Module_3: Investigating Cancer

Team Members:

Maggie and Aarnav

Project Title:

Angiogenesis Signalling and Pathways

Project Goal:

This project seeks to investigate how cancer stage is related to angiogenesis signalling proteins like VEGFA, FGF1, and THBS1

Disease Background:

- Cancer hallmark focus: **Sustained Angiogenesis**
- Overview of hallmark: For tumors to grow, they must gain angiogenetic capabilities in order for the cells to be supplied with the nutrients to live and proliferate. Tumors are thought to gain this ability in discrete steps. Through steps which aren't understood, but believed to involve altered gene transcription, where tumors shift the balance by either increasing expression for growth factors, or diminish expression for inhibitors.
- Genes associated with hallmark to be studied (describe the role of each gene, signaling pathway, or gene set you are going to investigate):
    - VEGFR1 (Flt-1): Vascular endothelial growth factor receptor 1 is an important gene for angiogenesis. It's activated by VEGF-A or VEGF-B, and promots the function of macrophages, which can promote tumor growth.
    - FGF1: This gene encodes fibroblast growth factor 1, which is a growth factor that promotes proliferation, angiogenesis and healing. Its disregulation is linked to cancer. It acts as a ligand, binding to receptors which trigger signals causing growth.

  - THBS1: This gene encodes thrombospondin-1, typically a growth inhibitor. However, in some studies, it's been shown to promote cancer growth.

We will be looking across all cancer types to see how levels of certain signalling proteins change depending on the stage of cancer. If time allows, we will compare between the cancer types and see if levels of signalling proteins according to cancer stage change across types of cancer.

- Prevalence & incidence: The rate of new cases of cancer (cancer incidence) is 445.8 per 100,000 men and women per year (based on 2018–2022 cases). Approximately 38.9% of men and women will be diagnosed with cancer at some point during their lifetimes (based on 2018–2021 data, not including 2020, due to COVID). https://www.cancer.gov/about-cancer/understanding/statistics. The prevalance of the three most common cancer types in 2025 (breast, prostate, and lung) were 319,750 ; 313,780 ; and 226,650 respectively. The most deadly cancer is lung cancer, which killed 124,730 people in 2025. https://seer.cancer.gov/statfacts/html/common.html

- Risk factors (genetic, lifestyle) & Societal determinants: There are both lifestyle and genetic risk factors associated with all types of cancer. Commonly cited lifestyle risk factors include smoking (specifically for lung cancer), alcohol consumption, exposure to sun(skin cancer), physical inactivity, poor diet/obesity, and environmental pollutants. A family history of cancer is also considered a risk factor. There are known heritable genetic mutations such as BRCA1 and BRCA2. Additionally, certain viruses such as Human papillomavirus (HPV), Hepatitis B and C viruses, and Epstein-Barr virus are cancer risk factors. Risk increases with age and certain hormonal factors, as well as exposure to radiation. https://www.cancer.gov/about-cancer/causes-prevention/risk#:~:text=Cancer%20risk%20factors%20include%20exposure%20to%20chemicals,sign%20of%20a%20possible%20inherited%2 and Google AI. Important socioeconomic factors that can be barriers to proper screening and treatement are income, education, and employment. Access to clean air and water affects overall health, and thus cancer development. A strong social support network is linked to overall health as well, and has been shown to result in higher survival rates in cancer patients. https://pmc.ncbi.nlm.nih.gov/articles/PMC8494398/

- Standard of care treatments (& reimbursement): Typical cancer treatments are surgery, radiation therapy, chemotherapy, hormonal therapy, immunotherapy, and stem cell transplants, depending on the type and stage of cancer. https://www.cancer.gov/about-cancer/treatment/types#:~:text=Types%20of%20Cancer%20Treatment%20in,Contact%20Us Cancer standard of care reimbursement is complex, involving public and private insurance coverage, alternative payment models, and financial assistance programs. from Google AI. Most plans cover standard cancer care, but costs and coverage specifics vary by plan. Patients should review their benefits and work with their providers to understand potential out-of-pocket expenses https://www.facingourrisk.org/support/insurance-paying-for-care/paying-for-cancer-treatment

- Biological mechanisms (anatomy, organ physiology, cell & molecular physiology). Based on the "hallmarks of cancer" (Hanahan, Douglas et al. Cell, Volume 144, Issue 5, 646 - 674), the biological mechanisms of cancer cells are complicated. Cancer cells grow by evading growth supressors, avoiding immune destruction, enabling replicative immortality, tumor promoting inflammation, activating invasion + metastasis, inducing angiogenesis, genome instability, resisting cell death, deregulating cellular energetics, and sustaining proliferative signaling. In short, cancer cells have mutated to bypass the safeguards placed on normal cells. They stimulate their own growth and ignore signals to stop growing and/or die. The biological mechanism we gocus on is inducing sustained angiogenesis. Cancer cells develop the ability to stimulate the formation of new blood vessels from existing ones to ensure they have plenty of nutrients to grow out of proportion. Hypoxia: Low oxygen levels are a major trigger. Cells in hypoxic tissues release chemical signals, including hypoxia-inducing factors (HIFs), that lead to the release of pro-angiogenic factors to create new blood vessels. Inflammation: Inflammatory responses activate cells to release signaling molecules that encourage new blood vessel growth. Chemokines like IL-8 (CXCL8) and CCL2 are examples that promote this process. Growth Factors: Vascular Endothelial Growth Factor (VEGF): This is a dominant pro-angiogenic factor, especially in tumors. It binds to receptors on endothelial cells, causing them to proliferate and form new vessels. Fibroblast Growth Factors (FGFs): Another group of growth factors involved in stimulating blood vessel formation. From Google AI

Data-Set:

To answer our question about how cancer stage is correlated with the expression of angionesis growth factors and inhibitors, we will use the following data:

- Metadata: ajcc_pathologic_tumor_stage and the key
    - As we might expand our investigation to different types of cancer, we will also use that column of data
- Genetic data: VEGFA, FGF1, THBS1
    - The units used for the data is log2TPM, which is the logarithm of TPM (Transcripts per million) to the base 2. This unit normalises for the length of genes, and takes a per million count from that. Its logarithm is then taken.

The database being used is processed and sequenced from The Cancer Genome Atlas. The total dataset includes 9000+ tumor samples across 24 types of cancer and 700+ normal samples. This was released in a paper titled 'Alternatively processed and compiled RNA-Sequencing and clinical data for thousands of samples from The Cancer Genome Atlas' in 2015.

https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE62944

Rahman M, Jackson LK, Johnson WE, Li DY et al. Alternative preprocessing of RNA-Sequencing data in The Cancer Genome Atlas leads to improved analysis results. Bioinformatics 2015 Nov 15;31(22):3666-72. PMID: 26209429

Data Analyis:

Methods

The machine learning technique we are using is a decision tree. A decision tree is a supervised learning algorithm that models decisions leading to a certain outcome, which can be used to predict categories. The process begins at the top of the tree, where all the data is together. As the algorithm progesses, it identifies a question to ask of the data to most cleanly separate it into two groups. This process continues as the tree grows and more groups are formed, until the specified "depth" is reached.

What is this method optimizing? How does the model decide it is "good enough"?

Our decision tree classifier uses gini impurity to measure the likelihood of incorrectly classifying a randomly chosen element in a dataset if it were randomly labelled. Essentailly, a lower gini impurity indicates more purity, or better likelihood that the classification is correct, and is desirable. By minimising this, the model is trying to reduce the 'entropy' of the dataset.

However, with a limited sample size, this can easily lead to overfitting if the depth of the binary tree isn't limited.

Analysis

In [2]:

```python
import pandas as pd

# Step 1: Read the CSV file for the metadata and the sample data
metadata = pd.read_csv('GSE62944_metadata.csv')
sample_data = pd.read_csv('GSE62944_subsample_topVar_log2TPM.csv')
```

In [3]:

```python
# Step 2: Process the data to filter out irrelevant data for our investigation

# Only keep the sample, cancer type, and ajcc_pathologic_tumor_stage columns from the metadata
metadata_filtered = metadata[['sample', 'cancer_type', 'ajcc_pathologic_tumor_stage']]

# Transpose the sample data so that samples are rows and genes are columns
```

```
sample_data = sample_data.transpose()
# Fixing the column names
column_names = sample_data.iloc[0]
sample_data = sample_data[1:]
sample_data.columns = column_names
sample_data.reset_index(inplace=True)
sample_data.rename(columns={'index': 'sample'}, inplace=True)

# Filter out everything but the sample, VEGFA, FGF1, and THBS1 columns
sample_data_filtered = sample_data[['sample', 'VEGFA', 'FGF1', 'THBS1']]

print(metadata_filtered.head())
print(sample_data_filtered.head())
```

```
                         sample cancer_type ajcc_pathologic_tumor_stage
0  TCGA-E9-A1NI-01A-11R-A14D-07        BRCA                   Stage IIA
1  TCGA-E2-A1LK-01A-21R-A14D-07        BRCA                  Stage IIIC
2  TCGA-BH-A0B2-01A-11R-A10J-07        BRCA                     Stage I
3  TCGA-E2-A107-01A-11R-A10J-07        BRCA                  Stage IIIA
4  TCGA-LL-A5YN-01A-11R-A28M-07        BRCA                   Stage IIA
Unnamed: 0                        sample     VEGFA      FGF1     THBS1
0           TCGA-E9-A1NI-01A-11R-A14D-07  6.169153  2.806279  7.924146
1           TCGA-E2-A1LK-01A-21R-A14D-07   6.86095  1.401355  6.520983
2           TCGA-BH-A0B2-01A-11R-A10J-07   5.44373  3.227564  8.343626
3           TCGA-E2-A107-01A-11R-A10J-07  5.012467  2.047964   5.84812
4           TCGA-LL-A5YN-01A-11R-A28M-07  4.160556  2.315567  6.282073
```

In [4]:

```
# Step 3: Merge the metadata and sample data into a single DataFrame
merged_data = pd.merge(metadata_filtered, sample_data_filtered, on='sample')

merged_data
```

Out[4]:

	sample	cancer_type	ajcc_pathologic_tumor_stage	VEGFA	FGF1	THBS1
0	TCGA-E9-A1NI-01A-11R-A14D-07	BRCA	Stage IIA	6.169153	2.806279	7.924146
1	TCGA-E2-A1LK-01A-21R-A14D-07	BRCA	Stage IIIC	6.86095	1.401355	6.520983
2	TCGA-BH-A0B2-01A-11R-A10J-07	BRCA	Stage I	5.44373	3.227564	8.343626
3	TCGA-E2-A107-01A-11R-A10J-07	BRCA	Stage IIIA	5.012467	2.047964	5.84812
4	TCGA-LL-A5YN-01A-11R-A28M-07	BRCA	Stage IIA	4.160556	2.315567	6.282073
...	...	...	...	...	...	...
1797	TCGA-N5-A4RO-01A-11R-A28V-07	UCS	NaN	7.726898	0.284816	5.042546
1798	TCGA-N5-A4RV-01A-21R-A28V-07	UCS	NaN	6.890211	0.37829	3.775268
1799	TCGA-N6-A4VD-01A-11R-A28V-07	UCS	NaN	6.707867	0.880173	5.269738
1800	TCGA-N5-A4RT-01A-11R-A28V-07	UCS	NaN	4.810951	0.078323	5.115454
1801	TCGA-ND-A4WC-01A-21R-A28V-07	UCS	NaN	8.022486	1.007999	5.98153

1802 rows × 6 columns

In [5]:

```
# Drop values which don't have a tumor stage assigned or are labelled not available
merged_data = merged_data[merged_data['ajcc_pathologic_tumor_stage'].notna()]
merged_data = merged_data[merged_data['ajcc_pathologic_tumor_stage'] != '[Not Available]']

# Exporting to a csv
merged_data.to_csv('processed_data.csv', index=False)
merged_data
```

Out[5]:

	sample	cancer_type	ajcc_pathologic_tumor_stage	VEGFA	FGF1	THBS1
0	TCGA-E9-A1NI-01A-11R-A14D-07	BRCA	Stage IIA	6.169153	2.806279	7.924146
1	TCGA-E2-A1LK-01A-21R-A14D-07	BRCA	Stage IIIC	6.86095	1.401355	6.520983
2	TCGA-BH-A0B2-01A-11R-A10J-07	BRCA	Stage I	5.44373	3.227564	8.343626
3	TCGA-E2-A107-01A-11R-A10J-07	BRCA	Stage IIIA	5.012467	2.047964	5.84812
4	TCGA-LL-A5YN-01A-11R-A28M-07	BRCA	Stage IIA	4.160556	2.315567	6.282073
...	...	...	...	...	...	...
1740	TCGA-KL-8344-01A-11R-2315-07	KICH	Stage III	7.84551	3.271281	5.980591
1741	TCGA-KM-8438-01A-11R-2315-07	KICH	Stage II	7.829908	0.447008	4.215869
1742	TCGA-KO-8414-01A-11R-2315-07	KICH	Stage II	7.642517	2.092672	3.364362
1743	TCGA-KL-8333-01A-11R-2315-07	KICH	Stage II	8.144321	1.883174	8.785348
1744	TCGA-KO-8411-01A-11R-2315-07	KICH	Stage I	7.275827	1.665704	4.783421

1103 rows × 6 columns

Decision Tree

Because we were trying to predict categorical data (Stage of cancer), we decided to use a decision tree. Classification wouldn't work becuase it can only distinguish between two categories.

The features were the 3 genes: VEGFA, FGF1 and THBS1.

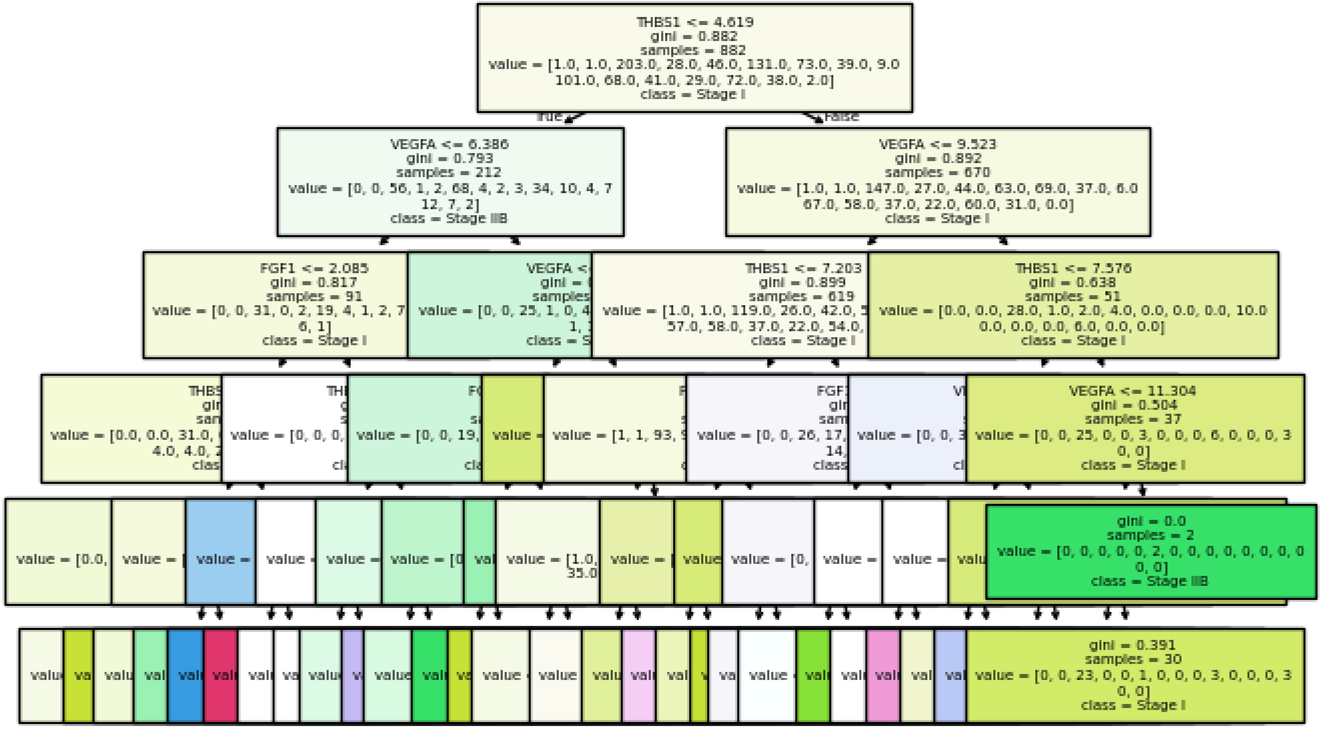
We divided our 1102 data points into training and testing sections with an 80:20 split, which we thought would maximise training size while still giving a fair amount of testing data.

Limiting the max depth of the tree to 5 layers mitigated overfitting, while allowing for enough outcomes to cover all the stages.

In [6]:

```
from sklearn.tree import DecisionTreeClassifier, plot_tree
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder, label_binarize
from sklearn.metrics import roc_curve, auc
from sklearn.multiclass import OneVsRestClassifier
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
# merged_data = pd.read_csv('C:/Users/MagPi/comp bme mod 3/bme2315_module_3/processed_data.csv') # this line is here to make i


# Define features and labels
features = merged_data[['VEGFA', 'FGF1', 'THBS1']]
labels = merged_data['ajcc_pathologic_tumor_stage']

# Train test split
X_train, X_test, y_train, y_test = train_test_split(features, labels, test_size=0.2, random_state=42)

# train decision tree
model = DecisionTreeClassifier(max_depth=5)
model.fit(X_train, y_train)

# Make predictions
predictions = model.predict(X_test)
print(model.feature_importances_)

# plot decision tree
tree = model.tree_
plot_tree(model, feature_names=features.columns.tolist(), class_names=labels.tolist(), filled=True, fontsize=5)
plt.show()
```

```
[0.42932848 0.26557142 0.30510009]
```



In [7]:

```
from sklearn.metrics import accuracy_score

# Make predictions
predictions = model.predict(X_test)

# Evaluate the model
accuracy = accuracy_score(y_test, predictions)
print("Model accuracy:", accuracy)

print("the chance of a random guess being correct is:", 1/len(labels.unique()))
```

```
Model accuracy: 0.248868778280543
the chance of a random guess being correct is: 0.0625
```

Accuracy

Our model accuracy was ~24%, which means 24% of predictions were correct. Since there were 16 categories, randomly guessing would lead to a ~6% accuracy rate on average, so our model is definitely an improvement from the baseline, although not very accurate.

In [9]:

```
# Testing positive and negative tendencies for each independent variable
# Fixes two of the variables to see how the third one affects the outcome across a range of values

vegf_a_median = features['VEGFA'].median()
fgf_1_median = features['FGF1'].median()
thbs1_median = features['THBS1'].median()

vegf_a_range = np.linspace(features['VEGFA'].min(), features['VEGFA'].max(), 10)
fgf_1_range = np.linspace(features['FGF1'].min(), features['FGF1'].max(), 10)
thbs1_range = np.linspace(features['THBS1'].min(), features['THBS1'].max(), 10)

for value in vegf_a_range:
    test_sample = pd.DataFrame({
        'VEGFA': [value],
        'FGF1': [fgf_1_median],
        'THBS1': [thbs1_median]
    })
    prediction = model.predict(test_sample)
    print(f'VEGFA: {value}, Predicted Stage: {prediction[0]}')

for value in fgf_1_range:
    test_sample = pd.DataFrame({
        'VEGFA': [vegf_a_median],
        'FGF1': [value],
        'THBS1': [thbs1_median]
    })
    prediction = model.predict(test_sample)
    print(f'FGF1: {value}, Predicted Stage: {prediction[0]}')

for value in thbs1_range:
    test_sample = pd.DataFrame({
        'VEGFA': [vegf_a_median],
        'FGF1': [fgf_1_median],
        'THBS1': [value]
    })
    prediction = model.predict(test_sample)
    print(f'THBS1: {value}, Predicted Stage: {prediction[0]}')
```

```
VEGFA: 2.563197612471409, Predicted Stage: Stage I
VEGFA: 3.586346074383834, Predicted Stage: Stage I
VEGFA: 4.609494536296259, Predicted Stage: Stage I
VEGFA: 5.632642998208684, Predicted Stage: Stage I
VEGFA: 6.655791460121108, Predicted Stage: Stage I
VEGFA: 7.678939922033534, Predicted Stage: Stage I
VEGFA: 8.70208838394596, Predicted Stage: Stage I
VEGFA: 9.725236845858383, Predicted Stage: Stage IB
VEGFA: 10.748385307770809, Predicted Stage: Stage IV
VEGFA: 11.771533769683234, Predicted Stage: Stage IV
FGF1: 0.0, Predicted Stage: Stage I
FGF1: 0.8181785666930721, Predicted Stage: Stage I
FGF1: 1.6363571333861442, Predicted Stage: Stage I
FGF1: 2.4545357000792163, Predicted Stage: Stage I
FGF1: 3.2727142667722884, Predicted Stage: Stage I
FGF1: 4.090892833465361, Predicted Stage: Stage I
FGF1: 4.909071400158433, Predicted Stage: Stage IIIC
FGF1: 5.727249966851504, Predicted Stage: Stage IIIC
FGF1: 6.545428533544577, Predicted Stage: Stage IIIC
FGF1: 7.363607100237649, Predicted Stage: Stage IIIC
THBS1: 0.6064032330508123, Predicted Stage: Stage II
THBS1: 1.804203263977337, Predicted Stage: Stage II
THBS1: 3.002003294903862, Predicted Stage: Stage II
THBS1: 4.199803325830387, Predicted Stage: Stage II
THBS1: 5.397603356756912, Predicted Stage: Stage I
THBS1: 6.595403387683437, Predicted Stage: Stage I
THBS1: 7.793203418609962, Predicted Stage: Stage IIIA
THBS1: 8.991003449536487, Predicted Stage: Stage IIIA
THBS1: 10.188803480463012, Predicted Stage: Stage IIIA
THBS1: 11.386603511389536, Predicted Stage: Stage IIIA
```

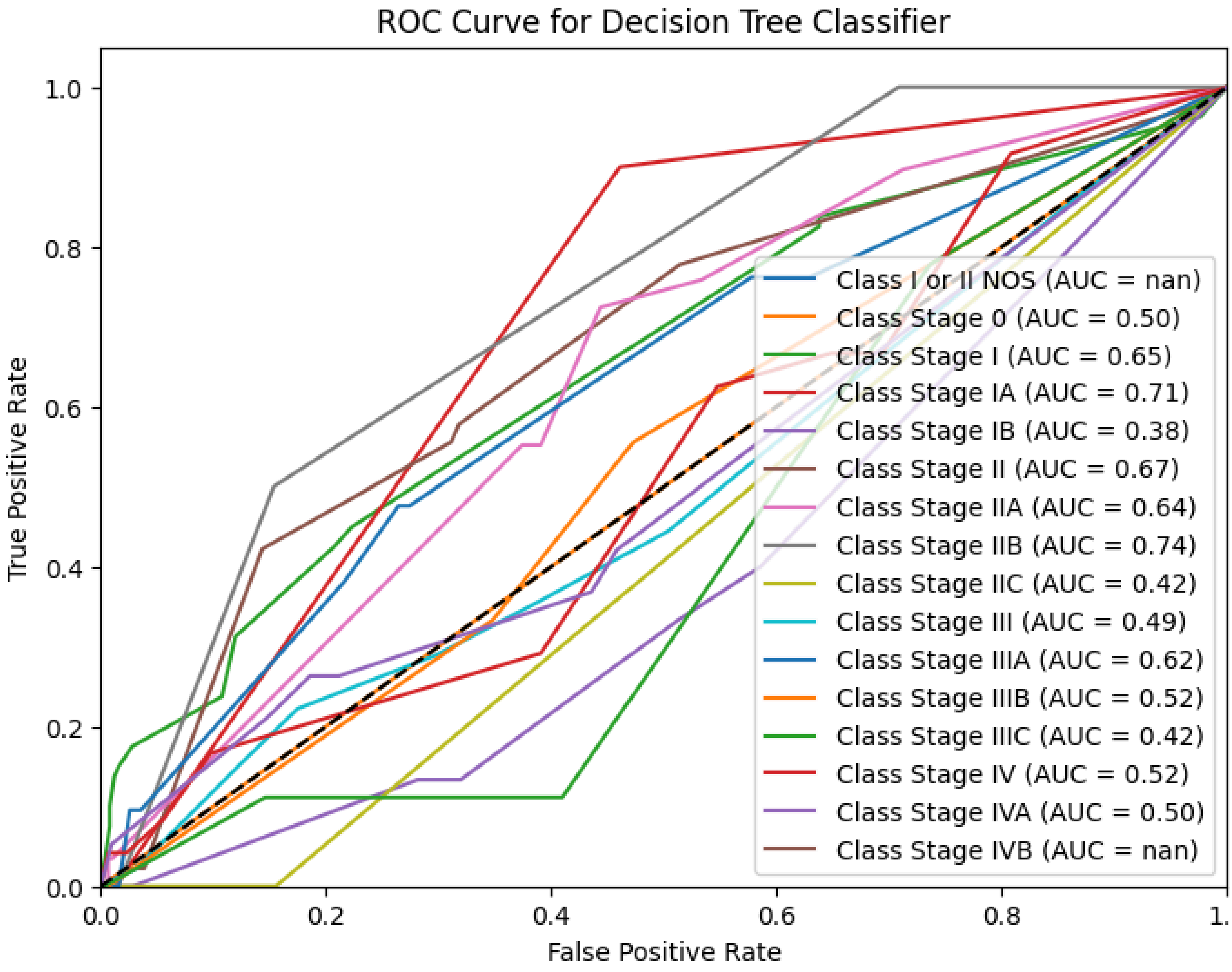
In [10]:

```
# --- prepare for roc curve---#
# Encode labels
label_encoder = LabelEncoder()
y_encoded = label_encoder.fit_transform(labels)

# Binarize for multi-class ROC
y_bin = label_binarize(y_encoded, classes=np.unique(y_encoded))
```

```
# Split data
X_train, X_test, y_train, y_test = train_test_split(features, y_bin, test_size=0.3, random_state=42)

# Train Decision Tree with OneVsRest for multi-class ROC
classifier = OneVsRestClassifier(DecisionTreeClassifier(max_depth=5))
classifier.fit(X_train, y_train)

# Predict probabilities
y_score = classifier.predict_proba(X_test)

# Compute ROC curve and AUC for each class
fpr, tpr, roc_auc = {}, {}, {}
for i in range(y_bin.shape[1]):
    fpr[i], tpr[i], _ = roc_curve(y_test[:, i], y_score[:, i])
    roc_auc[i] = auc(fpr[i], tpr[i])

# Plot ROC curves
plt.figure(figsize=(8, 6))
for i in range(y_bin.shape[1]):
    plt.plot(fpr[i], tpr[i], label=f"Class {label_encoder.inverse_transform([i])[0]} (AUC = {roc_auc[i]:.2f})")

plt.plot([0, 1], [0, 1], 'k--')
plt.xlim([0.0, 1.0])
plt.ylim([0.0, 1.05])
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.title('ROC Curve for Decision Tree Classifier')
plt.legend(loc="lower right")
plt.show()

# print area under the curve values for each class
for i in range(y_bin.shape[1]):
    print(f"AUC for class {label_encoder.inverse_transform([i])[0]}: {roc_auc[i]:.3f}")

# Micro-average AUC
fpr_micro, tpr_micro, _ = roc_curve(y_test.ravel(), y_score.ravel())
roc_auc_micro = auc(fpr_micro, tpr_micro)

# Macro-average AUC
roc_auc_macro = np.mean(list(roc_auc.values()))
```

```python
print(f"Micro-average AUC: {roc_auc_micro:.3f}")
print(f"Macro-average AUC: {roc_auc_macro:.3f}")
```

```
c:\Users\aarna\miniconda3\envs\bme2220\Lib\site-packages\sklearn\multiclass.py:90: UserWarning: Label not 1 is present in all t
raining examples.
  warnings.warn(
c:\Users\aarna\miniconda3\envs\bme2220\Lib\site-packages\sklearn\metrics\_ranking.py:1201: UndefinedMetricWarning: No positive
samples in y_true, true positive value should be meaningless
  warnings.warn(
c:\Users\aarna\miniconda3\envs\bme2220\Lib\site-packages\sklearn\metrics\_ranking.py:1201: UndefinedMetricWarning: No positive
samples in y_true, true positive value should be meaningless
  warnings.warn(
```


ROC Curve for Decision Tree Classifier
True Positive Rate
False Positive Rate
0.0
0.2
0.4
0.6
0.8
1.0
Class I or II NOS (AUC = nan)
Class Stage 0 (AUC = 0.50)
Class Stage I (AUC = 0.65)
Class Stage IA (AUC = 0.71)
Class Stage IB (AUC = 0.38)
Class Stage II (AUC = 0.67)
Class Stage IIA (AUC = 0.64)
Class Stage IIB (AUC = 0.74)
Class Stage IIC (AUC = 0.42)
Class Stage III (AUC = 0.49)
Class Stage IIIA (AUC = 0.62)
Class Stage IIIB (AUC = 0.52)
Class Stage IIIC (AUC = 0.42)
Class Stage IV (AUC = 0.52)
Class Stage IVA (AUC = 0.50)
Class Stage IVB (AUC = nan)

```
AUC for class I or II NOS: nan
AUC for class Stage 0: 0.500
AUC for class Stage I: 0.655
AUC for class Stage IA: 0.714
AUC for class Stage IB: 0.384
AUC for class Stage II: 0.672
AUC for class Stage IIA: 0.637
AUC for class Stage IIB: 0.742
AUC for class Stage IIC: 0.422
AUC for class Stage III: 0.486
AUC for class Stage IIIA: 0.624
AUC for class Stage IIIB: 0.522
AUC for class Stage IIIC: 0.417
AUC for class Stage IV: 0.517
AUC for class Stage IVA: 0.499
AUC for class Stage IVB: nan
Micro-average AUC: 0.734
Macro-average AUC: nan
```

Verify and validate your analysis:

Based on our AUC analysis, we can say that our decision tree model is quite inconsistent. While some of the curves perform better than baseline (AUC >> 0.5), a lot of them don't, or even perform worse than the baseline. Stages IA, II and IIB have highest AUC results, which suggests that the model is more promising for predicting the earlier stages of cancer. It should be noted that this may just be because of a larger availability of data of earlier cancer stages (shown below).

Overall, the model and the AUC analysis suggests that data about VEGF, FBF and THBS expression is better at indicating early stages of cancer, but it's hard to be definite about this given the weak results.

Our tendency analysis, in which we froze two of the three genes at their median value, and varied the third one to analyze its relationship with stage of cancer revealed the following trends: Increased VEGFA expression leads to more advanced cancer. Increased FGF1 leads to more advanced cancer. There is no clear trend between THBS and the stage of cancer.

These genes are well researched and well-linked to cancer progression in previous literature. A paper titled 'VEGFA Gene Expression in Breast Cancer Is Associated With Worse Prognosis, but Better Response to Chemotherapy and Immunotherapy' found a link between greater VEGFA Expression and greater cell proliferation in cancerous regions, which lines up with our findings that the genetic data was able to provide us with a model that predicted cancer stage better than random, and our observed positive trend between VEGF expression and stage of cancer. Similarly, a paper titled 'Altered Splicing of FGFR1 Is Associated with High Tumor Grade and Stage and Leads to Increased Sensitivity to FGF1 in Bladder Cancer' validates the relationship we found between FGF1 and cancer stage.

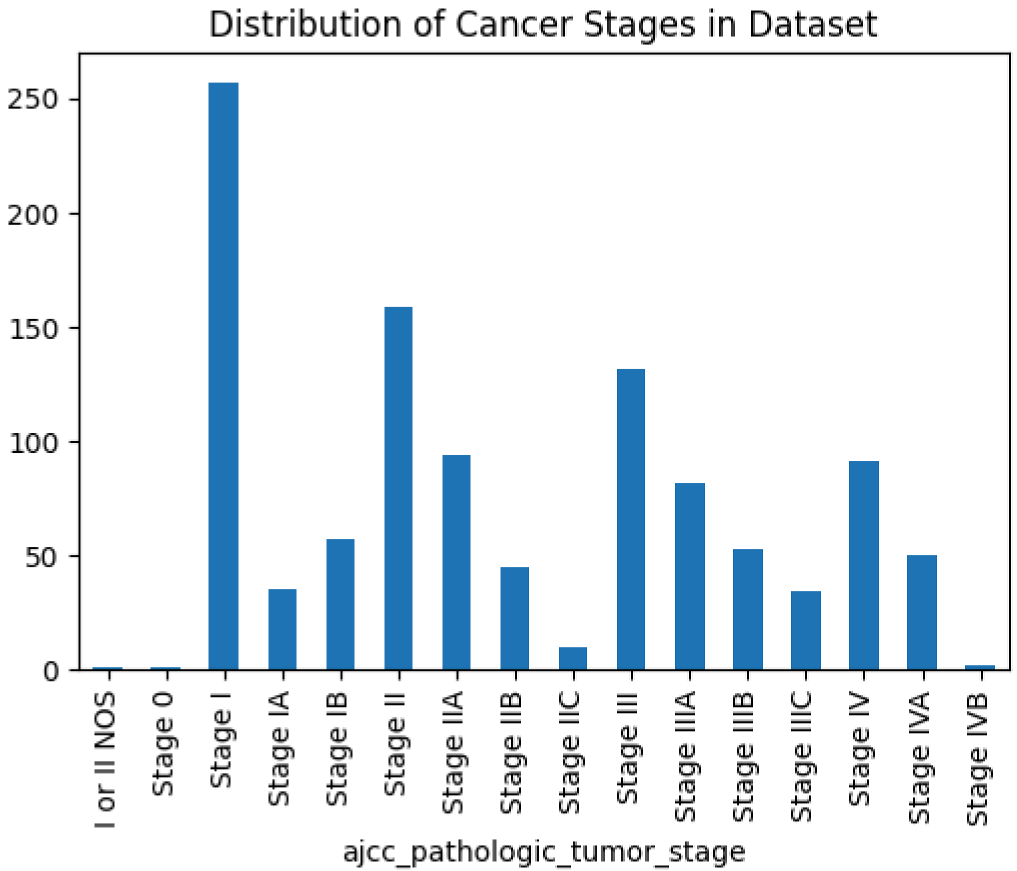
Literature, like our decision tree, is less clear about how THBS1 affects angiongenesis and cancer stage. Some papers, such as 'THBS1-producing tumor-infiltrating monocyte-like cells contribute to immunosuppression and metastasis in colorectal cancer' suggest a positive effect. However, others which have used gene-editing to block THBS1 expression have observed that 'THBS1 inhibits proliferation, adhesion, and migration and affects the cell cycle of gliomas via TNF/MAPK/NF-κB and TGF-β/Smad signaling pathways.' Again, the literature agrees with our general trend over here.

However, the reason we don't see great accuracy or results is likely that these are only a few of the genes that have been shown to be important with cancer progression, and there are many other factors at play which we don't consider. This, combined with a limited sample size for many of the stages of cancer (displayed in the bar chart below), contribute to the poor accuracy.

https://pmc.ncbi.nlm.nih.gov/articles/PMC11750749/
https://pubmed.ncbi.nlm.nih.gov/20889570/#:~:text=in%20bladder%20cancer-,Altered%20splicing%20of%20FGFR1%20is%20associated%20with%
https://pubmed.ncbi.nlm.nih.gov/37749092/#:~:text=Mesenchymal%20activation%2C%20characterized%20by%20dense,malignancy%20with%20a
https://www.sciencedirect.com/science/article/abs/pii/S0898656823000700

In [13]:

```python
# Our data by stage of cancer

# Plotting a bar graph of the different stages of cancer in the dataset
plt.figure(figsize=(6, 4))
merged_data['ajcc_pathologic_tumor_stage'].value_counts().sort_index().plot(kind='bar')
plt.title('Distribution of Cancer Stages in Dataset')
plt.show()
```



Conclusions and Ethical Implications:

Our decision tree grouped data into categories labelled by the different cancer stages based on expression of the three genes VEGFA, FGF1, and THBS1. We did a simple accurary test and a ROC curve on the results of our decision tree. The general accuracy test revealed an accuracy of 24%, which is not high, but significantly greater than 6% (which would be the accuracy of guessing). The area under the ROC curve revealed that two stages of cancer (IIB and IA) have what is considered an "acceptable" auc value, meaning the model predicts these classes well. The micro-average auc was 0.735, which also meets the threshold of an 'acceptable' value. This metric gives equal weight to each data point and how well it was predicted. Overall, this shows that our model has some accuracy in predicting classes, and is generally better than plain guessing. Our project question was to investigate how the signalling proteins from these genes relate to the different stages of cancer.

Because the model classified based on the expression of VEGFA, FGF1, and THBS1, we conclude that the expression of these genes is an indication of cancer stage. More specifically, expression of VEGFA, FGF1, and THBS1 is a potential indicator of the development of early stages of cancer across cancer types.

Part of our project question was to explore the relation between angiogenesis and stage of cancer. We investigated the question by looking into expression of the genes for angiogenesis signalling proteins. Because these genes code for angiogenesis signalling proteins, another conclusion is that angiogenesis ocurrence indicates the development of early stages of cancer.

Overall, our model suggests that it is possible to explore using angiogensis and/or the expression of angiogenesis signalling proteins to classify the stage of cancer a patient has developed.

Because of the low accuracy of our model and the fact that predicting some classes was no better than randomly guessing, it would be dangerous to rely on this method without further testing and specification. If the model could be further specified to be more accurate, testing gene expression could be a less invasive method to diagnose patients' cancer stage without requiring access to the tumor.

Limitations and Future Work:

This investigation was limited by a very broad scope, including all types of cancers. Considering that different cancers operate in different ways, with different genetic pathways, it's worth looking at individual cancer types to see how results change. However, this would introduce the limitation of a smaller dataset.

An area for future improvement is our loss function. Instead of a loss function that only distinguishes between correct and incorrect predictions, we could use a loss function that penalises for how far our prediction is. For example, our current loss function would give the same outcome if the model predicted that Stage I cancer was actually Stage II or Stage IV, although the Stage IV prediction should be penalised more. This is a large area for improvement and could significantly improve the accuracy of our model, and avoid large errors.

NOTES FROM YOUR TEAM:

week one: maggie filled in the background on cancer and aarnav did the dataset background and information about the genes we will investigate. we decided our question in class and started to investigate a model to answer it. week two: aarnav wrote the code for the decision tree model. maggie filled in the information about how it works. week three: maggie added the code for the roc curve and the area under the curve. aarnav filled in external validation information as well as the explanation for the roc-auc test and what we can learn about our model.

final submission: Maggie added conclusions and ethical implications and aarnav added limitations and future work, and the distribution of cancer stages in dataset analysis.

QUESTIONS FOR YOUR TA:

none! Thank you!