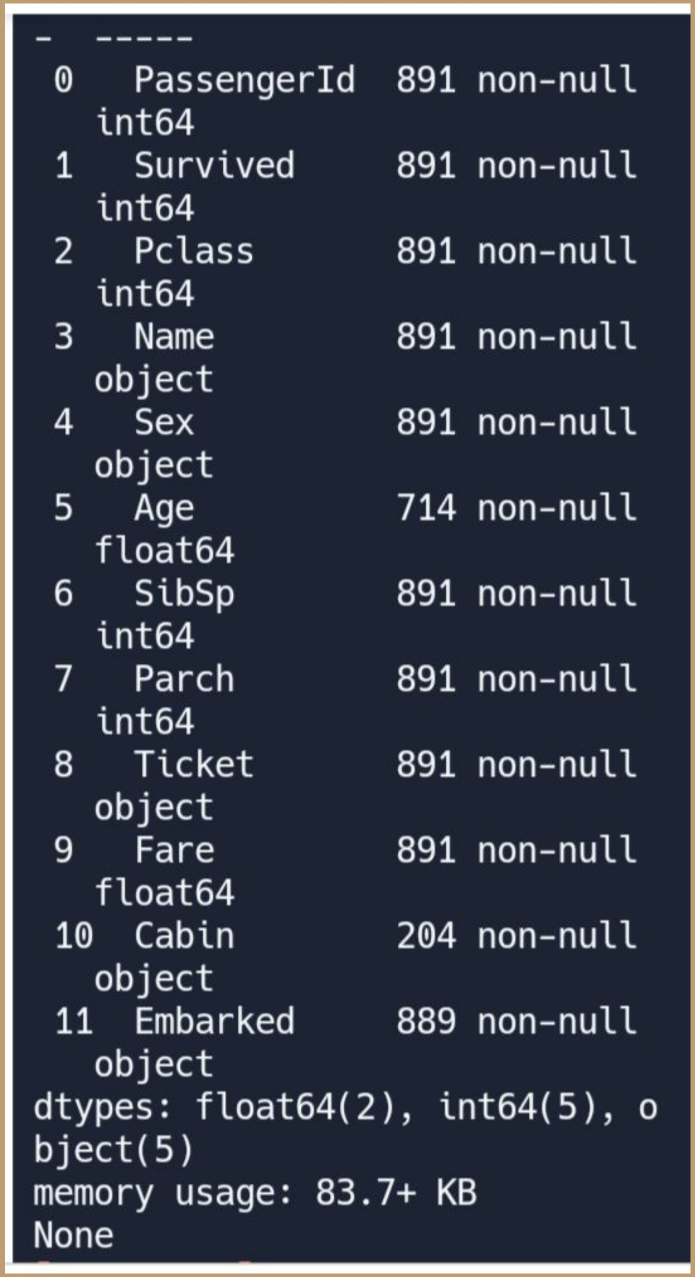
```
from sklearn.metrics import accuracy_score, classification_report
# Input data files are available in the read-only "../input/" directory
# For example, running this (by clicking run or pressing Shift+Enter) will list all files under the input directory
import os
for dirname, _, filenames in os.walk('/kaggle/input'):
 for filename in filenames:
 print(os.path.join(dirname, filename))
# You can write up to 20GB to the current directory (/kaggle/working/) that gets preserved as output when you create a version using "Save & Run All"
# You can also write temporary files to /kaggle/temp/, but they won't be saved outside of the current session
/kaggle/input/fake-and-real-news-dataset/True.csv
/kaggle/input/fake-and-real-news-dataset/Fake.csv
```

Source code:

Loading Data:

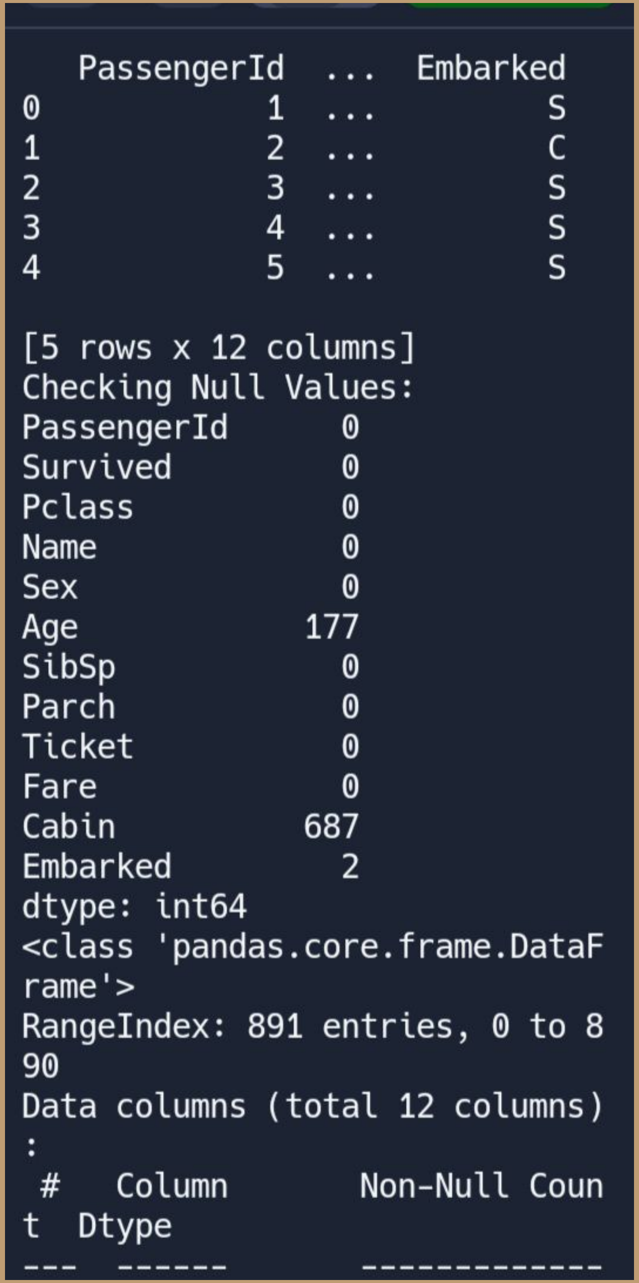
```
true = pd.read_csv('/kaggle/input/fake-and-real-news-dataset/True.csv')
fake = pd.read_csv('/kaggle/input/fake-and-real-news-dataset/Fake.csv')In [3]:
```

OUTPUT:



```
-  -----
 0   PassengerId  891 non-null
   int64
 1   Survived     891 non-null
   int64
 2   Pclass       891 non-null
   int64
 3   Name         891 non-null
   object
 4   Sex          891 non-null
   object
 5   Age          714 non-null
   float64
 6   SibSp        891 non-null
   int64
 7   Parch        891 non-null
   int64
 8   Ticket       891 non-null
   object
 9   Fare         891 non-null
   float64
 10  Cabin        204 non-null
   object
 11  Embarked     889 non-null
   object
dtypes: float64(2), int64(5), o
bject(5)
memory usage: 83.7+ KB
None
```

Screen shot for output



```
   PassengerId  ...  Embarked
0            1  ...         S
1            2  ...         C
2            3  ...         S
3            4  ...         S
4            5  ...         S

[5 rows x 12 columns]
Checking Null Values:
PassengerId      0
Survived         0
Pclass           0
Name             0
Sex              0
Age            177
SibSp            0
Parch            0
Ticket           0
Fare             0
Cabin          687
Embarked         2
dtype: int64
<class 'pandas.core.frame.DataF
rame'>
RangeIndex: 891 entries, 0 to 8
90
Data columns (total 12 columns)
:
 #   Column       Non-Null Coun
t  Dtype
---  ------       -------------
```

Thank you