Module_3: *Cancer*

Team Members:

Addison and Ustav

Project Title:

Using binary classification to predict whether Stomach Adenocarcinoma samples are benign or malignant by using reported tumor status as the label and EGFR expression as the predictor.

Project Goal:

This project seeks to... *determine if binary classification methods (specifically logistic regression) can be used to predict if carcinomas are benign or malignant. This will be based on gene expression of EGFR, and subsequent other genes involved in evading apoptosis, in stomach adenocarcinoma. In this machine learning method, the loss function will be minimized in order to produce probabilities vlose to the true labels.*

Disease Background:

- Cancer hallmark focus: Evading Apoptosis

- Overview of hallmark: The second hallmark of cancer is evading apoptosis. Apoptosis refers to regulated cell death and is essential for tissue maintenance. The process is influenced by both sensors and effectors. Sensors recognize through intracellular and extracellular conditions if a cell should be terminated, and effectors carry out the process of cell death through cell surface receptors and factors that initiate cell death. Some receptor pairs and pathways involved in this are the IGF and IL receptors, and the FAS and TNF binding acting as death factors. Additionally, the suppression of the p53 protein that typically allows for signaling in apoptosis is a pathway factor to consider in tumor cells evading apoptosis. Finally, the PI3 kinase - AKT/PKB contributes to this by transmitting antiapoptotic signals.

- Genes associated with hallmark to be studied (describe the role of each gene, signaling pathway, or gene set you are going to investigate): We will be focusing on EGFR, FLT3, and FGFR1, as they are genes associated with resisting cell death and present in the dataset. These genes block cell death by activating certain pathways such as the P13K/AKT pathway, which allow for increased cell proliferation and survival. When EGFR (a receptor tyrosine kinase) is activated, it activates this pathway, which causes AKT to phosphorylate pro-apoptotic proteins and inhibit transcription factors. This prevents the transcription of other pro-apoptotic proteins such as BIM and FASL. FLT3 is a hematopoietic receptor tyrosine kinase. This also inhibits transcription of pro-apoptotic proteins, but also upregulated MCL-1 which prevents mitochondrial apoptosis. FGFR1 activates the MAPK/ERK pathway, which leads to survival and proliferation by upregulating genes associated with promoting the cell-cycle, suppressing pro-apoptotic proteins, and increasing expression of BCL-2 proteins. FGFR1 specifically also activates the P13K/AKT pathway, which was previously discussed, and enhances STAT signaling, which enter the nucleas and increase anti-apoptotic gene expression. (ChatGPT: prompt: describe how EGFR and FLT3 function to allow cells to evade apoptosis. Describe how FGFR1 functions to allow cells to evade apoptosis.).

- Will be focusing on Stomach Adenocarcinoma

Prevalence & incidence

- In 2020, 1.1 million new cases of stomach cancer were recorded globally. The majority of these cases (75%) were recorded in Asia, and these cases made of 5.6% of all new cancer cases. After 5 years, there is a survival rate of around 20%, making it one of the more lethal cancers (https://pmc.ncbi.nlm.nih.gov/articles/PMC8968487/). The incidence of stomach adenocarcinoma is about 30,000 new cases per year in the United States, and the prevalence is about 140,000 people living with stomach adenocarcinoma in the United States. It should be noted that incidence rates are declining (ChatGPT prompt: what are the prevalence and incidence rates of stomach adenocarcinoma in the United States).

Risk factors (genetic, lifestyle) & Societal determinants

- Stomach cancer is more common in Asians, Hispanic Americans, African Americans, and Natives than in non-Hispanic whites. Behavioral risks include smoking, alcohol use, obesity, dies rich in salty food or low in fruits and vegetables, and exposure to dust. Other groups that have a higher risk are being aged 50 and over, being male, having blood type A, having gastritis (long-term inflammation of the stomach), autoimmune diseases, megaloblastic anemia, and H. pylori bacteria infection. Some genetic mutations that may cause someone to be at higher risks are mutations of the CDH!, CTNNA1, ATM, KIT, and PDGFRA genes (https://nostomachforcancer.org/about-stomach-cancer/risks-genetics-prevention-of-stomach-cancer).

Symptoms

- According to Mayo Clinic, some symptoms of stomach cancers inclode trouble swallowing, belly pain, feeling bloated or full after eating, not feeling hungry when expected, heartburn, indigestion, nausea, vomiting, weight loss, fatigue, or black stool. However, some people may not experience symptoms until later stages of the cancer. The symptoms of more advanced stages of stomach cancer are extreme fatigue, weight loss, vomiting blood, and black stool. When stomach cancer is metastatic, various symptoms may appear in other areas of the body (https://www.mayoclinic.org/diseases-conditions/stomach-cancer/symptoms-causes/syc-20352438)

Standard of care treatments (& reimbursement)

- Some FDA-approved immunotherapies are used in combination with chemotherapy to treat this cancer type (https://www.opdivo.com/gastroesophageal-cancer). If the disease is in an early stage, surgery may be used to remove the tumor. However, for more advanced cancer types, chemotherapy may also be necessary in order to treat cancer. If it is determined that the cancer is not able to be resected/removed, systemic therapy is used in the form of fluoropyrimidine treatment or immunotherapy. Many treatment options are covered by MediCare, however novel therapies may require additional approval (ChatGPT prompt: what are the standard of care treatments and reimbursement options for patients with stomach adenocarcinoma?)

Biological mechanisms (anatomy, organ physiology, cell & molecular physiology)

- Gastric cancer is influenced by bacterial, environmental, and genetic factors. Stomach cancer originates as a mucosal atrophy. As the stomach becomes more inflamed, the epithelial cells in the mucosal layer are damaged, leading to less stomach acid, abnormal growth, and malignant tumors. As mentioned, some bacterial factors can cause stomach cancer. Bacterial infections from H. pyolri are especially problematic as they are able to neutralize stomach acid and survive in the stomach. Because of this, the bacteria is able to embed itself in the mucus layer and cause the aforementioned inflammation. As the patient responds to infection, the immune system worsens inflammation through the release of cytokines and nitric oxide. Eventually, genes that promote growth become hyperactive (including the p53 gene involved in evading apoptosis). Growth signals are released that promote the growth of new cells, and support cells stimulating their own growth. Growth signals that cause this are EGF, VEGF, and IL-1a. As damage accumulates, the DNA repair genes are damages, causing further instability. Physiologically, the mucosal layer of the stomach is damaged, and cancer cells are able to invade deeper layers (https://pmc.ncbi.nlm.nih.gov/articles/PMC4124370/).*

Data-Set:

- We will be looking at Stomach Adenocarcinoma, largely focusing on ALK and BCL-2.

Once you decide on the subset of data you want to use (i.e. only 1 cancer type or many; any clinical features needed?; which genes will you look at?) describe the dataset. There are a ton of clinical features, so you don't need to describe them all, only the ones pertinent to your question.

- One clinical feature needed is tumor cell stage, which describes the progession of cancer, describing the amount of tumor growth and spread.

(Describe the data set(s) you will analyze. Cite the source(s) of the data. Describe how the data was collected -- What techniques were used? What units are the data measured in? Etc.)

- The dataset is from The Cancer Genome Atlas (TCGA) which is a big repository of multiple types of cancer. 24 cancer types are included. The atlas comprises RNA sequencing data from numerous tumor and healthy cells. The RNA sequencing data were collected from patient tumor biopsies that underwent the extraction of their RNA. The original format of the TCGA used a tool called Tophat to align the sequencing genes to the human genome so that researchers could see which gene the RNA comes from. The metadata included is matched with the patient ID, connecting it to tumor stage, cancer type, and age. Comparing gene expressions across different stages of tumors aims to identify biomarkers and signaling patterns that may contribute to the growth of tumors and resistance to apoptosis. The data is subsetted to include 3000 out of the 15000 protein coding genes, and 50-100 tumors per cancer type out of 9264 total tumors. The metadata, which describes patient-related information, includes 70 columns from the original 56, and not all columns are present for all cancer types. Genes X samples data shows log2(TPM+1). Besides the RNA sequencing data and metadata, information about the metadata and which cancer types have representative data for metadata categories are included in two other files. Citation: Rahman, M., Jackson, L., Johnson, W., Li, D., Bild, A., & Riccolo, S. (2015, July 24). Alternative preprocessing of RNA-sequencing data in the cancer genome atlas leads to improved analysis results. PubMed. https://pubmed.ncbi.nlm.nih.gov/26209429/

- Through search of the CSV, it was determined that the EGFR and FLT3 genes were present in the dataset.*

Data Analysis:

Methods

The machine learning technique I am using is: *The method that we are using is binary classification. We chose this method as we are deciding between two mutually exclusive categories, those being tumor-free (0) and with tumor (1). We are also using logistic regression, which estimates the probability that a sample belongs to the positive class based on a predictor. That predictor is gene expression level, and we chose genes that play a role in evading apoptosis.*

What is this method optimizing? How does the model decide it is "good enough"? The goal of this method is to minimize the loss function, which minimizes how far its predictions are from actual labels. The loss function here is a log loss, and minimizing the loss produces probabilities closer to the true labels. The method is determined to be "good enough" through validation methods. The training set is used to optimize the parameters used, and the test set is used to determine how well the generalization works. An ROC view is used to determine this, which determines the discriminative ability. Validation methods are further discussed in the verify and validate section.

**

Analysis

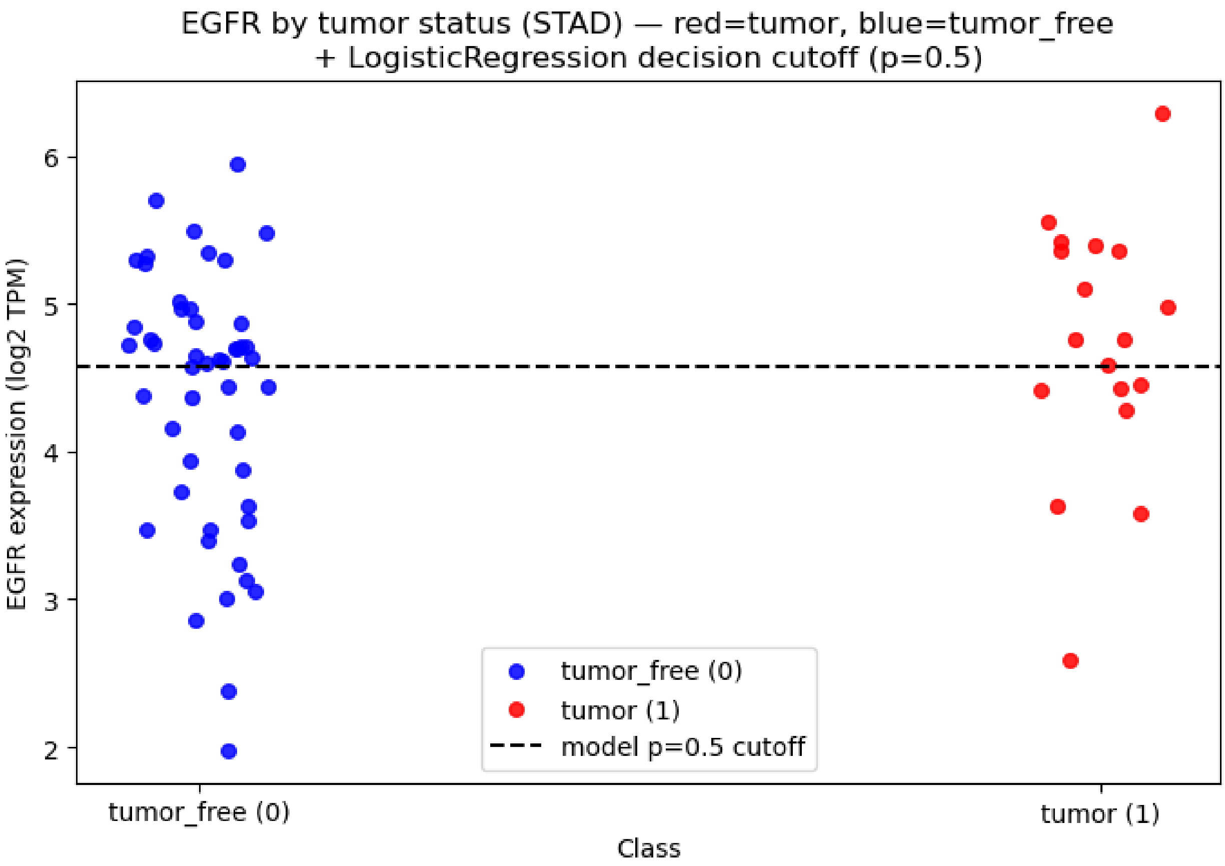
Shown below, first kernel is for EGFR gene and subsequent kernels test other genes/types of cancer

```
"""
There are 2 plots being graphed in this code block. The first plot is
a binary scatter
plot where each dot represents a sample, Red = tumor (1), Blue =
tumor_free (0).
The y axis is the corresponding EGFR value of each sample.
The dashed line is the threshhold for EGFR value for a predicted tumor
where p=0.5 (By default)
"""
"""
The second plot sweeps through all the tumor stage (In this case 1 ->
0)
The y axis is the fraction of tumors that were correctly called tumors
The x axis is the fraction of tumors that were incorrectly called
tumors (false positives)
"""
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt

from sklearn.pipeline import make_pipeline
from sklearn.preprocessing import StandardScaler
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score, precision_score,
recall_score, f1_score, roc_auc_score, RocCurveDisplay
from sklearn.model_selection import train_test_split

# ---- Files in same folder ----
METADATA_CSV = "GSE62944_metadata.csv"
EXPR_CSV     = "GSE62944_subsample_topVar_log2TPM.csv"
ID_COL       = "sample"   # adjust if your metadata uses a different
sample ID

# 1) Get labels for STAD (sample id + tumor_status)
meta = pd.read_csv(METADATA_CSV)
stad = meta.loc[meta["cancer_type"].astype(str).str.upper()=="STAD",
[ID_COL, "tumor_status"]].copy()

# 2) Pull EGFR expression (genes-as-rows; first column is the gene
name)
expr = pd.read_csv(EXPR_CSV)
gene_col = expr.columns[0]
egfr_row = expr.loc[expr[gene_col].astype(str).str.upper()=="EGFR"]

# Convert EGFR row to long format: (sample_id, EGFR_expr)
egfr_long = (
    egfr_row
    .melt(id_vars=[gene_col], var_name=ID_COL, value_name="EGFR_expr")
    [[ID_COL, "EGFR_expr"]]
)
egfr_long["EGFR_expr"] = pd.to_numeric(egfr_long["EGFR_expr"],
errors="coerce")

# 3) Merge features + labels and map tumor_status -> y ∈ {0,1}
df = pd.merge(stad, egfr_long, on=ID_COL,
how="inner").dropna(subset=["EGFR_expr", "tumor_status"]).copy()
lab =
(df["tumor_status"].astype(str).str.strip().str.lower().str.replace(r"
\s+", " ", regex=True))
exact_map = {"tumor free": 0, "with tumor": 1}
df["y"] = lab.map(exact_map)
```

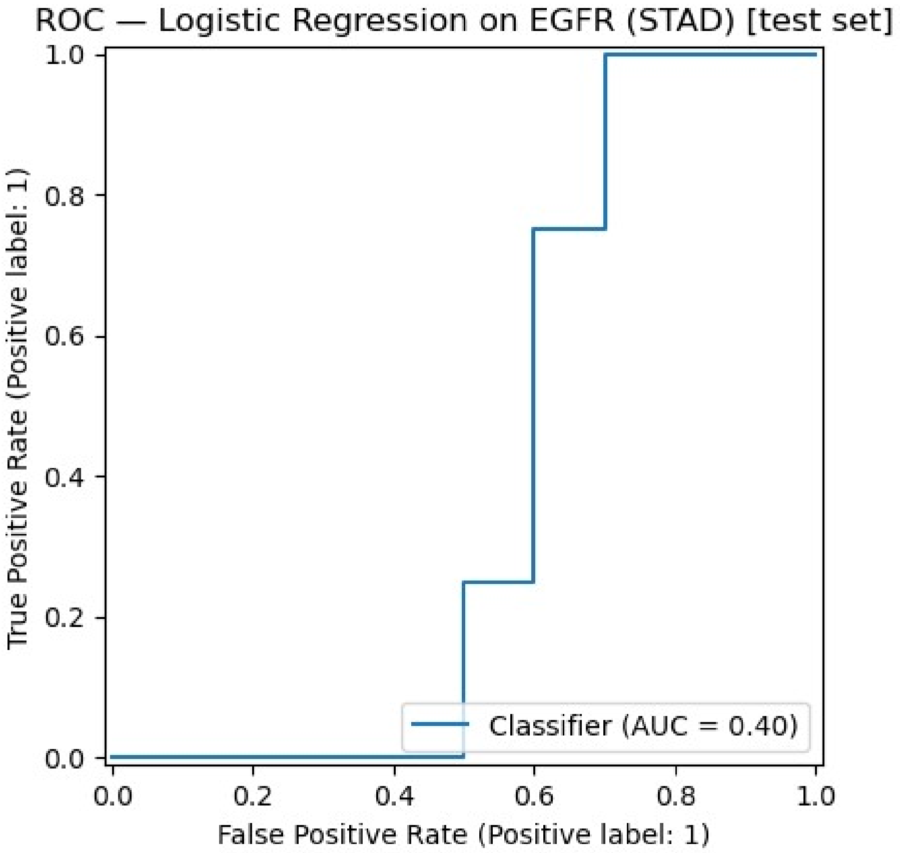
```
mask = df["y"].isna()
df.loc[mask & lab.str.contains(r"\btumor free\b"), "y"] = 0
df.loc[mask & lab.str.contains(r"normal|free|control|benign|negative"), "y"] = 0
df.loc[mask & lab.str.contains(r"with tumor|tumor|cancer|malignant|primary|recurrent"), "y"] = 1
df = df.dropna(subset=["y"]).copy()
df["y"] = df["y"].astype(int)

# Feature/target
X = df[["EGFR_expr"]].to_numpy()
y = df["y"].to_numpy()

# 4) Hold-out evaluation split (keeps class balance via stratify)
X_tr, X_te, y_tr, y_te = train_test_split(X, y, test_size=0.2, random_state=42, stratify=y)

# 5) Pipeline: standardize EGFR + logistic regression (balanced classes)
model = make_pipeline(StandardScaler(), LogisticRegression(solver="liblinear", class_weight="balanced", random_state=42))
model.fit(X_tr, y_tr)

# Quick test metrics (default threshold p>=0.5)
p_te = model.predict_proba(X_te)[:, 1]
y_pred = (p_te >= 0.5).astype(int)
print("Test metrics (EGFR-only):")
print(f"  Accuracy : {accuracy_score(y_te, y_pred):.3f}")
print(f"  Precision: {precision_score(y_te, y_pred, zero_division=0):.3f}")
print(f"  Recall   : {recall_score(y_te, y_pred, zero_division=0):.3f}")
print(f"  F1       : {f1_score(y_te, y_pred, zero_division=0):.3f}")
print(f"  AUC      : {roc_auc_score(y_te, p_te):.3f}")

# 6) Convert the model's p=0.5 rule back to a raw EGFR cutoff (draw as dashed line)
#    For standardized x: z=(x-mean)/scale; decision boundary at w*z + b = 0 → x_cut = mean - (b*scale)/w
scaler = model.named_steps["standardscaler"]
clf    = model.named_steps["logisticregression"]
w = clf.coef_[0, 0]
b = clf.intercept_[0]
mean_  = scaler.mean_[0]
scale_ = scaler.scale_[0]
egfr_cut = mean_ - (b * scale_) / w if w != 0 else np.nan
print(f"\nDecision threshold on EGFR (p=0.5): {egfr_cut:.4f} (raw EGFR units)")
```

```
# 7) Dot plot: each sample as a point (blue=tumor_free, red=tumor) +
horizontal cutoff line
rng = np.random.default_rng(42)
jitter = rng.uniform(-0.08, 0.08, size=len(df))          # small x-
jitter so points don't stack
xpos = df["y"].to_numpy().astype(float) + jitter          #
0≈tumor_free, 1≈tumor
yvals = df["EGFR_expr"].to_numpy()
is_tumor = df["y"].to_numpy() == 1

plt.figure(figsize=(7,5))
plt.scatter(xpos[~is_tumor], yvals[~is_tumor], s=30, alpha=0.85,
c="blue", label="tumor_free (0)")
plt.scatter(xpos[ is_tumor], yvals[ is_tumor], s=30, alpha=0.85,
c="red",  label="tumor (1)")
if np.isfinite(egfr_cut):
    plt.axhline(egfr_cut, linestyle="--", linewidth=1.5,
color="black", label="model p=0.5 cutoff")
plt.xticks([0, 1], ["tumor_free (0)", "tumor (1)"])
plt.xlabel("Class")
plt.ylabel("EGFR expression (log2 TPM)")
plt.title("EGFR by tumor status (STAD) – red=tumor, blue=tumor_free\n+
LogisticRegression decision cutoff (p=0.5)")
plt.legend(loc="best")
plt.tight_layout()
plt.show()

# 8) ROC curve on the test split (threshold-free performance view)
plt.figure()
RocCurveDisplay.from_predictions(y_te, p_te)
plt.title("ROC – Logistic Regression on EGFR (STAD) [test set]")
```

```
Test metrics (EGFR-only):
  Accuracy : 0.357
  Precision: 0.143
  Recall   : 0.250
  F1       : 0.182
  AUC      : 0.400

Decision threshold on EGFR (p=0.5): 4.5728 (raw EGFR units)
```



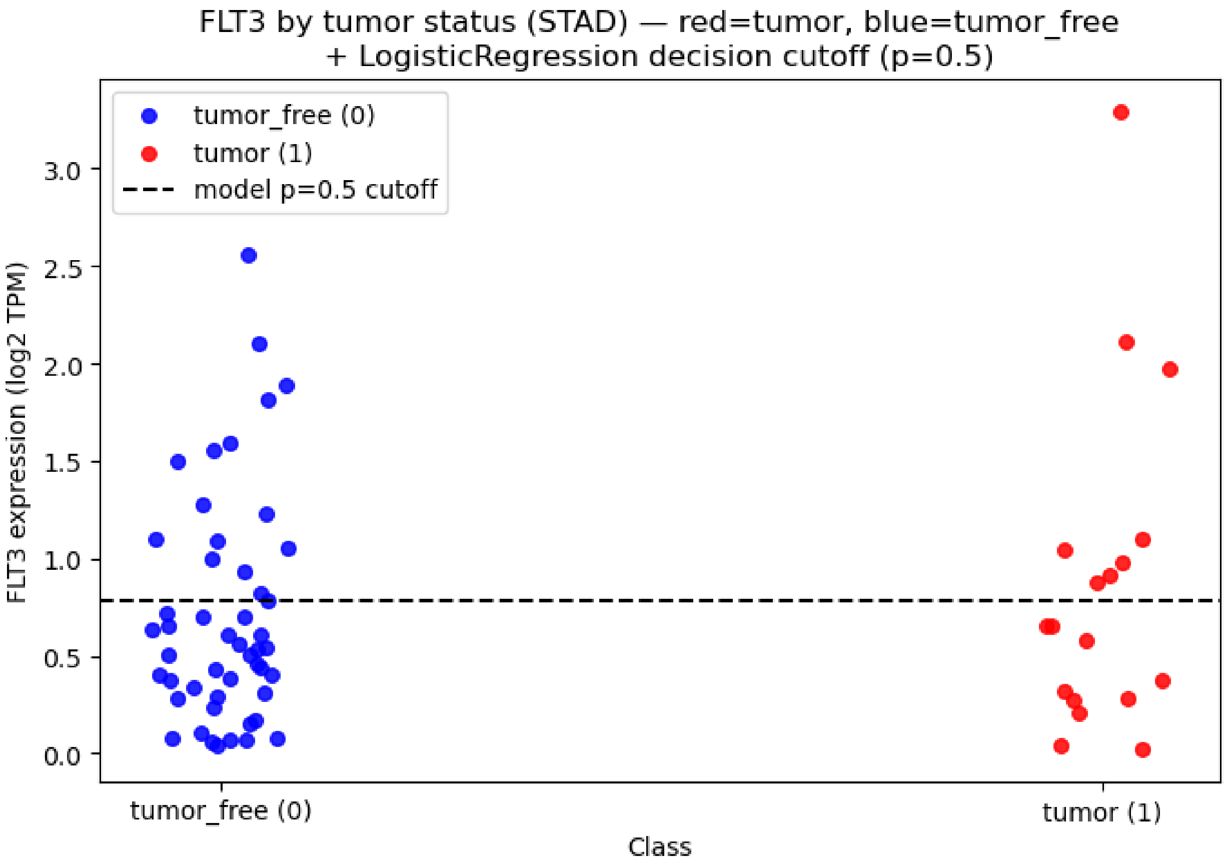
```
Text(0.5, 1.0, 'ROC – Logistic Regression on EGFR (STAD) [test set]')

<Figure size 640x480 with 0 Axes>
```



```
# As our original model for EGFR was not performing well (based on the
below-random AUC curve), we looked at other genes in the same pathway
to determine if they were better predictors of tumor status
# first alternative gene is FLT3
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt

from sklearn.pipeline import make_pipeline
from sklearn.preprocessing import StandardScaler
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score, precision_score,
recall_score, f1_score, roc_auc_score, RocCurveDisplay
from sklearn.model_selection import train_test_split

# ---- Files in same folder ----
METADATA_CSV =
"/Users/addisonbuck/Downloads/Fibrosis/Cancer/GSE62944_metadata
(1).csv"
EXPR_CSV     =
"/Users/addisonbuck/Downloads/Fibrosis/Cancer/GSE62944_subsample_topVa
r_log2TPM 3.csv"
ID_COL       = "sample"   # adjust if your metadata uses a different
```

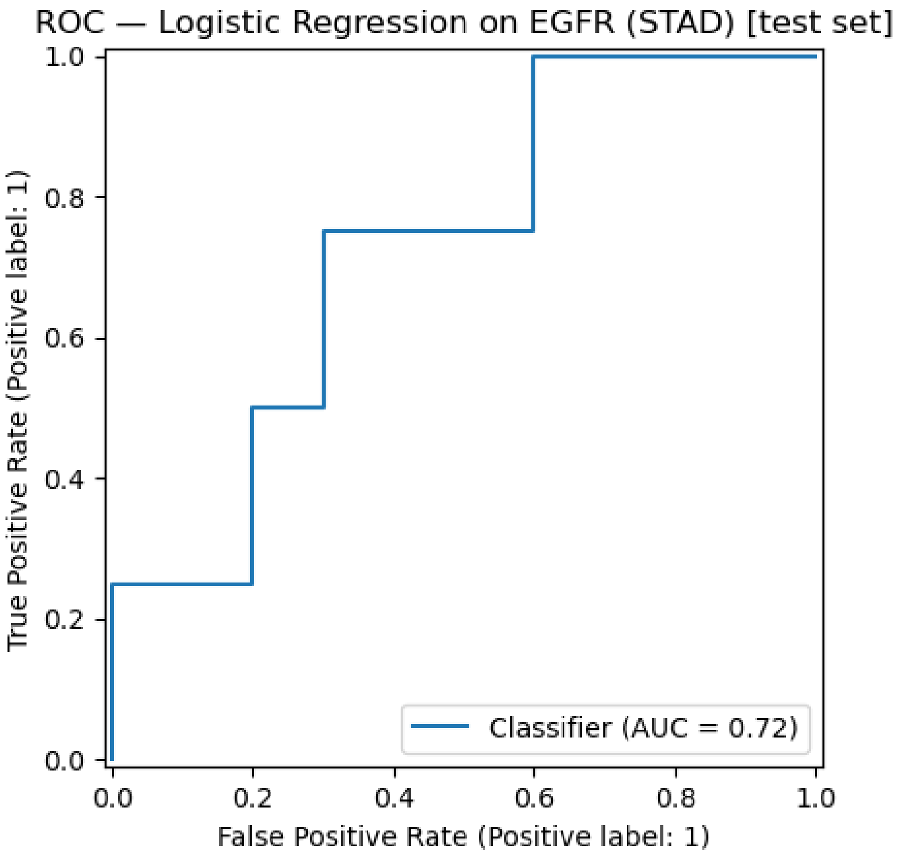
```
sample ID

# 1) Get labels for STAD (sample id + tumor_status)
meta = pd.read_csv(METADATA_CSV)
stad = meta.loc[meta["cancer_type"].astype(str).str.upper()=="STAD",
[ID_COL, "tumor_status"]].copy()

# 2) Pull FLT3 expression (genes-as-rows; first column is the gene
name)
expr = pd.read_csv(EXPR_CSV)
gene_col = expr.columns[0]
egfr_row = expr.loc[expr[gene_col].astype(str).str.upper()=="FLT3"]

# Convert FLT3 row to long format: (sample_id, FLT3_expr)
egfr_long = (
    egfr_row
    .melt(id_vars=[gene_col], var_name=ID_COL, value_name="FLT3_expr")
    [[ID_COL, "FLT3_expr"]]
)
egfr_long["FLT3_expr"] = pd.to_numeric(egfr_long["FLT3_expr"],
errors="coerce")

# 3) Merge features + labels and map tumor_status -> y ∈ {0,1}
df = pd.merge(stad, egfr_long, on=ID_COL,
how="inner").dropna(subset=["FLT3_expr", "tumor_status"]).copy()
lab =
(df["tumor_status"].astype(str).str.strip().str.lower().str.replace(r"
\s+", " ", regex=True))
exact_map = {"tumor free": 0, "with tumor": 1}
df["y"] = lab.map(exact_map)
mask = df["y"].isna()
df.loc[mask & lab.str.contains(r"\btumor free\b"), "y"] = 0
df.loc[mask & lab.str.contains(r"normal|free|control|benign|
negative"), "y"] = 0
df.loc[mask & lab.str.contains(r"with tumor|tumor|cancer|malignant|
primary|recurrent"), "y"] = 1
df = df.dropna(subset=["y"]).copy()
df["y"] = df["y"].astype(int)

# Feature/target
X = df[["FLT3_expr"]].to_numpy()
y = df["y"].to_numpy()

# 4) Hold-out evaluation split (keeps class balance via stratify)
X_tr, X_te, y_tr, y_te = train_test_split(X, y, test_size=0.2,
random_state=42, stratify=y)

# 5) Pipeline: standardize EGFR + logistic regression (balanced
classes)
model = make_pipeline(StandardScaler(),
LogisticRegression(solver="liblinear", class_weight="balanced",
random_state=42))
model.fit(X_tr, y_tr)

# Quick test metrics (default threshold p>=0.5)
p_te = model.predict_proba(X_te)[:, 1]
y_pred = (p_te >= 0.5).astype(int)
print("Test metrics (FLT3-only):")
print(f"  Accuracy : {accuracy_score(y_te, y_pred):.3f}")
print(f"  Precision: {precision_score(y_te, y_pred,
zero_division=0):.3f}")
print(f"  Recall   : {recall_score(y_te, y_pred,
zero_division=0):.3f}")
print(f"  F1       : {f1_score(y_te, y_pred, zero_division=0):.3f}")
print(f"  AUC      : {roc_auc_score(y_te, p_te):.3f}")

# 6) Convert the model's p=0.5 rule back to a raw FLT3 cutoff (draw as
dashed line)
#    For standardized x: z=(x-mean)/scale; decision boundary at w*z +
b = 0 → x_cut = mean - (b*scale)/w
scaler = model.named_steps["standardscaler"]
clf    = model.named_steps["logisticregression"]
w = clf.coef_[0, 0]
b = clf.intercept_[0]
mean_  = scaler.mean_[0]
scale_ = scaler.scale_[0]
egfr_cut = mean_ - (b * scale_) / w if w != 0 else np.nan
print(f"\nDecision threshold on FLT3 (p=0.5): {egfr_cut:.4f} (raw FLT3
units)")

# 7) Dot plot: each sample as a point (blue=tumor_free, red=tumor) +
horizontal cutoff line
rng = np.random.default_rng(42)
jitter = rng.uniform(-0.08, 0.08, size=len(df))           # small x-
jitter so points don't stack
xpos = df["y"].to_numpy().astype(float) + jitter          #
0≈tumor_free, 1≈tumor
yvals = df["FLT3_expr"].to_numpy()
is_tumor = df["y"].to_numpy() == 1

plt.figure(figsize=(7,5))
plt.scatter(xpos[~is_tumor], yvals[~is_tumor], s=30, alpha=0.85,
c="blue", label="tumor_free (0)")
plt.scatter(xpos[ is_tumor], yvals[ is_tumor], s=30, alpha=0.85,
c="red",  label="tumor (1)")
if np.isfinite(egfr_cut):
    plt.axhline(egfr_cut, linestyle="--", linewidth=1.5,
color="black", label="model p=0.5 cutoff")
plt.xticks([0, 1], ["tumor_free (0)", "tumor (1)"])
plt.xlabel("Class")
plt.ylabel("FLT3 expression (log2 TPM)")
plt.title("FLT3 by tumor status (STAD) — red=tumor, blue=tumor_free\n+
LogisticRegression decision cutoff (p=0.5)")
plt.legend(loc="best")
plt.tight_layout()
plt.show()

# 8) ROC curve on the test split (threshold-free performance view)
plt.figure()
RocCurveDisplay.from_predictions(y_te, p_te)
plt.title("ROC — Logistic Regression on FLT3 (STAD) [test set]")
```

```
Test metrics (FLT3-only):
  Accuracy : 0.643
  Precision: 0.400
  Recall   : 0.500
  F1       : 0.444
  AUC      : 0.725

Decision threshold on FLT3 (p=0.5): 0.7802 (raw FLT3 units)
```



```
Text(0.5, 1.0, 'ROC — Logistic Regression on EGFR (STAD) [test set]')
```

```
<Figure size 640x480 with 0 Axes>
```



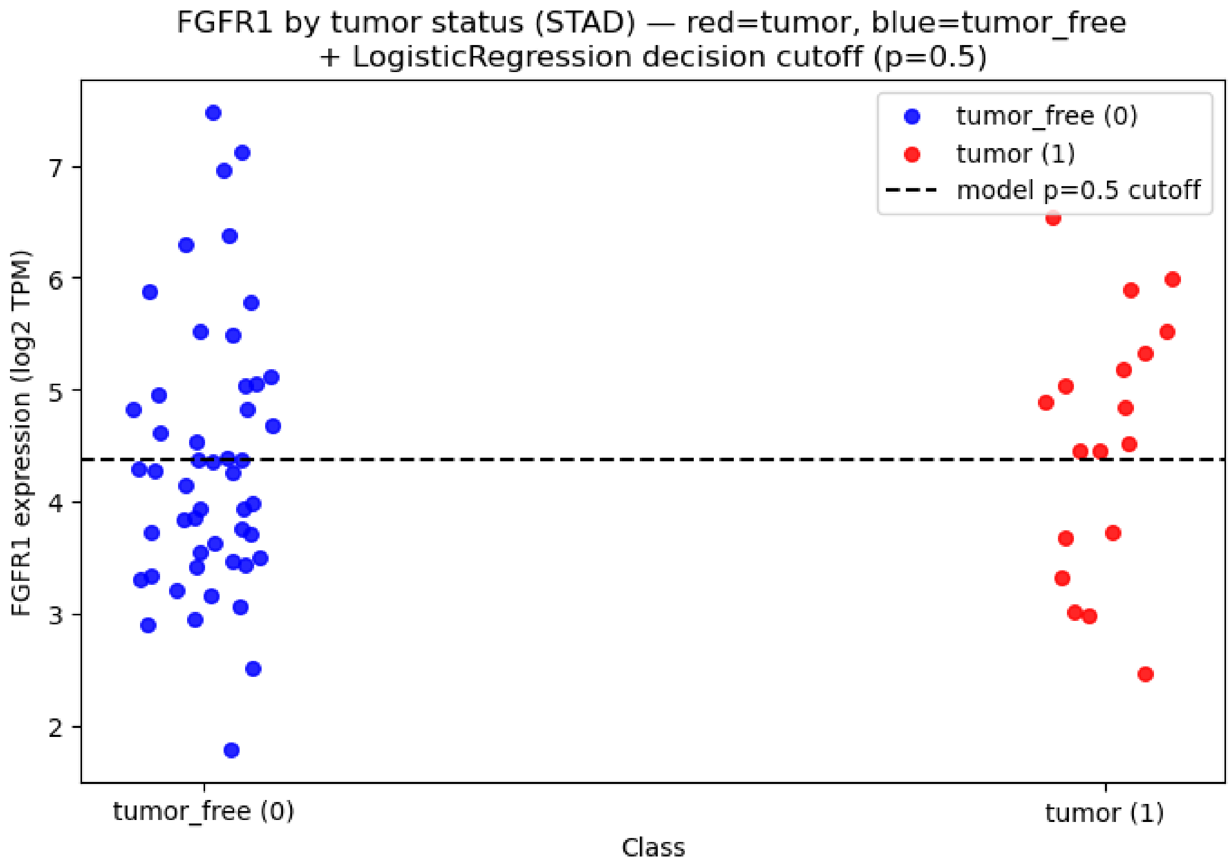
```
# While FLT3 was a better predictor, we decided to test 1 more gene
with the same pathway. We will be using FGFR1.
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt

from sklearn.pipeline import make_pipeline
from sklearn.preprocessing import StandardScaler
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score, precision_score,
recall_score, f1_score, roc_auc_score, RocCurveDisplay
from sklearn.model_selection import train_test_split

# ---- Files in same folder ----
METADATA_CSV =
"/Users/addisonbuck/Downloads/Fibrosis/Cancer/GSE62944_metadata
(1).csv"
EXPR_CSV     =
"/Users/addisonbuck/Downloads/Fibrosis/Cancer/GSE62944_subsample_topVa
r_log2TPM 3.csv"
```

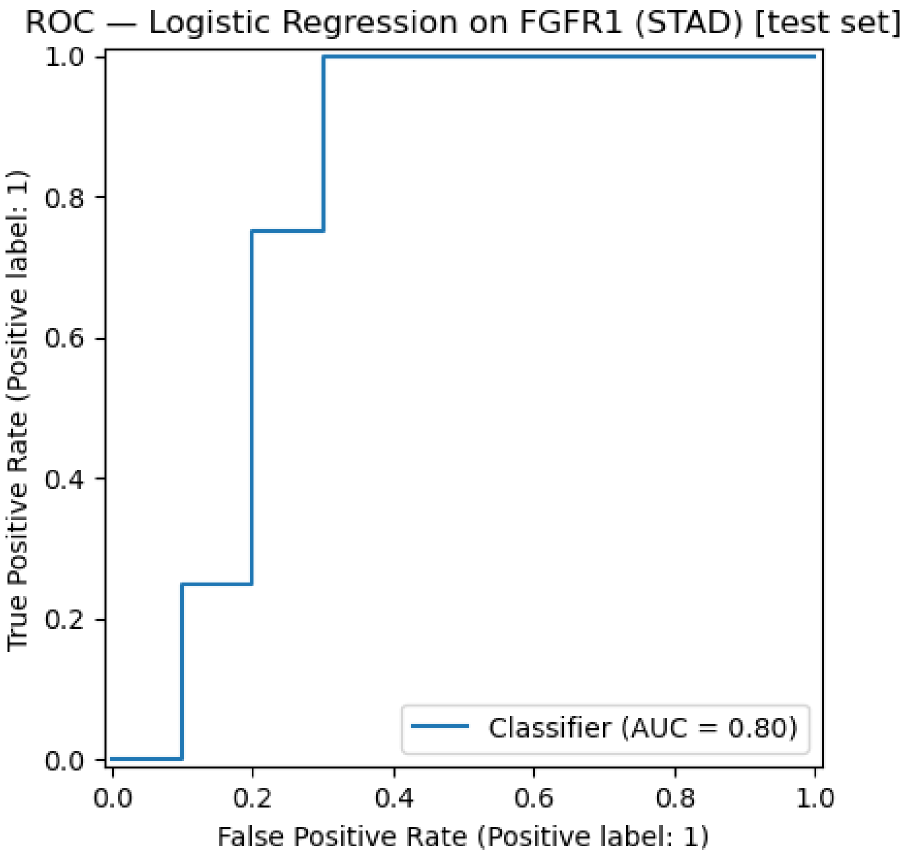
```
ID_COL       = "sample"   # adjust if your metadata uses a different
sample ID

# 1) Get labels for STAD (sample id + tumor_status)
meta = pd.read_csv(METADATA_CSV)
stad = meta.loc[meta["cancer_type"].astype(str).str.upper()=="STAD",
[ID_COL, "tumor_status"]].copy()

# 2) Pull FGFR1 expression (genes-as-rows; first column is the gene
name)
expr = pd.read_csv(EXPR_CSV)
gene_col = expr.columns[0]
FGFR1_row = expr.loc[expr[gene_col].astype(str).str.upper()=="FGFR1"]

# Convert FGFR1 row to long format: (sample_id, FGFR1_expr)
FGFR1_long = (
    FGFR1_row
    .melt(id_vars=[gene_col], var_name=ID_COL,
value_name="FGFR1_expr")
    [[ID_COL, "FGFR1_expr"]]
)
FGFR1_long["FGFR1_expr"] = pd.to_numeric(FGFR1_long["FGFR1_expr"],
errors="coerce")

# 3) Merge features + labels and map tumor_status -> y ∈ {0,1}
df = pd.merge(stad, FGFR1_long, on=ID_COL,
how="inner").dropna(subset=["FGFR1_expr", "tumor_status"]).copy()
lab =
(df["tumor_status"].astype(str).str.strip().str.lower().str.replace(r"
\s+", " ", regex=True))
exact_map = {"tumor free": 0, "with tumor": 1}
df["y"] = lab.map(exact_map)
mask = df["y"].isna()
df.loc[mask & lab.str.contains(r"\btumor free\b"), "y"] = 0
df.loc[mask & lab.str.contains(r"normal|free|control|benign|
negative"), "y"] = 0
df.loc[mask & lab.str.contains(r"with tumor|tumor|cancer|malignant|
primary|recurrent"), "y"] = 1
df = df.dropna(subset=["y"]).copy()
df["y"] = df["y"].astype(int)

# Feature/target
X = df[["FGFR1_expr"]].to_numpy()
y = df["y"].to_numpy()

# 4) Hold-out evaluation split (keeps class balance via stratify)
X_tr, X_te, y_tr, y_te = train_test_split(X, y, test_size=0.2,
random_state=42, stratify=y)

# 5) Pipeline: standardize FLT3 + logistic regression (balanced
```

```
classes)
model = make_pipeline(StandardScaler(),
LogisticRegression(solver="liblinear", class_weight="balanced",
random_state=42))
model.fit(X_tr, y_tr)

# Quick test metrics (default threshold p>=0.5)
p_te = model.predict_proba(X_te)[:, 1]
y_pred = (p_te >= 0.5).astype(int)
print("Test metrics (FGFR1-only):")
print(f"  Accuracy : {accuracy_score(y_te, y_pred):.3f}")
print(f"  Precision: {precision_score(y_te, y_pred,
zero_division=0):.3f}")
print(f"  Recall   : {recall_score(y_te, y_pred,
zero_division=0):.3f}")
print(f"  F1       : {f1_score(y_te, y_pred, zero_division=0):.3f}")
print(f"  AUC      : {roc_auc_score(y_te, p_te):.3f}")

# 6) Convert the model's p=0.5 rule back to a raw FLT3 cutoff (draw as
dashed line)
#    For standardized x: z=(x-mean)/scale; decision boundary at w*z +
b = 0 → x_cut = mean - (b*scale)/w
scaler = model.named_steps["standardscaler"]
clf    = model.named_steps["logisticregression"]
w = clf.coef_[0, 0]
b = clf.intercept_[0]
mean_  = scaler.mean_[0]
scale_ = scaler.scale_[0]
egfr_cut = mean_ - (b * scale_) / w if w != 0 else np.nan
print(f"\nDecision threshold on FGFR1 (p=0.5): {egfr_cut:.4f} (raw
FGFR1 units)")

# 7) Dot plot: each sample as a point (blue=tumor_free, red=tumor) +
horizontal cutoff line
rng = np.random.default_rng(42)
jitter = rng.uniform(-0.08, 0.08, size=len(df))           # small x-
jitter so points don't stack
xpos = df["y"].to_numpy().astype(float) + jitter          #
0≈tumor_free, 1≈tumor
yvals = df["FGFR1_expr"].to_numpy()
is_tumor = df["y"].to_numpy() == 1

plt.figure(figsize=(7,5))
plt.scatter(xpos[~is_tumor], yvals[~is_tumor], s=30, alpha=0.85,
c="blue", label="tumor_free (0)")
plt.scatter(xpos[ is_tumor], yvals[ is_tumor], s=30, alpha=0.85,
c="red",  label="tumor (1)")
if np.isfinite(egfr_cut):
    plt.axhline(egfr_cut, linestyle="--", linewidth=1.5,
color="black", label="model p=0.5 cutoff")
```

```
plt.xticks([0, 1], ["tumor_free (0)", "tumor (1)"])
plt.xlabel("Class")
plt.ylabel("FGFR1 expression (log2 TPM)")
plt.title("FGFR1 by tumor status (STAD) — red=tumor, blue=tumor_free\
n+ LogisticRegression decision cutoff (p=0.5)")
plt.legend(loc="best")
plt.tight_layout()
plt.show()

# 8) ROC curve on the test split (threshold-free performance view)
RocCurveDisplay.from_predictions(y_te, p_te)
plt.title("ROC — Logistic Regression on FGFR1 (STAD) [test set]")
plt.show()
```

```
Test metrics (FGFR1-only):
  Accuracy : 0.786
  Precision: 0.571
  Recall   : 1.000
  F1       : 0.727
  AUC      : 0.800

Decision threshold on FGFR1 (p=0.5): 4.3738 (raw FGFR1 units)
```





Verify and validate your analysis:

- We mainly used an ROC/AUC curve to validate our results, however other metrics such as accuracy, precision, recall, and an F1 score were also used as supplemental metrics. For the ROC/AUC curve, the goal is to achieve a curve that is "above" the random curve. This measures how will the model sperates tumors from non-tumors at all possible thresholds, and shows the tradeoff between the true positive rate and the false positive rate. As the goal is to have a curve that is high in true positive rate (1.0) and low in the false positive rate (0.0), the most effective models will have curves in the upper left of the graph. In our code, we generated three AUC/ROC curves. Our first curve had a classifier of AUC=0.4, and was below the random curve. This indicates that the model is doing worse than random using the EGFR gene as a predictor. Due to this, we looked at two other genes involved in evading apoptosis, FLT3 and FGFR1. Both genes showed improvement in the model using this metric. The FLT3 gene has above the random curve with classifier 0.72 and vloser to the upper left of the graph, and the FGFR1 model improved upon this still by moving farther up and having a classifier of 0.8.2
- Other metrics using the train_test_split model were using in our code as well. These included accuracy (the overall fraction of correct preductions), precision (of all of the predicted positive values, how many were tru positive values?), recall (of the actual positive values, how many did the model correctly identify as postive values?), and the f1 score (a harmonic mean of precision and recall that only gives a high value when both values are high). In the EGFR gene model, these metrics as listed were 0.357, 0.143, 0.250, and 0.182, respectively. These metrics incresed in the FLT3 model, with values of

0.643, 0.400, 0.500, and 0.444 respectively. In this model, it can be seen that recall (0.500) is higher than precision (0.400), which shows a tradeoff of certain models. In the last model, which used the FGFR1 gene, the values were 0.786, 0.571, 1.000, and 0.727. While precision did not increase much, recall was 100%, which shows that of the true tumor values, the model recalled 100% of them. Using all of these metrics, the model which used FGFR1 as the predictor seems to be the best.

- The impact of the FGFR1 gene on cancer is supported by the literature. Having multiple copies of this gene can lead to uncontrolled cell growth. Additionally, amplification of this gene is closely associated with tobacco or alcohol related cancers. As these are risk factors for stomach adenocarcinoma, this may explain why expression of FGFR1 is a better predictor for cancer agressiveness in stomach adenocarcinoma in our machine learning model. High levels of FGFR1 have been linked to several other carcinomas, such as laryngeal squamous cell carcinoma, and typically present a poor prognosis, again associated with more aggressive cancer types (https://pmc.ncbi.nlm.nih.gov/articles/PMC11593329/). While EGFR overexpression has been shown to negatively impact the prognosis and survival in gastic cancers, and drugs targeting this biomarker has improved patient outcomes, it is possible that the machine learning model did not show this relationship becuase tumor presence was used as the class. Because EGFR has been linked to more advanced gastric cancers, perhaps a different class type may show this relationship (https://pmc.ncbi.nlm.nih.gov/articles/PMC7418523/). FLT3, on the other hand, seems to be underesearched in its effects on stomach cancer; in cancerous cells there was found to be only a 1.6% amplification of the FLT3 gene. This corroborates FLT3 as a poor classifier in our model (https://pmc.ncbi.nlm.nih.gov/articles/PMC5356878/).

Conclusions and Ethical Implications:

- In conclusion, our machine learning model of binary classification with logistic regression was effective in classifying stomach cancer samples by how aggressive/progressed they are when using with tumor/without tumor as a classifier and the FGFR1 gene as a predictor. FGFR1 (ROC/AUC = 0.80) expression was the most effective predictor when compared to EGFR (ROC/AUC = 0.40) and FLT3 (ROC/AUC = 0.72). Along side the ROC curve, this conclusion was validated based on precision and recall scores, F1 scores, and accuracy. It should be noted that recall was especially high for the model that utilized FGFR1 as the predictor, with a recall score of 1.00. Though the model that used FGFR1 was not 100% accurate or precise, the validation methods still show that a model can be made using a single gene as a predictor to be mostly effective means of predicting if a cancer will be aggressive or not.
- Some ethical considerations to make before wider application of this model are that there is a risk of misclassification. An ideal model has an AUC score of 1.00, while our model has a score of 0.80. While this does show that our model is a pretty good predictor, it will not always be 100% accurate and may still output false negative/positives. In a clinical setting, misdiagnoses due to this room for error may drastically impact a patient. Additionally, the dataset may not have fully been representative of all populations. It is recommended that a larger dataset may be tested on the model in order to ensure that the model works across all race, gender, age groups, etc. When applied, the model should also ensure a way to be accessible to all patients, which this includes ensuring that patient data is private.

Limitations and Future Work:

- One limitation is that our model may have overly simplified the problem at hand. As discussed in our background, there are multiple genes that contribute to a cell evading apoptosis, and even within one gene such as FGFR1, there are multiple anit-apoptotic pathways. The class labels that we used (with tumor/tumor free) also overly simplify the cancer progression, as it does not take into account factors such as the cancer stage or if it is metastatic. A more effective model may consider these factors and use a different machine learning method such as a decision tree classifier. As previously discussed, we were also constrained by sample size, and a more effective model may include a more comprehensive and diverse data pool.
- For future work, it is recommended that a model utilizing multiple genes is used in order to better understand how impactful evading apoptosis is as a cancer hallmark. Similarly, more expansive tumor labels may be used, describing more complex categories such as cancer stage. Such a model would need to use a decision tree classifier or another non-binary machine learning method. This project may also be developed on by performing classification methods on different pathways. This may be done by using the expression of different proteins as a predictor. For instance, certain pathways activated by FGFR1 increase BCL-2 production while other inhibit apoptotic proteins. Using these as predictors instead may help us achieve a better understanding of which pathways contribute the most to stomach cancer progression. Finally, this machine learning method may be used to determine if these genes are effective biomarkers for other types of cancer. Because it was found that FGFR1 was especially linked to tobacco and alcohol related cancers, its effect on types such as lung cancer may show interesting relationships.

NOTES FROM YOUR TEAM:

- Need to apply different genes to machine learning method in order to achieve a more effective model
- Need to modify project goal in order to show better understanding of cancer type.

QUESTIONS FOR YOUR TA:

These are questions we have for our TA.