**SOURCE CODE**

import streamlit as st

import numpy as np

import pandas as pd

from sklearn.preprocessing import LabelEncoder

from sklearn.model\_selection import train\_test\_split

from sklearn.feature\_extraction.text import TfidfVectorizer

from sklearn.multiclass import OneVsRestClassifier

from sklearn.neighbors import KNeighborsClassifier

from sklearn import metrics

import nltk

import re

from nltk.corpus import stopwords

import string

# Function to clean text

def clean(text):

text = re.sub('http\S+\s\*', ' ', text)

text = re.sub('RT|cc', ' ', text)

text = re.sub('#\S+', '', text)

text = re.sub('@\S+', '', text)

text = re.sub('[%s]' % re.escape("""!"#$%&'()\*=,-./:;<=>?@[\]^\_`{|}~"""), ' ', text)

text = re.sub('\s+', ' ', text)

text = re.sub(r'[^\x00-\x7f]', r' ', text)

return text

# Main function

def main():

st.title("Resume Category Prediction")

st.markdown(

"""

<style>

.reportview-container {

background: url(r"C:/Users/hp/Downloads/resume.jpeg") no-repeat center center;

background-size: cover;

}

</style>

""",

unsafe\_allow\_html=True

)

# File uploader

uploaded\_file = st.file\_uploader("Upload CSV file", type="csv")

if uploaded\_file is not None:

data = pd.read\_csv(uploaded\_file)

# Clean text

data['clean text'] = data.Resume.apply(lambda x: clean(x))

# Model training

var = ['Category']

le = LabelEncoder()

for i in var:

data[i] = le.fit\_transform(data[i])

text = data['clean text'].values

target = data['Category'].values

vect = TfidfVectorizer(

sublinear\_tf=True,

stop\_words='english',

max\_features=2000)

vect.fit(text)

Word\_feature = vect.transform(text)

x\_train, x\_test, y\_train, y\_test = train\_test\_split(Word\_feature, target, random\_state=0, test\_size=0.2)

model = OneVsRestClassifier(KNeighborsClassifier())

model.fit(x\_train, y\_train)

prediction = model.predict(x\_test)

training\_score = model.score(x\_train, y\_train)

testing\_score = model.score(x\_test, y\_test)

st.write("Training Accuracy: {:.2f}".format(training\_score))

st.write("Testing Accuracy: {:.2f}".format(testing\_score))

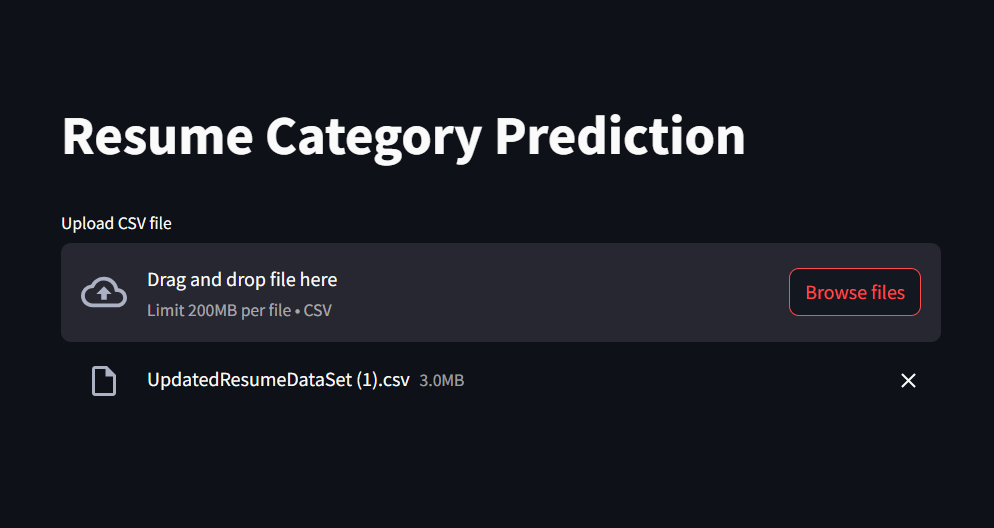
if \_\_name\_\_ == '\_\_main\_\_':

main()

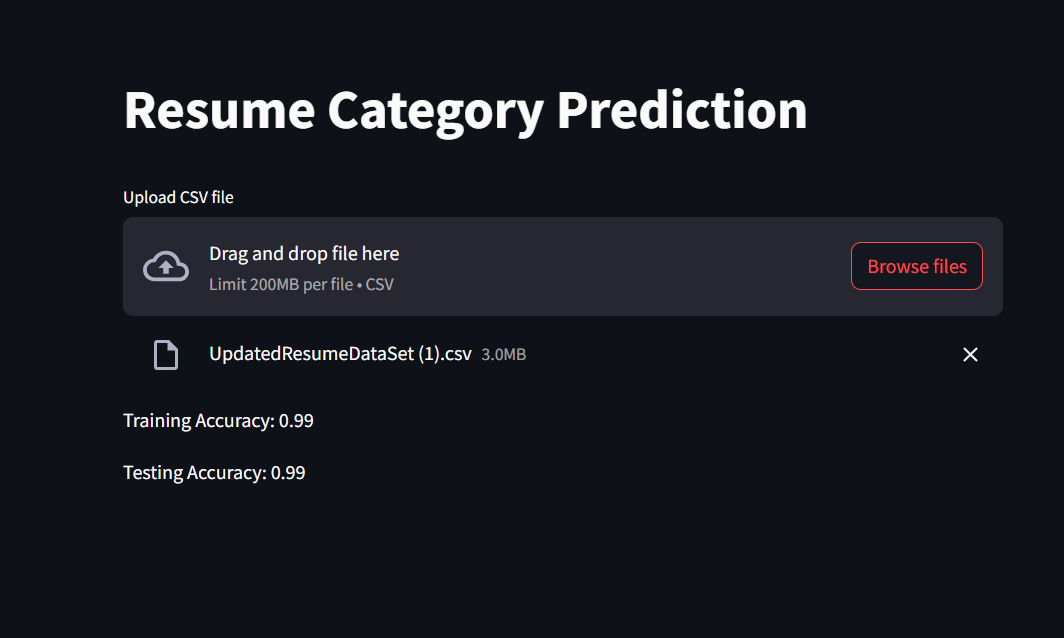
**OUTPUT**

****

**Figure 1.1 output window**

****

**Figure 1.2 uploading the csv file**

****

**Figure 1.3 result**