

B3 -Machine Learning Based Approaches for Detecting COVID-19 using Clinical Text Data

```
In [2]: from tkinter import messagebox
from tkinter import *
from tkinter import simpledialog
import tkinter
import matplotlib.pyplot as plt
import numpy as np
from tkinter import ttk
from tkinter import filedialog
import pandas as pd
from sklearn.model_selection import train_test_split
from string import punctuation
from nltk.corpus import stopwords
import nltk
from nltk.stem import WordNetLemmatizer
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.preprocessing import LabelEncoder
from sklearn.metrics import precision_score
from sklearn.metrics import recall_score
from sklearn.metrics import f1_score
from sklearn.metrics import accuracy_score
from sklearn.linear_model import LogisticRegression
from sklearn import svm
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier
import seaborn as sns
from sklearn.linear_model import SGDClassifier
from sklearn.naive_bayes import MultinomialNB
from sklearn.ensemble import BaggingClassifier
from sklearn.ensemble import AdaBoostClassifier

main = Tk()
main.title("Machine Learning Based Approaches for Detecting COVID-19 using Clinical Text Data")
main.geometry("1300x1200")

global filename
global X, Y
global X_train, X_test, y_train, y_test
accuracy = []
precision = []
recall = []
fscore = []

stop_words = set(stopwords.words('english'))
lemmatizer = WordNetLemmatizer()

disease = ['Pneumonia/Viral/COVID-19', 'Pneumonia/Viral/MERS-CoV', 'Pneumonia/Viral/SARS-CoV-2']
sentence = []
textdata = []
labels = []
global covid, ards, sars, both

def getLabel(label):
    global covid, ards, sars, both
    output = ''
```

```

if label == 'Pneumonia/Viral/COVID-19':
    output = 'COVID'
    covid = covid + 1
if label == 'Pneumonia/Viral/MERS-CoV':
    output = 'BOTH'
    both = both + 1
if label == 'Pneumonia/Viral/SARS':
    output = 'SARS'
    sars = sars + 1
if label == 'Pneumonia/Bacterial/Staphylococcus/MRSA':
    output = 'ARDS'
    ards = ards
return output

def cleanPost(doc):
    tokens = doc.split()
    table = str.maketrans('', '', punctuation)
    tokens = [w.translate(table) for w in tokens]
    tokens = [word for word in tokens if word.isalpha()]
    tokens = [w for w in tokens if not w in stop_words]
    tokens = [word for word in tokens if len(word) > 1]
    tokens = [lemmatizer.lemmatize(token) for token in tokens]
    tokens = ' '.join(tokens)
    return tokens

def uploadDataset():
    global filename
    text.delete('1.0', END)
    filename = filedialog.askopenfilename(initialdir="dataset")
    dataset = pd.read_csv(filename)
    for i in range(len(dataset)):
        msg = dataset.get_value(i, 'clinical_notes')
        label = dataset.get_value(i, 'finding')
        msg = str(msg)
        msg = msg.strip()
        text.insert(END, msg+"\n")

def preprocess():
    sentence.clear()
    textdata.clear()
    labels.clear()
    text.delete('1.0', END)
    global covid, ards, sars, both
    covid = 0
    ards = 0
    sars = 0
    both = 0
    dataset = pd.read_csv(filename)
    for i in range(len(dataset)):
        msg = dataset.get_value(i, 'clinical_notes')
        label = dataset.get_value(i, 'finding')
        msg = str(msg)
        msg = msg.strip().lower()
        if str(label) in disease and msg != 'nan':
            if msg not in sentence:
                sentence.append(msg)
                lbl = getLabel(str(label))
                labels.append(lbl)
                clean = cleanPost(msg)
                textdata.append(clean)
                text.insert(END, clean+"==== "+lbl+"\n")
    f = open("findings.txt", "w")
    f.write("COVID, ARDS, SARS, BOTH\n"+str(covid)+" "+str(ards)+" "+str(sars)+" "+str(both))

```

```

f.close()
df = pd.read_csv("findings.txt")
ax = sns.boxplot(data=df)
plt.show()

def featureEng():
    text.delete('1.0', END)
    global X_train, X_test, y_train, y_test
    stopwords=stopwords = nltk.corpus.stopwords.words("english")
    vectorizer = TfidfVectorizer(stop_words=stopwords, use_idf=True, smooth_idf=False)
    tfidf = vectorizer.fit_transform(textdata).toarray()
    df = pd.DataFrame(tfidf, columns=vectorizer.get_feature_names())
    text.insert(END, str(df))
    print(df.shape)
    df = df.values
    X = df[:, 0:40]
    Y = np.asarray(labels)
    le = LabelEncoder()
    Y = le.fit_transform(Y)
    print(Y)
    indices = np.arange(X.shape[0])
    np.random.shuffle(indices)
    X = X[indices]
    Y = Y[indices]
    X_train, X_test, y_train, y_test = train_test_split(X, Y, test_size=0.2)

def runTraditional():
    text.delete('1.0', END)
    accuracy.clear()
    precision.clear()
    recall.clear()
    fscore.clear()

    cls1 = MultinomialNB()
    cls1.fit(X_train, y_train)
    predict = cls1.predict(X_test)
    acc = accuracy_score(y_test, predict)*100
    p = precision_score(y_test, predict, average='macro') * 100
    r = recall_score(y_test, predict, average='macro') * 100
    f = f1_score(y_test, predict, average='macro') * 100
    accuracy.append(acc)
    precision.append(p)
    recall.append(r)
    fscore.append(f)
    text.insert(END, 'MultinomialNB Accuracy : '+str(acc)+"\n")
    text.insert(END, 'MultinomialNB Precision : '+str(p)+"\n")
    text.insert(END, 'MultinomialNB Recall : '+str(r)+"\n")
    text.insert(END, 'MultinomialNB F1Score : '+str(f)+"\n\n")

    cls2 = DecisionTreeClassifier()
    cls2.fit(X_train, y_train)
    predict = cls2.predict(X_test)
    print(X_test)
    acc = accuracy_score(y_test, predict)*100
    p = precision_score(y_test, predict, average='macro') * 100
    r = recall_score(y_test, predict, average='macro') * 100
    f = f1_score(y_test, predict, average='macro') * 100
    accuracy.append(acc)
    precision.append(p)
    recall.append(r)
    fscore.append(f)
    text.insert(END, 'Decision Tree Accuracy : '+str(acc)+"\n")
    text.insert(END, 'Decision Tree Precision : '+str(p)+"\n")
    text.insert(END, 'Decision Tree Recall : '+str(r)+"\n")

```

```

text.insert(END,'Decision Tree F1Score   : '+str(f)+"\n\n")

def runClassical():
    text.delete('1.0', END)
    global cls3
    cls3 = SGDClassifier(class_weight='balanced')
    cls3.fit(X_train, y_train)
    predict = cls3.predict(X_test)
    acc = accuracy_score(y_test,predict)*100
    p = precision_score(y_test, predict,average='macro') * 100
    r = recall_score(y_test, predict,average='macro') * 100
    f = f1_score(y_test, predict,average='macro') * 100
    accuracy.append(acc)
    precision.append(p)
    recall.append(r)
    fscore.append(f)
    text.insert(END,'SGD Accuracy   : '+str(acc)+"\n")
    text.insert(END,'SGD Precision : '+str(p)+"\n")
    text.insert(END,'SGD Recall    : '+str(r)+"\n")
    text.insert(END,'SGD F1Score   : '+str(f)+"\n\n")

def graph():
    '''
    df = pd.DataFrame([[ 'Precision','Logistic Regression',precision[0]],['Recall',
        [ 'Precision','Naive Bayes',precision[1]],['Recall','Naive Bayes',recall[1]],
        [ 'Precision','SVM',precision[2]],['Recall','SVM',recall[2]],
        [ 'Precision','Decision Tree',precision[3]],['Recall','Decision Tree',recall[3]]],columns=[ 'Metrics','Algorithms','Value'])
    '''
    df = pd.DataFrame([[ 'Logistic Regression','Precision',precision[0]],['Logistic Regression','Recall',recall[0]],
        [ 'Naive Bayes','Precision',precision[1]],['Naive Bayes','Recall',recall[1]],
        [ 'SVM','Precision',precision[2]],['SVM','Recall',recall[2]],
        [ 'Decision Tree','Precision',precision[3]],['Decision Tree','Recall',recall[3]],
        [ 'Bagging','Precision',precision[4]],['Bagging','Recall',recall[4]],
        [ 'AdaBoost','Precision',precision[5]],['AdaBoost','Recall',recall[5]],
        [ 'Random Forest','Precision',precision[6]],['Random Forest','Recall',recall[6]],
        [ 'Stochastic Gradient','Precision',precision[7]],['Stochastic Gradient','Recall',recall[7]]],columns=[ 'Metrics','Algorithms','Value'])
    df.pivot("Metrics", "Algorithms", "Value").plot(kind='bar')
    plt.show()

def predictnew():
    global cls3
    fn = filedialog.askopenfilename(initialdir="dataset")
    tf = pd.read_csv(fn)
    tf =tf.values
    X=tf[:, 0:40]
    predict = cls3.predict(X)
    text.insert(END,'Predicted type is   : '+str(predict)+"\n")

font = ('times', 15, 'bold')
title = Label(main, text='Machine Learning Based Approaches for Detecting COVID-19')
#title.config(bg='powder blue', fg='olive drab')
title.config(font=font)
title.config(height=3, width=120)
title.place(x=0,y=5)

font1 = ('times', 13, 'bold')
ff = ('times', 12, 'bold')

uploadButton = Button(main, text="Upload Covid-19 Dataset", command=uploadDataset)
uploadButton.place(x=20,y=100)

```

```

uploadButton.config(font=ff)

processButton = Button(main, text="Preprocess Dataset", command=preprocess)
processButton.place(x=20,y=150)
processButton.config(font=ff)

featureButton = Button(main, text="Feature Engineering", command=featureEng)
featureButton.place(x=20,y=200)
featureButton.config(font=ff)

traButton = Button(main, text="Run Logistic Regression, Naive Bayes, SVM & Decision", command=tra)
traButton.place(x=20,y=250)
traButton.config(font=ff)

clsButton = Button(main, text="Run Bagging, Adaboost, Random Forest & Gradient Boosting", command=cls)
clsButton.place(x=20,y=300)
clsButton.config(font=ff)

clsButton = Button(main, text="Predict for New Data", command=predictnew)
clsButton.place(x=20,y=350)
clsButton.config(font=ff)

graphButton = Button(main, text="Comparative Analysis Graph", command=graph)
graphButton.place(x=20,y=400)
graphButton.config(font=ff)

font1 = ('times', 12, 'bold')
text=Text(main,height=30,width=85)
scroll=Scrollbar(text)
text.configure(yscrollcommand=scroll.set)
text.place(x=550,y=100)
text.config(font=font1)

main.config()
main.mainloop()

```

```

Exception in Tkinter callback
Traceback (most recent call last):
  File "C:\Users\hi\anaconda3\lib\tkinter\__init__.py", line 1892, in __call__
    return self.func(*args)
  File "C:\Users\hi\AppData\Local\Temp\ipykernel_1716\3690767033.py", line 86, in
uploadDataset
    msg = dataset.get_value(i, 'clinical_notes')
  File "C:\Users\hi\anaconda3\lib\site-packages\pandas\core\generic.py", line 557
5, in __getattr__
    return object.__getattribute__(self, name)
AttributeError: 'DataFrame' object has no attribute 'get_value'
Exception in Tkinter callback
Traceback (most recent call last):
  File "C:\Users\hi\anaconda3\lib\tkinter\__init__.py", line 1892, in __call__
    return self.func(*args)
  File "C:\Users\hi\AppData\Local\Temp\ipykernel_1716\3690767033.py", line 86, in
uploadDataset
    msg = dataset.get_value(i, 'clinical_notes')
  File "C:\Users\hi\anaconda3\lib\site-packages\pandas\core\generic.py", line 557
5, in __getattr__
    return object.__getattribute__(self, name)
AttributeError: 'DataFrame' object has no attribute 'get_value'
Exception in Tkinter callback
Traceback (most recent call last):
  File "C:\Users\hi\anaconda3\lib\tkinter\__init__.py", line 1892, in __call__
    return self.func(*args)
  File "C:\Users\hi\AppData\Local\Temp\ipykernel_1716\3690767033.py", line 105, in
preprocess
    msg = dataset.get_value(i, 'clinical_notes')
  File "C:\Users\hi\anaconda3\lib\site-packages\pandas\core\generic.py", line 557
5, in __getattr__
    return object.__getattribute__(self, name)
AttributeError: 'DataFrame' object has no attribute 'get_value'
Exception in Tkinter callback
Traceback (most recent call last):
  File "C:\Users\hi\anaconda3\lib\tkinter\__init__.py", line 1892, in __call__
    return self.func(*args)
  File "C:\Users\hi\AppData\Local\Temp\ipykernel_1716\3690767033.py", line 84, in
uploadDataset
    dataset = pd.read_csv(filename)
  File "C:\Users\hi\anaconda3\lib\site-packages\pandas\util\_decorators.py", line
311, in wrapper
    return func(*args, **kwargs)
  File "C:\Users\hi\anaconda3\lib\site-packages\pandas\io\parsers\readers.py", lin
e 680, in read_csv
    return _read(filepath_or_buffer, kwds)
  File "C:\Users\hi\anaconda3\lib\site-packages\pandas\io\parsers\readers.py", lin
e 575, in _read
    parser = TextFileReader(filepath_or_buffer, **kwds)
  File "C:\Users\hi\anaconda3\lib\site-packages\pandas\io\parsers\readers.py", lin
e 933, in __init__
    self._engine = self._make_engine(f, self.engine)
  File "C:\Users\hi\anaconda3\lib\site-packages\pandas\io\parsers\readers.py", lin
e 1217, in _make_engine
    self.handles = get_handle( # type: ignore[call-overload]
  File "C:\Users\hi\anaconda3\lib\site-packages\pandas\io\common.py", line 789, in
get_handle
    handle = open(
FileNotFoundError: [Errno 2] No such file or directory: ''
Exception in Tkinter callback
Traceback (most recent call last):
  File "C:\Users\hi\anaconda3\lib\tkinter\__init__.py", line 1892, in __call__
    return self.func(*args)
  File "C:\Users\hi\AppData\Local\Temp\ipykernel_1716\3690767033.py", line 86, in

```

```

uploadDataset
    msg = dataset.get_value(i, 'clinical_notes')
    File "C:\Users\hi\anaconda3\lib\site-packages\pandas\core\generic.py", line 557
5, in __getattr__
    return object.__getattribute__(self, name)
AttributeError: 'DataFrame' object has no attribute 'get_value'
Exception in Tkinter callback
Traceback (most recent call last):
  File "C:\Users\hi\anaconda3\lib\tkinter\__init__.py", line 1892, in __call__
    return self.func(*args)
  File "C:\Users\hi\AppData\Local\Temp\ipykernel_1716\3690767033.py", line 86, in
uploadDataset
    msg = dataset.get_value(i, 'clinical_notes')
    File "C:\Users\hi\anaconda3\lib\site-packages\pandas\core\generic.py", line 557
5, in __getattr__
    return object.__getattribute__(self, name)
AttributeError: 'DataFrame' object has no attribute 'get_value'
Exception in Tkinter callback
Traceback (most recent call last):
  File "C:\Users\hi\anaconda3\lib\tkinter\__init__.py", line 1892, in __call__
    return self.func(*args)
  File "C:\Users\hi\AppData\Local\Temp\ipykernel_1716\3690767033.py", line 105, in
preprocess
    msg = dataset.get_value(i, 'clinical_notes')
    File "C:\Users\hi\anaconda3\lib\site-packages\pandas\core\generic.py", line 557
5, in __getattr__
    return object.__getattribute__(self, name)
AttributeError: 'DataFrame' object has no attribute 'get_value'
Exception in Tkinter callback
Traceback (most recent call last):
  File "C:\Users\hi\anaconda3\lib\tkinter\__init__.py", line 1892, in __call__
    return self.func(*args)
  File "C:\Users\hi\AppData\Local\Temp\ipykernel_1716\3690767033.py", line 105, in
preprocess
    msg = dataset.get_value(i, 'clinical_notes')
    File "C:\Users\hi\anaconda3\lib\site-packages\pandas\core\generic.py", line 557
5, in __getattr__
    return object.__getattribute__(self, name)
AttributeError: 'DataFrame' object has no attribute 'get_value'
Exception in Tkinter callback
Traceback (most recent call last):
  File "C:\Users\hi\anaconda3\lib\tkinter\__init__.py", line 1892, in __call__
    return self.func(*args)
  File "C:\Users\hi\AppData\Local\Temp\ipykernel_1716\3690767033.py", line 105, in
preprocess
    msg = dataset.get_value(i, 'clinical_notes')
    File "C:\Users\hi\anaconda3\lib\site-packages\pandas\core\generic.py", line 557
5, in __getattr__
    return object.__getattribute__(self, name)
AttributeError: 'DataFrame' object has no attribute 'get_value'
Exception in Tkinter callback
Traceback (most recent call last):
  File "C:\Users\hi\anaconda3\lib\tkinter\__init__.py", line 1892, in __call__
    return self.func(*args)
  File "C:\Users\hi\AppData\Local\Temp\ipykernel_1716\3690767033.py", line 129, in
featureEng
    tfidf = vectorizer.fit_transform(textdata).toarray()
    File "C:\Users\hi\anaconda3\lib\site-packages\sklearn\feature_extraction\text.p
y", line 2077, in fit_transform
    X = super().fit_transform(raw_documents)
    File "C:\Users\hi\anaconda3\lib\site-packages\sklearn\feature_extraction\text.p
y", line 1330, in fit_transform
    vocabulary, X = self._count_vocab(raw_documents, self.fixed_vocabulary_)
    File "C:\Users\hi\anaconda3\lib\site-packages\sklearn\feature_extraction\text.p

```

```

y", line 1220, in _count_vocab
    raise ValueError(
ValueError: empty vocabulary; perhaps the documents only contain stop words
Exception in Tkinter callback
Traceback (most recent call last):
  File "C:\Users\hi\anaconda3\lib\tkinter\__init__.py", line 1892, in __call__
    return self.func(*args)
  File "C:\Users\hi\AppData\Local\Temp\ipykernel_1716\3690767033.py", line 84, in
uploadDataset
    dataset = pd.read_csv(filename)
  File "C:\Users\hi\anaconda3\lib\site-packages\pandas\util\_decorators.py", line
311, in wrapper
    return func(*args, **kwargs)
  File "C:\Users\hi\anaconda3\lib\site-packages\pandas\io\parsers\readers.py", lin
e 680, in read_csv
    return _read(filepath_or_buffer, kwds)
  File "C:\Users\hi\anaconda3\lib\site-packages\pandas\io\parsers\readers.py", lin
e 575, in _read
    parser = TextFileReader(filepath_or_buffer, **kwds)
  File "C:\Users\hi\anaconda3\lib\site-packages\pandas\io\parsers\readers.py", lin
e 933, in __init__
    self._engine = self._make_engine(f, self.engine)
  File "C:\Users\hi\anaconda3\lib\site-packages\pandas\io\parsers\readers.py", lin
e 1217, in _make_engine
    self.handles = get_handle( # type: ignore[call-overload]
  File "C:\Users\hi\anaconda3\lib\site-packages\pandas\io\common.py", line 789, in
get_handle
    handle = open(
FileNotFoundError: [Errno 2] No such file or directory: ''
Exception in Tkinter callback
Traceback (most recent call last):
  File "C:\Users\hi\anaconda3\lib\tkinter\__init__.py", line 1892, in __call__
    return self.func(*args)
  File "C:\Users\hi\AppData\Local\Temp\ipykernel_1716\3690767033.py", line 103, in
preprocess
    dataset = pd.read_csv(filename)
  File "C:\Users\hi\anaconda3\lib\site-packages\pandas\util\_decorators.py", line
311, in wrapper
    return func(*args, **kwargs)
  File "C:\Users\hi\anaconda3\lib\site-packages\pandas\io\parsers\readers.py", lin
e 680, in read_csv
    return _read(filepath_or_buffer, kwds)
  File "C:\Users\hi\anaconda3\lib\site-packages\pandas\io\parsers\readers.py", lin
e 575, in _read
    parser = TextFileReader(filepath_or_buffer, **kwds)
  File "C:\Users\hi\anaconda3\lib\site-packages\pandas\io\parsers\readers.py", lin
e 933, in __init__
    self._engine = self._make_engine(f, self.engine)
  File "C:\Users\hi\anaconda3\lib\site-packages\pandas\io\parsers\readers.py", lin
e 1217, in _make_engine
    self.handles = get_handle( # type: ignore[call-overload]
  File "C:\Users\hi\anaconda3\lib\site-packages\pandas\io\common.py", line 789, in
get_handle
    handle = open(
FileNotFoundError: [Errno 2] No such file or directory: ''
Exception in Tkinter callback
Traceback (most recent call last):
  File "C:\Users\hi\anaconda3\lib\tkinter\__init__.py", line 1892, in __call__
    return self.func(*args)
  File "C:\Users\hi\AppData\Local\Temp\ipykernel_1716\3690767033.py", line 103, in
preprocess
    dataset = pd.read_csv(filename)
  File "C:\Users\hi\anaconda3\lib\site-packages\pandas\util\_decorators.py", line
311, in wrapper

```



```

    return func(*args, **kwargs)
File "C:\Users\hi\anaconda3\lib\site-packages\pandas\io\parsers\readers.py", line 680, in read_csv
    return _read(filepath_or_buffer, kwds)
File "C:\Users\hi\anaconda3\lib\site-packages\pandas\io\parsers\readers.py", line 575, in _read
    parser = TextFileReader(filepath_or_buffer, **kwds)
File "C:\Users\hi\anaconda3\lib\site-packages\pandas\io\parsers\readers.py", line 933, in __init__
    self._engine = self._make_engine(f, self.engine)
File "C:\Users\hi\anaconda3\lib\site-packages\pandas\io\parsers\readers.py", line 1217, in _make_engine
    self.handles = get_handle( # type: ignore[call-overload]
File "C:\Users\hi\anaconda3\lib\site-packages\pandas\io\common.py", line 789, in get_handle
    handle = open(
FileNotFoundError: [Errno 2] No such file or directory: ''
Exception in Tkinter callback
Traceback (most recent call last):
  File "C:\Users\hi\anaconda3\lib\tkinter\__init__.py", line 1892, in __call__
    return self.func(*args)
  File "C:\Users\hi\AppData\Local\Temp\ipykernel_1716\3690767033.py", line 103, in preprocess
    dataset = pd.read_csv(filename)
  File "C:\Users\hi\anaconda3\lib\site-packages\pandas\util\_decorators.py", line 311, in wrapper
    return func(*args, **kwargs)
  File "C:\Users\hi\anaconda3\lib\site-packages\pandas\io\parsers\readers.py", line 680, in read_csv
    return _read(filepath_or_buffer, kwds)
  File "C:\Users\hi\anaconda3\lib\site-packages\pandas\io\parsers\readers.py", line 575, in _read
    parser = TextFileReader(filepath_or_buffer, **kwds)
  File "C:\Users\hi\anaconda3\lib\site-packages\pandas\io\parsers\readers.py", line 933, in __init__
    self._engine = self._make_engine(f, self.engine)
  File "C:\Users\hi\anaconda3\lib\site-packages\pandas\io\parsers\readers.py", line 1217, in _make_engine
    self.handles = get_handle( # type: ignore[call-overload]
  File "C:\Users\hi\anaconda3\lib\site-packages\pandas\io\common.py", line 789, in get_handle
    handle = open(
FileNotFoundError: [Errno 2] No such file or directory: ''
Exception in Tkinter callback
Traceback (most recent call last):
  File "C:\Users\hi\anaconda3\lib\tkinter\__init__.py", line 1892, in __call__
    return self.func(*args)
  File "C:\Users\hi\AppData\Local\Temp\ipykernel_1716\3690767033.py", line 103, in preprocess
    dataset = pd.read_csv(filename)
  File "C:\Users\hi\anaconda3\lib\site-packages\pandas\util\_decorators.py", line 311, in wrapper
    return func(*args, **kwargs)
  File "C:\Users\hi\anaconda3\lib\site-packages\pandas\io\parsers\readers.py", line 680, in read_csv
    return _read(filepath_or_buffer, kwds)
  File "C:\Users\hi\anaconda3\lib\site-packages\pandas\io\parsers\readers.py", line 575, in _read
    parser = TextFileReader(filepath_or_buffer, **kwds)
  File "C:\Users\hi\anaconda3\lib\site-packages\pandas\io\parsers\readers.py", line 933, in __init__
    self._engine = self._make_engine(f, self.engine)
  File "C:\Users\hi\anaconda3\lib\site-packages\pandas\io\parsers\readers.py", line 1217, in _make_engine

```

```

    self.handles = get_handle( # type: ignore[call-overload]
File "C:\Users\hi\anaconda3\lib\site-packages\pandas\io\common.py", line 789, in
get_handle
    handle = open(
FileNotFoundError: [Errno 2] No such file or directory: ''
Exception in Tkinter callback
Traceback (most recent call last):
  File "C:\Users\hi\anaconda3\lib\tkinter\__init__.py", line 1892, in __call__
    return self.func(*args)
  File "C:\Users\hi\AppData\Local\Temp\ipykernel_1716\3690767033.py", line 103, in
preprocess
    dataset = pd.read_csv(filename)
  File "C:\Users\hi\anaconda3\lib\site-packages\pandas\util\_decorators.py", line
311, in wrapper
    return func(*args, **kwargs)
  File "C:\Users\hi\anaconda3\lib\site-packages\pandas\io\parsers\readers.py", lin
e 680, in read_csv
    return _read(filepath_or_buffer, kwds)
  File "C:\Users\hi\anaconda3\lib\site-packages\pandas\io\parsers\readers.py", lin
e 575, in _read
    parser = TextFileReader(filepath_or_buffer, **kwds)
  File "C:\Users\hi\anaconda3\lib\site-packages\pandas\io\parsers\readers.py", lin
e 933, in __init__
    self._engine = self._make_engine(f, self.engine)
  File "C:\Users\hi\anaconda3\lib\site-packages\pandas\io\parsers\readers.py", lin
e 1217, in _make_engine
    self.handles = get_handle( # type: ignore[call-overload]
  File "C:\Users\hi\anaconda3\lib\site-packages\pandas\io\common.py", line 789, in
get_handle
    handle = open(
FileNotFoundError: [Errno 2] No such file or directory: ''

```

In []: