ML code for finding the stages of antenna

import numpy as np # linear algebra

import pandas as pd # data processing, CSV file I/O (e.g. pd.read\_csv)

import os

for dirname, \_, filenames in os.walk('/kaggle/input'):

    for filename in filenames:

        print(os.path.join(dirname, filename))

%matplotlib inline

import matplotlib.pyplot as plt

import seaborn as sns

import warnings

warnings.filterwarnings('ignore')

df = pd.read\_csv("lung\_data.csv")

df.info()

df.head()

| **e\_field** | **h\_field** | **stage\_of\_cancer** |
| --- | --- | --- |
| 0 | -0.875761 | -0.793068 | Benign |
| 1 | 2.894581 | 0.201416 | Benign |
| 2 | -0.745797 | -0.566857 | Benign |
| 3 | 1.094870 | -0.039563 | Malignant (Early) |
| 4 | 0.587965 | 0.221265 | Malignant (Mid) |
|  |  |  |  |

df['stage\_of\_cancer'].unique()

array(['Benign', 'Malignant (Early)', 'Malignant (Mid)',

'Malignant (Late)'], dtype=object)

df['stage\_of\_cancer'] = df['stage\_of\_cancer'].replace({'IA': 1, 'IB' : 1,'IIA':1,

                                                       'IIB':1,'IIIA':2,'IIIB':2,'IV':2})

df=df.fillna({'stage\_of\_cancer': 0, 'days\_to\_cancer': 0, 'race': 'Others'})

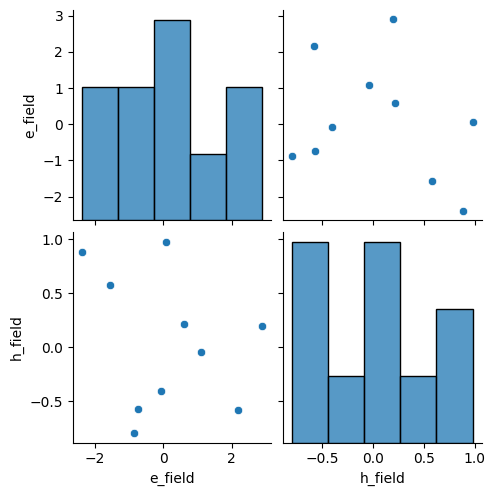
try:

       df['stage\_of\_cancer'] = df['stage\_of\_cancer'].astype('category')

except:

       pass  # Handle potential errors if conversion fails

sns.pairplot(df.drop(columns=['stage\_of\_cancer']));



numeric\_columns = df.select\_dtypes(include=[np.number]).columns

heatmap\_data = df[numeric\_columns].corr()

sns.heatmap(heatmap\_data, annot=True, cbar=False, cmap='Blues', fmt='.1f')

A blue and white squares

Description automatically generated

col=[ 'e\_field','h\_field','stage\_of\_cancer']

fig = plt.figure(figsize=(15,15))

for i in range(len(col)):

    plt.subplot(3,2,i+1)

    plt.title(col[i])

    sns.histplot(data=df,y=df[col[i]],hue='stage\_of\_cancer')

plt.tight\_layout()

plt.show()

A screenshot of a graph

Description automatically generated

# Assuming valid categories are 'Early Stage', 'Malignant'

df1 = df[df['stage\_of\_cancer'].isin(['Malignant'])]  # Filter for specific categories

df1 = df[df['stage\_of\_cancer'].isin(['Benign'])]

# it was manually coverted to integer code

sns.pairplot(df1.drop(columns=['stage\_of\_cancer']));

A graph of different sizes and colors

Description automatically generated with medium confidence

col=['e\_field', 'h\_field','stage\_of\_cancer']

fig = plt.figure(figsize=(15,15))

for i in range(len(col)):

    plt.subplot(3,2,i+1)

    plt.title(col[i])

    sns.histplot(data=df1,y=df1[col[i]],hue='stage\_of\_cancer')

plt.tight\_layout()

plt.show()

A screenshot of a graph

Description automatically generated

df2=pd.get\_dummies(df1)

df2=df2.reset\_index(drop=True)

df2.head()

| **e\_field** | **h\_field** | **stage\_of\_cancer\_Benign** | **stage\_of\_cancer\_Malignant (Early)** | **stage\_of\_cancer\_Malignant (Late)** | **stage\_of\_cancer\_Malignant (Mid)** |
| --- | --- | --- | --- | --- | --- |
| 0 | -0.875761 | -0.793068 | True | False | False | False |
| 1 | 2.894581 | 0.201416 | True | False | False | False |
| 2 | -0.745797 | -0.566857 | True | False | False | False |
| 3 | 0.066371 | 0.979153 | True | False | False | False |
| 4 | -2.388035 | 0.887376 | True | False | False | False |