Risk Stratification Algorithm Based on Breast Cancer Biopsy Images

PH 132 Final Project: **Nehal Sindhu, Ewen Sheng-Yao Huang, Vienna Huang, Bryan Sow, Tien Truong**

The Problem

Diagnostic rate of breast cancer is increasing due to more biopsies done, but the lack of accurate risk stratification (prognosis) tools lead to overtreatment of less aggressive cases and undertreatment of aggressive ones, yielding little improvement in case fatality rate. Our project aims to predict the risk of adverse outcome (metastasis or death) based on histopathology images from breast biopsies, which can help decide the subsequent patient treatment.

Goal of the Algorithm: More Accurate Prognosis (Risk stratification) for Breast Cancer based on Biopsy slides

The goal of our algorithm is to analyze digital pathology images from breast biopsies to predict which patients are at high risk of adverse outcomes, such as metastasis and death. By identifying patterns and features in the biopsy images that are indicative of aggressive cancer, the algorithm can help in distinguishing between cases requiring urgent treatment and those that do not (and need only followups), thus potentially reducing over-treatment and focusing resources on patients with more severe disease. This algorithm aims to improve the decision-making process regarding the urgency and intensity of treatment for patients with breast cancer. This includes decisions about surgery, chemotherapy, or the appropriateness of a watchful waiting approach, thus **optimizing patient care and resource allocation**.

The decision maker, in this case, would be the oncologists or pathologists that review the biopsy samples, or the medical technicians that do the biopsies, or any healthcare professional who evaluates the patient's biopsy slides and decides/advises on the course of treatment based on the predicted aggressiveness of the cancer.

We would like to know the risk of the cancer metastasizing or leading to death, based on the features observed in the biopsy images. This information would be used by oncologists to tailer the treatment stategy. Without the algorithm, the treatment team would rely on traditional histopathological assessments and staging criteria, which might not fully capture the complexity of cancer prognosis and could lead to less personalized, thus less proper, treatment decisions.

The problem can be summarized as "*If the treatment team knew the patients’ risk of adverse outcome based on the biopsy images, they could/would decide on the necessity and extent of active treatment, potentially improving the patient's prognosis and quality of care.*"

This case is **not a pure prediction problem**, since Y is the risk of metastasis or death, which is lowered by active treatment (X_0). There is a causal link from $X_0 \rightarrow Y$, since the treatment lowers the risk of metastasis and death.

In [1]:

```
import os
import random
import datetime, time
from tqdm import tqdm
import matplotlib.pyplot as plt
import seaborn as sns
import pandas as pd
import numpy as np
import pickle

import matplotlib.pyplot as plt
from PIL import Image

from concurrent.futures import ProcessPoolExecutor

from sklearn.model_selection import train_test_split
from sklearn.metrics import roc_curve, auc, precision_recall_curve, average_pre
from scipy.stats import sem

import torch
import torch.nn as nn
from torch.utils.data import Dataset, DataLoader
from torchvision import transforms as transforms
from torchvision.models import vgg19_bn, resnet50, densenet121, vit_b_16
import torch.optim as optim
from torch.utils.tensorboard import SummaryWriter
from sklearn.model_selection import train_test_split
from sklearn.metrics import classification_report, accuracy_score, roc_auc_sco
from sklearn.ensemble import RandomForestClassifier


log_dir = "~/logs"
writer = SummaryWriter(log_dir)
device = "cuda:0" if torch.cuda.is_available() else "cpu"
device = "cpu"
```

```
2023-12-15 08:32:23.059182: I tensorflow/core/platform/cpu_feature_guard.cc:18
2] This TensorFlow binary is optimized to use available CPU instructions in pe
rformance-critical operations.
To enable the following instructions: AVX2 AVX512F FMA, in other operations, r
ebuild TensorFlow with the appropriate compiler flags.
2023-12-15 08:32:23.936151: W tensorflow/compiler/tf2tensorrt/utils/py_utils.c
c:38] TF-TRT Warning: Could not find TensorRT
```

Inputs (X) used for prediction

The inputs (X variables used for prediction) are the pixel-level data from high-resolution digital pathology (IHC) images, features of the cancerous and non-cancerous tissue, the appearance of nuclei, and the rate of cell division, along with other features that the algorithm identifies as relevant.

The X is available at the time the decision X_0 is made, since X_0 is made after the biopsy images are fed into the algorithm. Given the complexity of cancer prognosis and the subtle features in the image that may predict outcomes, the function used to map $X \rightarrow Y$ needs to be sufficiently complex and expressive. The capability of deep learning models to identify patterns within large, high-resolution images makes them well-suited for this task.

Sample

Here, we loaded our datasets from Nightingale Open Source, and cleaned it so everything would be merged in a single dataset. We joined the datasets on the biopsy images provided.

The sample looks through 175,000 biopsy slides from 11,000 unique patients from cancer registry data from EHR data. This sample of biopsy slides looks for cancer stage, metastasis presence, and social security data based on mortality. The sample originates from the Providence Cancer Institute in Portland, Oregon, collected during the entire year of 2020 (January 1st to December 31st). Since the sample originates from Oregon, the data is not representative of breast cancer biopsy slides nationwide, which could affect algorithm performance if a different dataset of breast cancer biopsy slides were used to test the model.

Data and observations are collected from biopsy images collected from January 1st to December 31st, 2020, at the Providence Cancer Institute in Portland, Oregon. Each patient file includes multiple resolutions of images (from high to low), with the intent of allowing pathologists a better examination of the entire slide.

There would ideally be a true hold-out set to check the algorithm's performance after training and validation. It would be created by splitting the dataset by a determined threshold, where the images in the hold-out set would be inclusive of each category/classification of image type.

In [2]:

```
cancer_dx = pd.read_csv('/home/ngsci/datasets/brca-psj-path/v2/cancer-dx.csv')
comorb_dx = pd.read_csv('/home/ngsci/datasets/brca-psj-path/v2/comorbidities.c
demo_dx = pd.read_csv('/home/ngsci/datasets/brca-psj-path/v2/demographics.csv'
outcomes_dx = pd.read_csv('/home/ngsci/datasets/brca-psj-path/v2/outcomes.csv'
pathology_dx = pd.read_csv('/home/ngsci/datasets/brca-psj-path/v2/pathology-it
soc_det_dx = pd.read_csv('/home/ngsci/datasets/brca-psj-path/v2/social-determi
treatments_dx = pd.read_csv('/home/ngsci/datasets/brca-psj-path/v2/treatments.


cancer_dx
```

Out[2]:

	biopsy_id	dx_dt	icd9
0	00047e6d-cf9e-41f8-8901-eb9b0fe155a6	2118-01-20	174.9
1	00047e6d-cf9e-41f8-8901-eb9b0fe155a6	2121-04-26	174.9
2	00047e6d-cf9e-41f8-8901-eb9b0fe155a6	2118-05-26	174.9
3	00047e6d-cf9e-41f8-8901-eb9b0fe155a6	2118-07-17	174.9
4	00047e6d-cf9e-41f8-8901-eb9b0fe155a6	2117-05-24	174.9
...	...	...	...
75988	ffe94c67-18af-482a-afb8-90d75a4d640d	2116-07-30	174.4
75989	ffe94c67-18af-482a-afb8-90d75a4d640d	2116-07-08	174.9
75990	ffe94c67-18af-482a-afb8-90d75a4d640d	2116-10-21	174.9
75991	ffe94c67-18af-482a-afb8-90d75a4d640d	2118-01-31	174.4
75992	ffe94c67-18af-482a-afb8-90d75a4d640d	2118-09-22	174.4

75993 rows × 3 columns

In [3]:

```
all_data_merged = pd.merge(cancer_dx, comorb_dx, on='biopsy_id', how='outer', 
all_data_merged = pd.merge(all_data_merged, demo_dx, on='biopsy_id', how='oute
all_data_merged = pd.merge(all_data_merged, outcomes_dx, on='biopsy_id', how='
all_data_merged = pd.merge(all_data_merged, pathology_dx, on='biopsy_id', how=
all_data_merged = pd.merge(all_data_merged, soc_det_dx, on='biopsy_id', how='ou
all_data_merged = pd.merge(all_data_merged, treatments_dx, on='biopsy_id', how
```

In [4]:

```
all_data_merged
```

Out[4]:

	biopsy_id	dx_dt	icd9	dementia	peripheral_vascular_disease	pulmonary_disease
0	00047e6d-cf9e-41f8-8901-eb9b0fe155a6	2118-01-20	174.9	0	0	0
1	00047e6d-cf9e-41f8-8901-eb9b0fe155a6	2121-04-26	174.9	0	0	0
2	00047e6d-cf9e-41f8-8901-eb9b0fe155a6	2118-05-26	174.9	0	0	0
3	00047e6d-cf9e-41f8-8901-eb9b0fe155a6	2118-07-17	174.9	0	0	0
4	00047e6d-cf9e-41f8-8901-eb9b0fe155a6	2117-05-24	174.9	0	0	0
...	...	...	...	...	...	...
76538	fcc943af-1024-4aac-827c-ac195b34ff79	NaN	NaN	0	0	0
76539	fd0e6e3d-6515-4d1c-9e28-a7924834fb2a	NaN	NaN	0	0	0
76540	fdb35089-c272-48ca-aee3-c1b930c72aed	NaN	NaN	0	0	0
76541	fe86e332-ab10-4951-ac8b-61bad7e92801	NaN	NaN	0	0	0
76542	ff15bf31-9e58-42c4-ba05-de68f8dcf143	NaN	NaN	0	0	0

76543 rows × 58 columns

In [5]:

```
all_data_merged.columns
data = all_data_merged
```

In [6]:

```
print(data.head())

print(data.info())

print(data.describe())
```

```
                              biopsy_id       dx_dt   icd9  dementia  
0  00047e6d-cf9e-41f8-8901-eb9b0fe155a6  2118-01-20  174.9         0  \
1  00047e6d-cf9e-41f8-8901-eb9b0fe155a6  2121-04-26  174.9         0   
2  00047e6d-cf9e-41f8-8901-eb9b0fe155a6  2118-05-26  174.9         0   
3  00047e6d-cf9e-41f8-8901-eb9b0fe155a6  2118-07-17  174.9         0   
4  00047e6d-cf9e-41f8-8901-eb9b0fe155a6  2117-05-24  174.9         0   

   peripheral_vascular_disease  pulmonary_disease  liver_disease  diabetes  
0                            0                  0              0         0  \
1                            0                  0              0         0   
2                            0                  0              0         0   
3                            0                  0              0         0   
4                            0                  0              0         0   

   cerebral_vascular_accident  congestive_heart_failure  ...  chemo_summ_cd  
0                           0                         0  ...            3.0  \
1                           0                         0  ...            3.0   
2                           0                         0  ...            3.0   
3                           0                         0  ...            3.0   
4                           0                         0  ...            3.0   

   immuno_therapy_cd  hormone_summ_cd  rx_dx_stg_proc_dt  rx_mst_defn_srg_dt  
0                1.0              1.0         2117-03-15          2117-10-12  
\
1                1.0              1.0         2117-03-15          2117-10-12   
2                1.0              1.0         2117-03-15          2117-10-12   
3                1.0              1.0         2117-03-15          2117-10-12   
4                1.0              1.0         2117-03-15          2117-10-12   

   first_surgery_dt  radiation_start_dt  rx_chemo_dt  rx_hormone_dt  
0        2117-03-15          2118-02-12   2117-03-28     2118-04-15  \
1        2117-03-15          2118-02-12   2117-03-28     2118-04-15   
2        2117-03-15          2118-02-12   2117-03-28     2118-04-15   
3        2117-03-15          2118-02-12   2117-03-28     2118-04-15   
4        2117-03-15          2118-02-12   2117-03-28     2118-04-15   

   stg_dx_summ_cd  
0             2.0  
1             2.0  
2             2.0  
3             2.0  
4             2.0  

[5 rows x 58 columns]
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 76543 entries, 0 to 76542
Data columns (total 58 columns):
 #   Column                                 Non-Null Count  Dtype  
---  ------                                 --------------  -----  
 0   biopsy_id                              76543 non-null  object 
 1   dx_dt                                  75993 non-null  object 
 2   icd9                                   75993 non-null  float64
 3   dementia                               76543 non-null  int64  
 4   peripheral_vascular_disease            76543 non-null  int64  
 5   pulmonary_disease                      76543 non-null  int64  
 6   liver_disease                          76543 non-null  int64  
 7   diabetes                               76543 non-null  int64  
 8   cerebral_vascular_accident             76543 non-null  int64  
 9   congestive_heart_failure               76543 non-null  int64  
 10  diabetes_complications                 76543 non-null  int64  
 11  cancer                                 76543 non-null  int64  
 12  peptic_ulcer                           76543 non-null  int64  
 13  severe_liver_disease                   76543 non-null  int64  
 14  metastatic_cancer                      76543 non-null  int64  
 15  connective_tissue_disorder             76543 non-null  int64  
 16  acute_myocardial_infarction            76543 non-null  int64  
 17  renal_disease                          76543 non-null  int64  
 18  hiv                                    76543 non-null  int64  
 19  paraplegia                             76543 non-null  int64  
 20  sex                                    76543 non-null  object 
 21  race                                   76543 non-null  int64  
 22  ethnicity                              76543 non-null  int64  
 23  birth_dt                               75429 non-null  object 
 24  patient_ngsci_id                       76543 non-null  object 
 25  case_year                              76543 non-null  int64  
 26  biopsy_dt                              76543 non-null  object 
 27  mortality                              76543 non-null  int64  
 28  death_dt                               14775 non-null  object 
 29  in_registry                            76543 non-null  int64  
 30  stage                                  70613 non-null  object 
 31  strict_metastatic_dx                   76543 non-null  int64  
 32  strict_metastatic_dx_dt                24760 non-null  object 
 33  grade_clinical                         22794 non-null  object 
 34  grade_pathological                     23155 non-null  object 
 35  er_summary                             70995 non-null  float64
 36  pr_summary                             70504 non-null  float64
 37  her2_summary                           69977 non-null  float64
 38  multigene_signature_method             23599 non-null  float64
 39  multigene_signature_result             23599 non-null  object 
 40  response_neoadjuv_therapy              23375 non-null  float64
 41  bmi                                    52236 non-null  float64
 42  tobacco                                76543 non-null  int64  
 43  cancer_registry_dx_dt                  71311 non-null  object 
 44  most_definitive_surgical_procedure_cd  69937 non-null  float64
 45  most_definitive_radiation_modality_cd  71311 non-null  float64
 46  surgical_margin_cd                     70850 non-null  float64
 47  radiation_summ_cd                      57603 non-null  float64
 48  chemo_summ_cd                          71311 non-null  float64
 49  immuno_therapy_cd                      71311 non-null  float64
 50  hormone_summ_cd                        71311 non-null  float64
 51  rx_dx_stg_proc_dt                      69631 non-null  object 
 52  rx_mst_defn_srg_dt                     63782 non-null  object 
 53  first_surgery_dt                       68172 non-null  object 
 54  radiation_start_dt                     44928 non-null  object 
 55  rx_chemo_dt                            38926 non-null  object 
 56  rx_hormone_dt                          54567 non-null  object 
 57  stg_dx_summ_cd                         71311 non-null  float64
dtypes: float64(15), int64(24), object(19)
memory usage: 33.9+ MB
None
               icd9      dementia  peripheral_vascular_disease
count  75993.000000  76543.000000                 76543.000000  \
mean     176.881259      0.020342                     0.081131
std        6.839141      0.141167                     0.273038
min      174.000000      0.000000                     0.000000
25%      174.400000      0.000000                     0.000000
50%      174.900000      0.000000                     0.000000
75%      174.900000      0.000000                     0.000000
max      198.890000      1.000000                     1.000000
```

```
       pulmonary_disease  liver_disease      diabetes  
count       76543.000000   76543.000000  76543.000000  \
mean            0.253714       0.014841      0.151588   
std             0.435138       0.120918      0.358623   
min             0.000000       0.000000      0.000000   
25%             0.000000       0.000000      0.000000   
50%             0.000000       0.000000      0.000000   
75%             1.000000       0.000000      0.000000   
max             1.000000       1.000000      1.000000   

       cerebral_vascular_accident  congestive_heart_failure  
count                76543.000000              76543.000000  \
mean                     0.094404                  0.074141   
std                      0.292392                  0.262002   
min                      0.000000                  0.000000   
25%                      0.000000                  0.000000   
50%                      0.000000                  0.000000   
75%                      0.000000                  0.000000   
max                      1.000000                  1.000000   

       diabetes_complications        cancer  ...          bmi       tobacco  
count            76543.000000  76543.000000  ...  52236.00000  76543.000000  \
mean                 0.055250      0.993429  ...     29.28176      0.087532   
std                  0.228469      0.080798  ...      7.60801      0.282616   
min                  0.000000      0.000000  ...     15.00000      0.000000   
25%                  0.000000      1.000000  ...     23.00000      0.000000   
50%                  0.000000      1.000000  ...     28.00000      0.000000   
75%                  0.000000      1.000000  ...     33.00000      0.000000   
max                  1.000000      1.000000  ...     64.00000      1.000000   

       most_definitive_surgical_procedure_cd  
count                           69937.000000  \
mean                               28.581695   
std                                15.500331   
min                                 0.000000   
25%                                22.000000   
50%                                22.000000   
75%                                42.000000   
max                                99.000000   

       most_definitive_radiation_modality_cd  surgical_margin_cd  
count                           71311.000000        70850.000000  \
mean                               12.842970            0.914834   
std                                14.584544            2.486821   
min                                 0.000000            0.000000   
25%                                 0.000000            0.000000   
50%                                 0.000000            0.000000   
75%                                31.000000            0.000000   
max                                99.000000            9.000000   

       radiation_summ_cd  chemo_summ_cd  immuno_therapy_cd  hormone_summ_cd  
count       57603.000000   71311.000000       71311.000000     71311.000000  \
mean            0.757322       5.362567           0.612542         5.192383   
std             1.266443      17.520206           5.821592        18.831175   
min             0.000000       0.000000           0.000000         0.000000   
25%             0.000000       0.000000           0.000000         1.000000   
50%             1.000000       2.000000           0.000000         1.000000   
75%             1.000000       3.000000           0.000000         1.000000   
max             9.000000      99.000000          88.000000        99.000000   
```

```
       stg_dx_summ_cd
count    71311.000000
mean         1.951368
std          0.305591
min          0.000000
25%          2.000000
50%          2.000000
75%          2.000000
max          2.000000

[8 rows x 39 columns]
```

Label (Y) being predicted

The literal, measured variable Y is the risk of metastasis or death. The **source of truth of Y is the diagnosis of metastsis and registry for death that comes from Providence's cancer registry and is supplemented by ICD codes in the diagnosis tables from the Epic electronic medical record (EMR) system and the Social Security Death Index**.

The underlying true target $Y*$ will be whether or not the patient will experience preventable adverse outcome **if we don’t treat them**, since this counterfactual label can directly affect the decision of treatment. In this case, ,$Y*$ is not entirelymeasurable, creating a gap (Δ) between the true and measured targets.

$$Y = Y* + \Delta$$

The source of Δ mainly comes from the following two parts:

1. Treatment pollution: For the actively treated popullation, their risks of adverse outcomes were lowered, and we cannot know their risk "if they were not treated".
2. Even if we only look at the untreated population, we only have data from certain hospitals with limited followup time (only 2020), thus the occurence of adverse outcomes were not entirely recorded (if they took place in another hospital or after 2020).

Building our Model

We start by standardizing the features in our dataset, and then plotting the distribution of cancer stages.

In [7]:

```python
from sklearn.preprocessing import StandardScaler
scaler = StandardScaler()
data_scaled = scaler.fit_transform(data.select_dtypes(include=np.number))
```

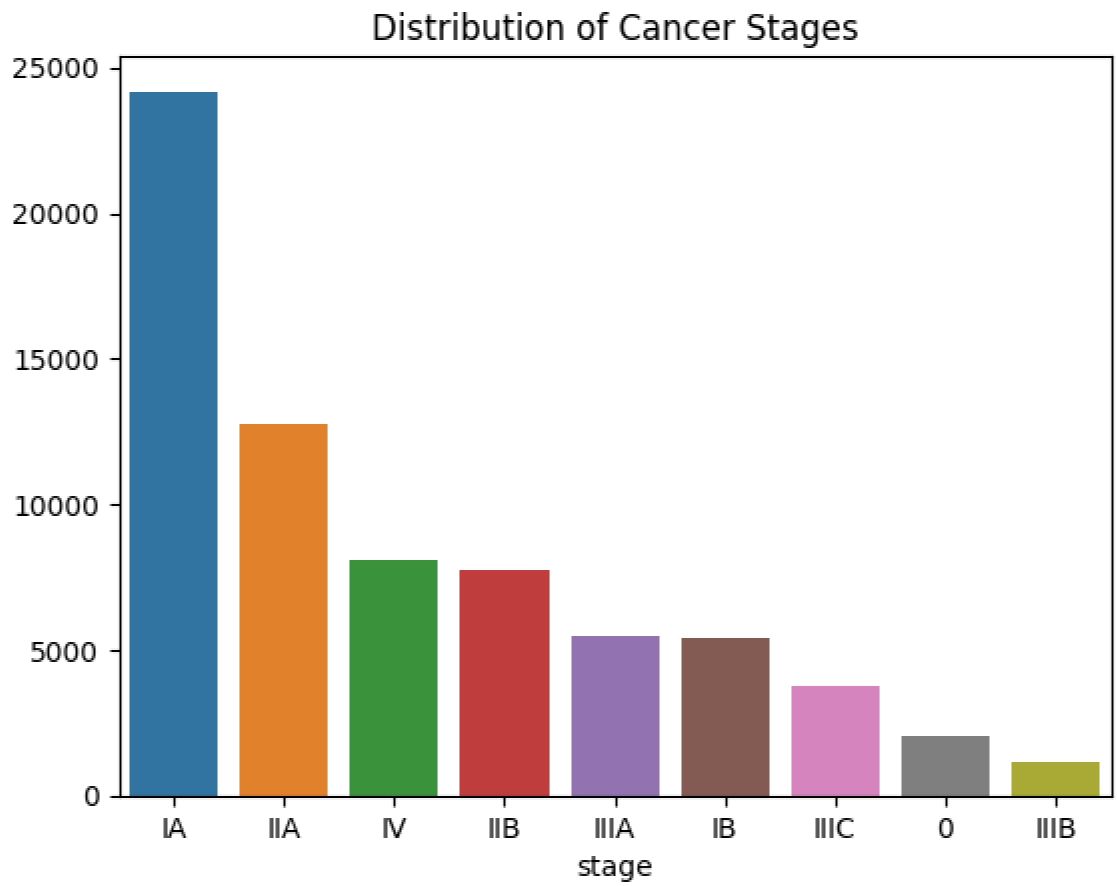
In [8]:

```python
print(f"Sample size: {len(data)}")
```

```
Sample size: 76543
```

In [9]:

```python
sns.barplot(x=data['stage'].value_counts().index, y=data['stage'].value_counts
plt.title('Distribution of Cancer Stages')
plt.show()
```



We did one-hot encoding for categorical data, and trained it on features that accurately predict based on y value: 'strict_metastatic_dx' which, according to the dataset descritpion is "strict" in the sense that it requires a breast cancer diagnosis to be present on the same day as a metastatic diagnosis.

We chose Random Forest Classifier because we know the Random Forest can handle big data with numerous variables, as is the case with this dataset.

In [10]:

```
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.preprocessing import LabelEncoder, OneHotEncoder

data_for_model = data.copy()

categorical_columns = data_for_model.select_dtypes(include=['object']).columns

data_for_model = pd.get_dummies(data_for_model, columns=categorical_columns, d

if 'biopsy_id' in data_for_model.columns:
    data_for_model = data_for_model.drop(['biopsy_id'], axis=1)

y = data_for_model['strict_metastatic_dx']
X = data_for_model.drop(['strict_metastatic_dx'], axis=1)

from sklearn.impute import SimpleImputer

num_imputer = SimpleImputer(strategy='mean')
for column in X.select_dtypes(include=np.number).columns:
    X[column] = num_imputer.fit_transform(X[[column]])
```

```
cat_imputer = SimpleImputer(strategy='most_frequent')
for column in X.select_dtypes(include=['object', 'category']).columns:
    X[column] = cat_imputer.fit_transform(X[[column]])

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, rando

rf_classifier = RandomForestClassifier(random_state=42)

rf_classifier.fit(X_train, y_train)
```

Out[10]:

```
▾        RandomForestClassifier
RandomForestClassifier(random_state=42)
```

Success Metric that the algorithm is judged

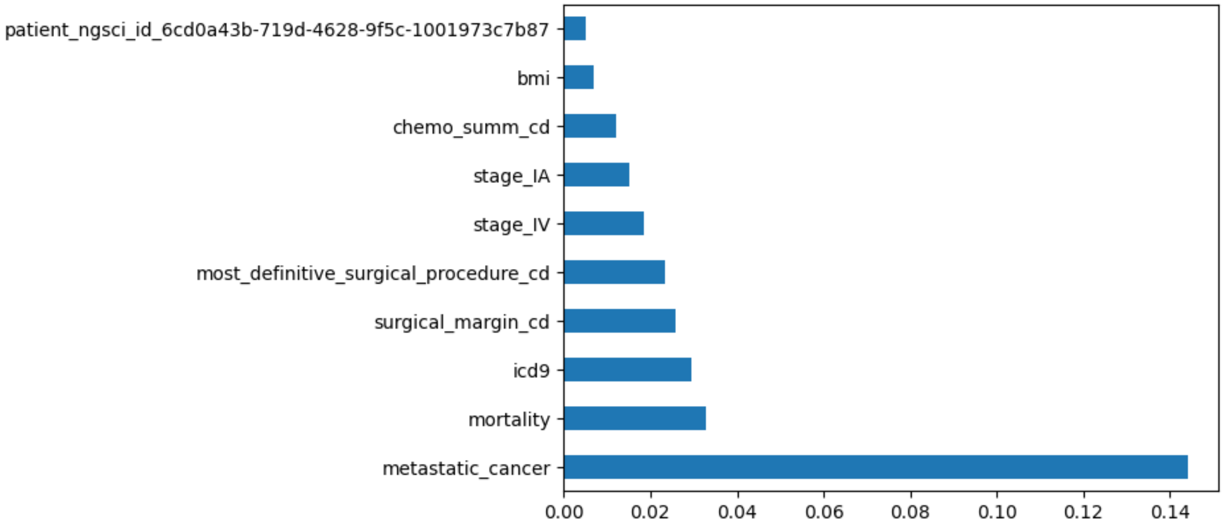
The success metric is determined by the accuracy of the algorithm in predicting the correct outcome (whether a patient will get breast cancer) based on the image presented.

Feature Importance, Cross-Validation, and Accuracy Score

Here, we checked feature importance to see which features contributed to the model's final prediction, as well as the cross-validation and average accuracy score of our model.

In [11]:

```
feature_importances = pd.Series(rf_classifier.feature_importances_, index=X.co
feature_importances.nlargest(10).plot(kind='barh')
plt.show()
```



In [12]:

```
from sklearn.model_selection import cross_val_score

cv_scores = cross_val_score(rf_classifier, X, y, cv=5)
print(f"Cross-validation scores: {cv_scores}")
print(f"Average score: {cv_scores.mean()}")
```

```
Cross-validation scores: [0.74949376 0.83689333 0.82056307 0.79481317 0.868892
08]
Average score: 0.8141310824384631
```

Our model has an accuracy of about 80%. Important features included are 'metastatic_cancer', 'mortality', 'icd9,' which suggests that there is a direct correlation between these features and the model's prediction.

Pitfalls related to hidden potential outcomes

Based on the information provided from the Nightingale Open Science dataset, observations (biopsies) get labeled by linking them to clinical outcomes such as metastasis and mortality. These outcomes are identified using strict criteria based on ICD codes recorded in the patient's medical records. This method is described as 'strict' because it requires a diagnosis of breast cancer to be present on the same day as a metastasis diagnosis.

Selective labeling is not a major problem in our dataset, since we have the labels (Y) of both $T = 0$ and $T = 1$. Namely, we have the data on the outcomes of the patients that are actively treated and not actively treated.

However there might be some other types of missing data problems (the **“missing rows”**), such as the possibility of the dataset not capturing all cases of breast cancer due to selection criteria or if it disproportionately represents certain demographics or stages of cancer. This could lead to biases in the algorithm, affecting its ability to generalize to the broader population of breast cancer patients. The dataset aims to minimize by including a broad range of biopsies and patient outcomes from an extensive time frame (2010 to 2020).

Value of the algorithm

An accurate prediction algorithm can lead to better personalized treatment plans, potentially improving patient survival rates and quality of life. Most importantly, it can alleviate the overtreatment problem arising from the advancement of screening technology that is becoming more accessible to all patients.

By potentially reducing unnecessary invasive procedures and enabling targeted treatments, the algorithm could improve the efficiency of the healthcare system.

Likely payers for the algorithm could be healthcare providers, insurers, or government health agencies, especially if the algorithm can demonstrate cost-saving benefits by improving treatment efficacy. The cost would reflect its development, effectiveness, and the economic benefits it provides.