hbling_UMAP

May 13, 2025

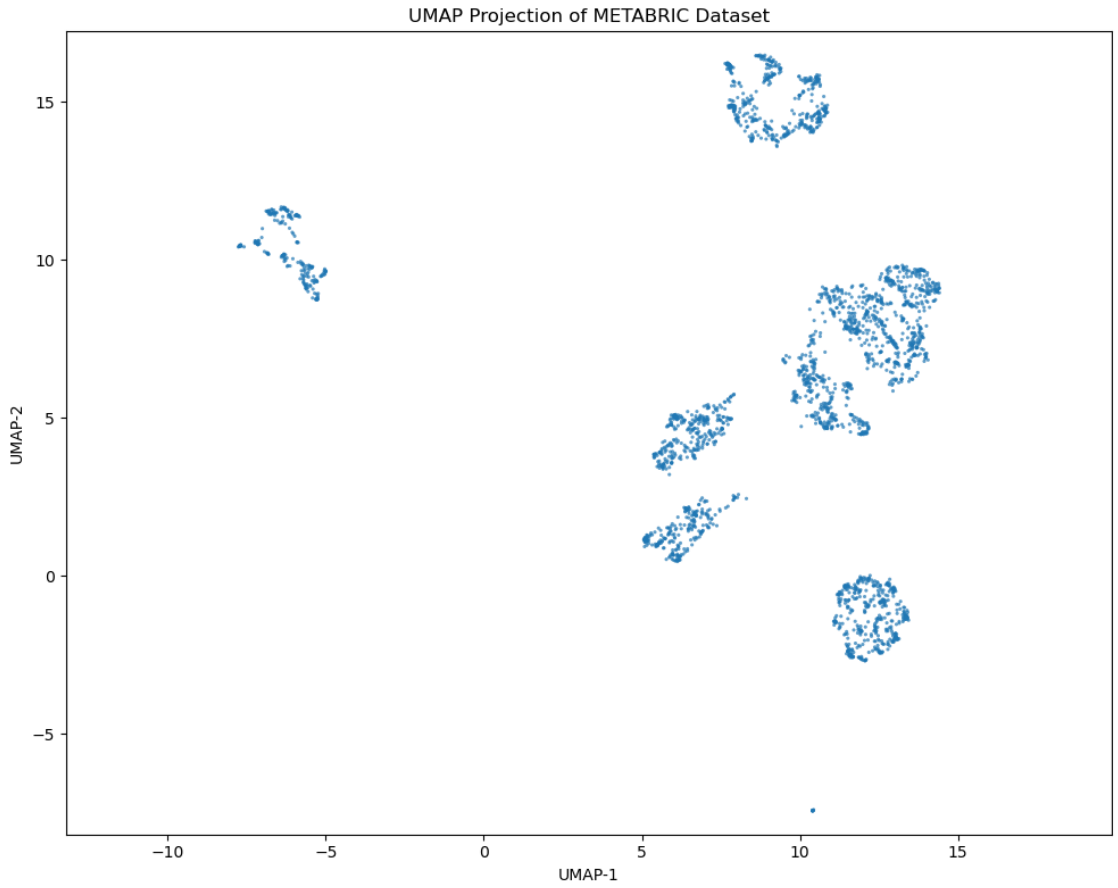
[1]:
```
!pip install umap-learn
```

```
Collecting umap-learn
  Downloading umap_learn-0.5.7-py3-none-any.whl.metadata (21 kB)
Requirement already satisfied: numpy>=1.17 in c:\users\smcca\anaconda3\lib\site-
packages (from umap-learn) (1.26.4)
Requirement already satisfied: scipy>=1.3.1 in
c:\users\smcca\anaconda3\lib\site-packages (from umap-learn) (1.13.1)
Requirement already satisfied: scikit-learn>=0.22 in
c:\users\smcca\anaconda3\lib\site-packages (from umap-learn) (1.5.1)
Requirement already satisfied: numba>=0.51.2 in
c:\users\smcca\anaconda3\lib\site-packages (from umap-learn) (0.60.0)
Collecting pynndescent>=0.5 (from umap-learn)
  Downloading pynndescent-0.5.13-py3-none-any.whl.metadata (6.8 kB)
Requirement already satisfied: tqdm in c:\users\smcca\anaconda3\lib\site-
packages (from umap-learn) (4.66.5)
Requirement already satisfied: llvmlite<0.44,>=0.43.0dev0 in
c:\users\smcca\anaconda3\lib\site-packages (from numba>=0.51.2->umap-learn)
(0.43.0)
Requirement already satisfied: joblib>=0.11 in
c:\users\smcca\anaconda3\lib\site-packages (from pynndescent>=0.5->umap-learn)
(1.4.2)
Requirement already satisfied: threadpoolctl>=3.1.0 in
c:\users\smcca\anaconda3\lib\site-packages (from scikit-learn>=0.22->umap-learn)
(3.5.0)
Requirement already satisfied: colorama in c:\users\smcca\anaconda3\lib\site-
packages (from tqdm->umap-learn) (0.4.6)
Downloading umap_learn-0.5.7-py3-none-any.whl (88 kB)
Downloading pynndescent-0.5.13-py3-none-any.whl (56 kB)
Installing collected packages: pynndescent, umap-learn
Successfully installed pynndescent-0.5.13 umap-learn-0.5.7
```

[15]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import umap
from sklearn.impute import SimpleImputer
from sklearn.preprocessing import StandardScaler
```

```
df = pd.read_csv('Breast Cancer METABRIC.csv')

binary_map = {'positive': 1, 'negative': 0}
df['ER Status'] = df['ER Status'].map(binary_map)
df['PR Status'] = df['PR Status'].map(binary_map)
df['HER2 Status'] = df['HER2 Status'].map(binary_map)

surgery_map = {'Mastectomy': 1, 'Breast Conserving': 0}
df['Type of Breast Surgery'] = df['Type of Breast Surgery'].map(surgery_map)

cellularity_map = {'High': 2, 'Moderate': 1, 'Low': 0}
df['Cellularity'] = df['Cellularity'].map(cellularity_map)

yesno_map = {'Yes': 1, 'No': 0}
df['Chemotherapy'] = df['Chemotherapy'].map(yesno_map)
df['Hormone Therapy'] = df['Hormone Therapy'].map(yesno_map)

sex_map = {'Male': 1, 'Female': 0}
df['Sex'] = df['Sex'].map(sex_map)

vital_map = {'Living': 0, 'Died of Disease': 1, 'Died of Other Causes': 2}
df["Patient's Vital Status"] = df["Patient's Vital Status"].map(vital_map)

X_numeric = df.select_dtypes(include=[np.number])
X_numeric = X_numeric.dropna(axis=1, how='all')

imputer = SimpleImputer(strategy='mean')
X_imputed = pd.DataFrame(
    imputer.fit_transform(X_numeric),
    index=X_numeric.index,
    columns=X_numeric.columns
)

scaler = StandardScaler()
X_scaled = scaler.fit_transform(X_imputed)

reducer = umap.UMAP(
    n_neighbors=15,
    min_dist=0.1,
    n_components=2,
    metric='euclidean',
    random_state=42,
    n_jobs=1
)
embedding = reducer.fit_transform(X_scaled)
```

```
plt.figure(figsize=(10, 8))
plt.scatter(
    embedding[:, 0], embedding[:, 1],
    s=5, alpha=0.7, edgecolors='none'
)
plt.gca().set_aspect('equal', 'datalim')
plt.title('UMAP Projection of METABRIC Dataset')
plt.xlabel('UMAP-1')
plt.ylabel('UMAP-2')
plt.tight_layout()
plt.show()
```

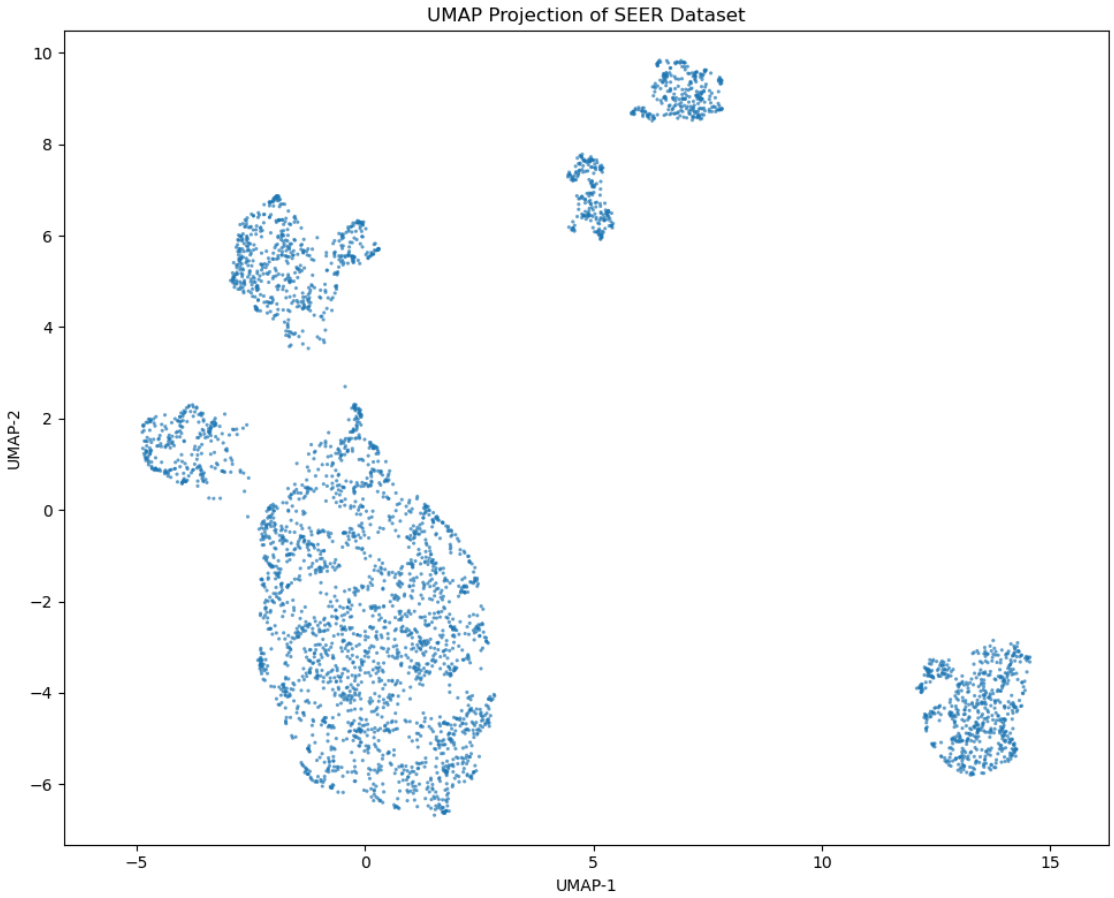


[19]:
```
df = pd.read_csv('SEER Breast Cancer Dataset .csv')
print(df.columns.tolist())
binary_map = {'positive': 1, 'negative': 0}
df['Estrogen Status'] = df['Estrogen Status'].map(binary_map)
df['Progesterone Status'] = df['Progesterone Status'].map(binary_map)
```

```
race_map = {'White': 0, 'Black': 1, 'Other (American Indian/AK Native, Asian/
 ↪Pacific Islander)': 2}
df['Race '] = df['Race '].map(race_map)

marriage_map = {'Married (including common law)': 0, 'Divorced': 1, 'Single␣
 ↪(never married)': 2, 'Widowed': 3}
df['Marital Status'] = df['Marital Status'].map(marriage_map)

vital_map = {'Alive': 0, 'Dead': 1}
df["Status"] = df["Status"].map(vital_map)

X_numeric = df.select_dtypes(include=[np.number])
X_numeric = X_numeric.dropna(axis=1, how='all')

imputer = SimpleImputer(strategy='mean')
X_imputed = pd.DataFrame(
    imputer.fit_transform(X_numeric),
    index=X_numeric.index,
    columns=X_numeric.columns
)

scaler = StandardScaler()
X_scaled = scaler.fit_transform(X_imputed)

reducer = umap.UMAP(
    n_neighbors=15,
    min_dist=0.1,
    n_components=2,
    metric='euclidean',
    random_state=42,
    n_jobs=1
)
embedding = reducer.fit_transform(X_scaled)

plt.figure(figsize=(10, 8))
plt.scatter(
    embedding[:, 0], embedding[:, 1],
    s=5, alpha=0.7, edgecolors='none'
)
plt.gca().set_aspect('equal', 'datalim')
plt.title('UMAP Projection of SEER Dataset')
plt.xlabel('UMAP-1')
plt.ylabel('UMAP-2')
plt.tight_layout()
plt.show()
```

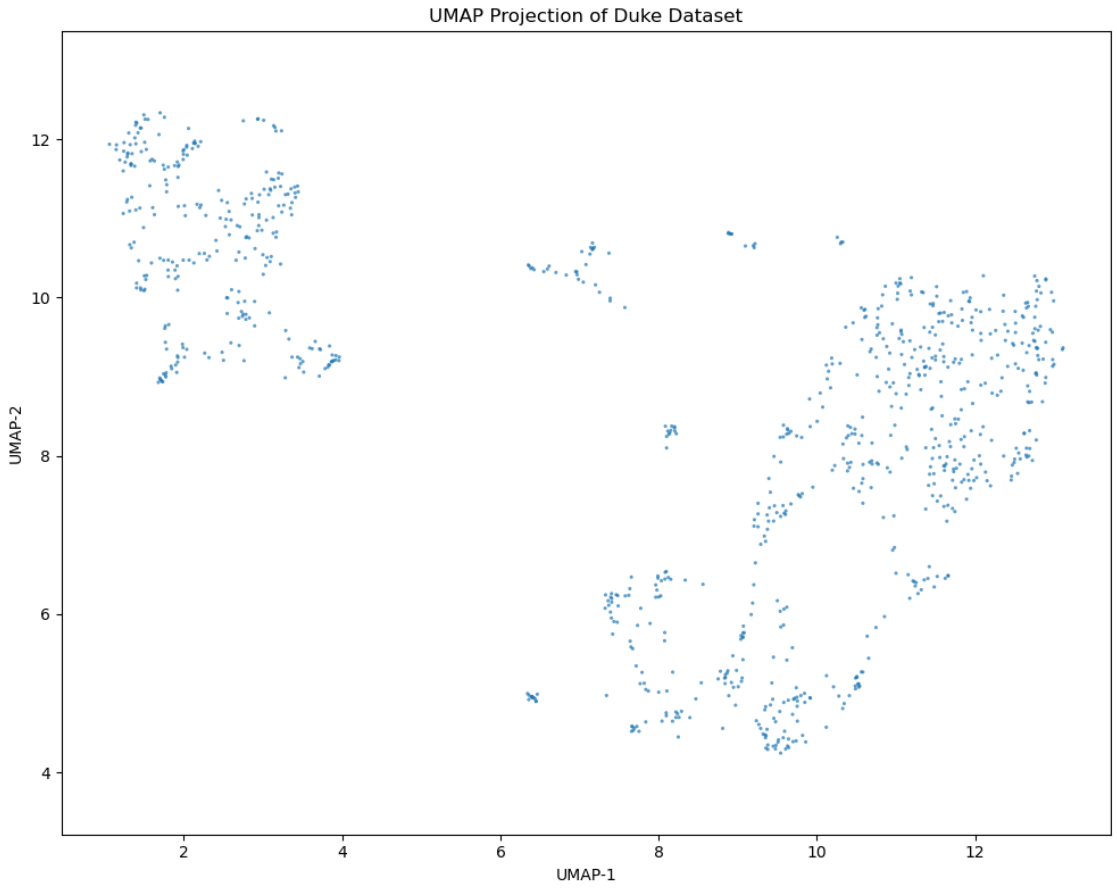
```
['Age', 'Race ', 'Marital Status', 'Unnamed: 3', 'T Stage ', 'N Stage', '6th
```

```
Stage', 'Grade', 'A Stage', 'Tumor Size', 'Estrogen Status', 'Progesterone
Status', 'Regional Node Examined', 'Reginol Node Positive', 'Survival Months',
'Status']
```



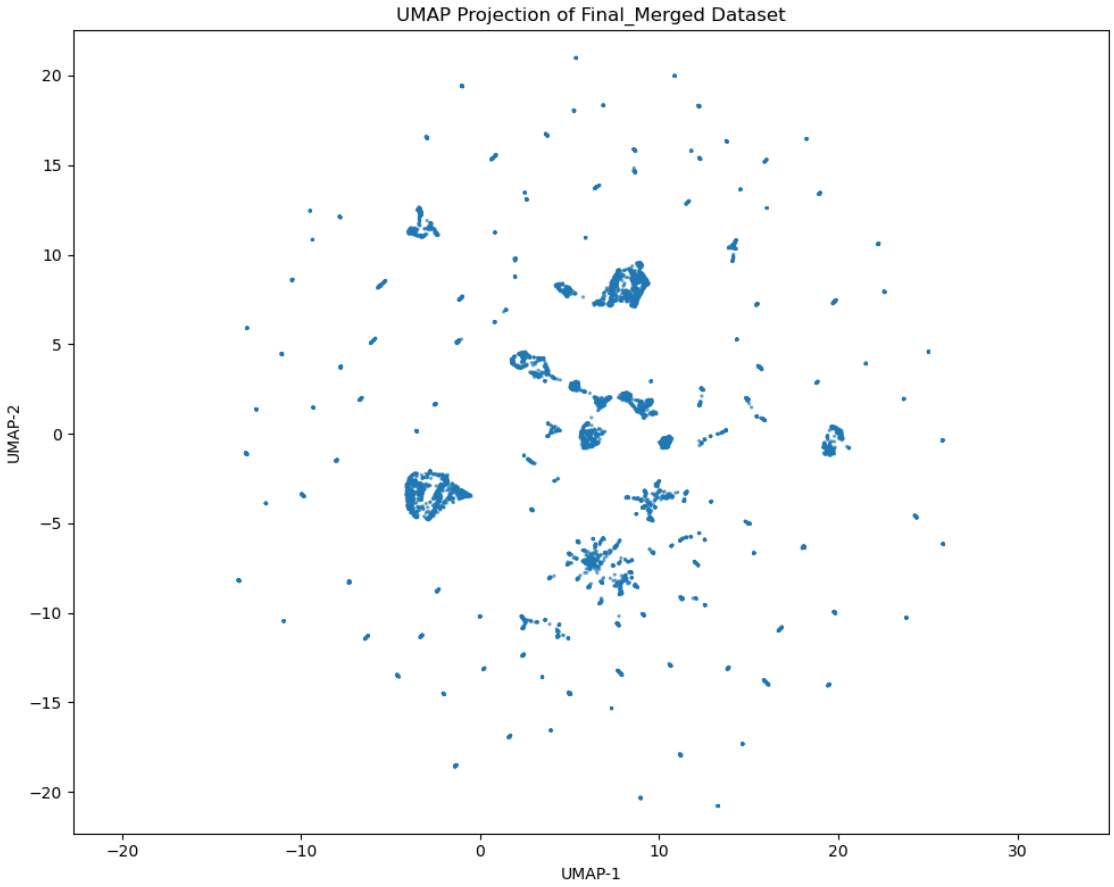
[22]:
```
df = pd.read_csv('Duke_Clinical_and_Other_Features.csv')
# print(df.columns.tolist())
X_numeric = df.select_dtypes(include=[np.number])
X_numeric = X_numeric.dropna(axis=1, how='all')

imputer = SimpleImputer(strategy='mean')
X_imputed = pd.DataFrame(
    imputer.fit_transform(X_numeric),
    index=X_numeric.index,
    columns=X_numeric.columns
)

scaler = StandardScaler()
X_scaled = scaler.fit_transform(X_imputed)
```

```
reducer = umap.UMAP(
    n_neighbors=15,
    min_dist=0.1,
    n_components=2,
    metric='euclidean',
    random_state=42,
    n_jobs=1
)
embedding = reducer.fit_transform(X_scaled)

plt.figure(figsize=(10, 8))
plt.scatter(
    embedding[:, 0], embedding[:, 1],
    s=5, alpha=0.7, edgecolors='none'
)
plt.gca().set_aspect('equal', 'datalim')
plt.title('UMAP Projection of Duke Dataset')
plt.xlabel('UMAP-1')
plt.ylabel('UMAP-2')
plt.tight_layout()
plt.show()
```



[21]:
```
df = pd.read_csv('final_merged.csv')
# print(df.columns.tolist())

X_numeric = df.select_dtypes(include=[np.number])
X_numeric = X_numeric.dropna(axis=1, how='all')

imputer = SimpleImputer(strategy='mean')
X_imputed = pd.DataFrame(
    imputer.fit_transform(X_numeric),
    index=X_numeric.index,
    columns=X_numeric.columns
)

scaler = StandardScaler()
X_scaled = scaler.fit_transform(X_imputed)

reducer = umap.UMAP(
    n_neighbors=15,
    min_dist=0.1,
    n_components=2,
    metric='euclidean',
    random_state=42,
    n_jobs=1
)
embedding = reducer.fit_transform(X_scaled)

plt.figure(figsize=(10, 8))
plt.scatter(
    embedding[:, 0], embedding[:, 1],
    s=5, alpha=0.7, edgecolors='none'
)
plt.gca().set_aspect('equal', 'datalim')
plt.title('UMAP Projection of Final_Merged Dataset')
plt.xlabel('UMAP-1')
plt.ylabel('UMAP-2')
plt.tight_layout()
plt.show()
```

```
C:\Users\86157\AppData\Local\Temp\ipykernel_3004\3688630918.py:1: DtypeWarning:
Columns (0,3,4,5,6,7,9,12,15,16,17,18,22,24,26,28,29,30,40,65,66,67,68,69,70,71,
72,73,74,75,76,82,83,89,90,91,92,93,96,97,98,99,100,101,102,103,104,105,106,107,
108,109,115) have mixed types. Specify dtype option on import or set
low_memory=False.
  df = pd.read_csv('final_merged.csv')
```


UMAP Projection of Final_Merged Dataset
UMAP-2
UMAP-1
20
15
10
5
0
−5
−10
−15
−20
−20
−10
0
10
20
30

[ ]: