

Things to fix

Chain of Thought

- Bio_clinical BERT embeddings, training, classification

LLM Tutorial

Mohsinuddin Ansari

Table of Contents

1. Synthetic Patient Data Generation

- Use ChatGPT+ and Python to generate 50 synthetic patient profiles (40 training/5 validation/5 test)
- Assign ground truths by using ChatGPT+ (GPT4-turbo) to be the "expert" to label risk for each of these patients for cardiovascular diseases, endocrine disorders, and respiratory diseases

2. In-Context Learning

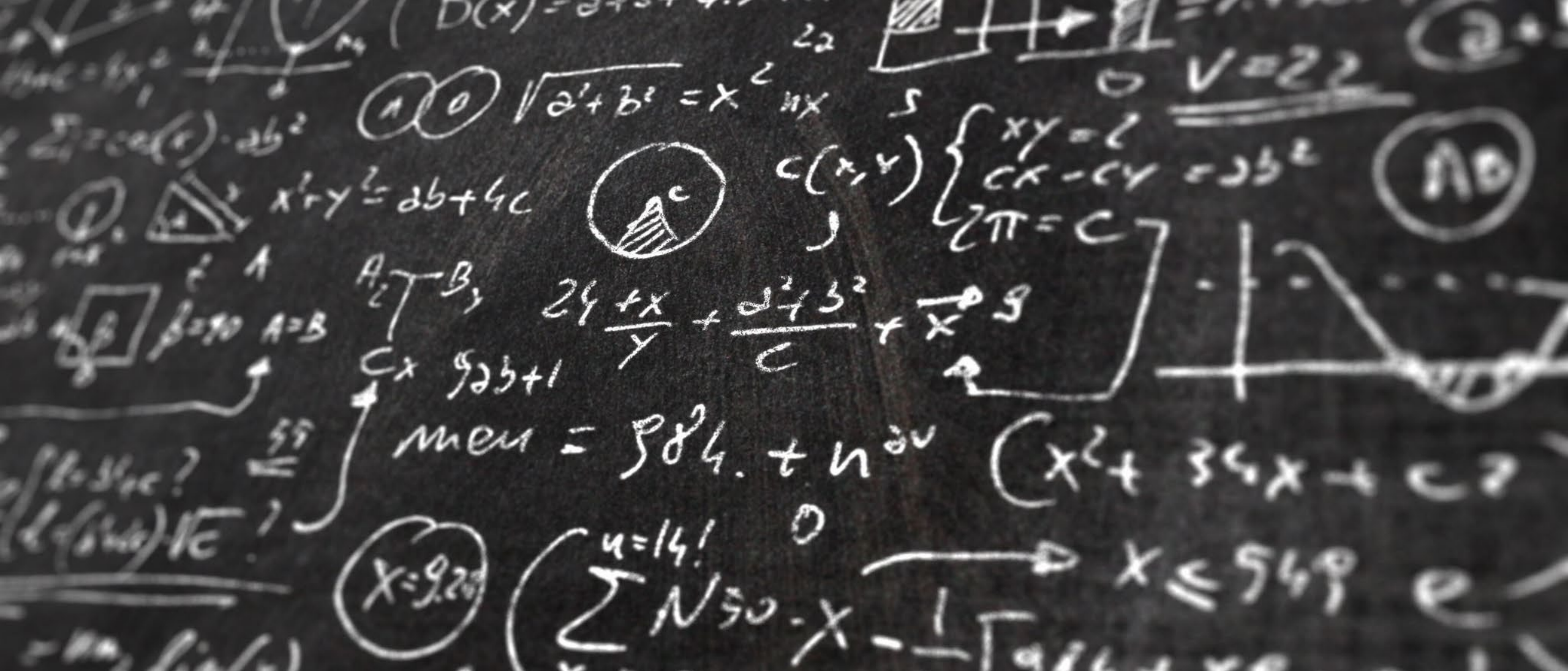
- Use OpenAI API (Chat3.5-turbo) for in-context learning on test dataset

3. Few-Shot Learning

- Use OpenAI API (Chat3.5-turbo) by providing training patients ground truths to learn test patients risks and compare to In-Context output

4. Chain-of-Thought

5. Bio_Clinical BERT Embeddings



Synthetic Patient Data Generation

Generate Risk Factors with ChatGPT+ (GPT4-turbo)

-
- The process of creating a patient template and generating synthetic patient data involves several steps leading to simulating realistic patient profiles that can be used for AI training and testing.
 - **1. Identification of Key Factors**
 - The first step involves determining the critical factors that influence the risk of specific diseases, such as cardiovascular disease (CVD), diabetes (endocrine disorders), and respiratory diseases.
 - I used ChatGPT+ to identify the top five factors for each disease category. For example:
 - **For CVD:** Blood pressure, cholesterol levels, smoking status, age, and body mass index (BMI).
 - **For Diabetes:** Blood glucose levels, family history of diabetes, age, obesity, and physical activity level.
 - **For Respiratory Diseases:** Smoking status, air quality exposure, age, history of respiratory infections, and genetic predispositions.
 - **2. Consolidation of Factors**
 - After identifying these factors, consolidate them into a list, both numerical and categorical. Merge overlapping factors (like age and smoking status, common to multiple diseases) and select the most impactful variables based on their relevance to each disease.
 - We use ChatGPT+ to create Python code with appropriate probability distributions for each factor.
 - We also generate a random list of comorbidities.

Generate Synthetic Patient Data with Python

```
def generate_realistic_patient_data(patient_id):  
  
    # Include gender  
    gender = np.random.choice(["Male", "Female"])  
  
    # Age is normally distributed around the mean age of 50 with a  
    standard deviation of 12  
    age = int(np.random.normal(50, 12))  
    age = max(18, min(age, 85)) # Limit age to realistic bounds  
  
    # Blood Pressure adjustments based on gender  
    if gender == "Male":  
        systolic_bp = int(np.random.normal(125, 15))  
        diastolic_bp = int(np.random.normal(80, 10))  
    else:  
        systolic_bp = int(np.random.normal(120, 15))  
        diastolic_bp = int(np.random.normal(75, 10))  
  
    # A1c levels adjusted for typical distribution in the adult  
    population  
    a1c = round(np.random.normal(5.5, 0.75), 1)  
    a1c = max(4.0, min(a1c, 10.0)) # Limit A1c to realistic bounds  
  
    # Cholesterol levels adjusted based on gender  
    total_cholesterol = int(np.random.normal(200, 40))  
    ldl_cholesterol = int(np.random.normal(115, 30))  
    hdl_cholesterol = int(np.random.normal(60 if gender == "Female"  
else 50, 15))  
  
    # BMI adjusted for more common distribution  
    bmi = round(np.random.normal(27, 5), 1)  
    bmi = max(15, min(bmi, 45)) # Limit BMI to realistic bounds  
  
    # Pulmonary Function FEV1/FVC percentage  
    pft_fev1_fvc = round(np.random.normal(80, 10), 1)  
    pft_fev1_fvc = max(50, min(pft_fev1_fvc, 100)) # Limit to  
    realistic bounds  
    patient_comorbidities = assign_comorbidities()  
    return {  
        "PatientID": patient_id,  
        "Gender": gender,  
        "Age": age,  
        "Blood Pressure": (systolic_bp, diastolic_bp),  
        "A1c Levels": a1c,  
        "Total Cholesterol": total_cholesterol,  
        "LDL Cholesterol": ldl_cholesterol,  
        "HDL Cholesterol": hdl_cholesterol,  
        "Smoking Status": np.random.choice(["Current smoker", "Former  
smoker", "Never smoked"], p=[0.2, 0.3, 0.5]),  
        "BMI": bmi,  
        "Physical Activity Level": np.random.choice(["High",  
"Moderate", "Low"], p=[0.2, 0.5, 0.3]),  
        "Exposure to Pollutants": np.random.choice(["High",  
"Moderate", "Low"], p=[0.1, 0.3, 0.6]),  
        "Family History of Chronic Diseases":  
np.random.choice(["Present", "Absent"], p=[0.3, 0.7]),  
        "Ethnicity": np.random.choice(["African American",  
"Hispanic/Latino", "Native American", "Asian American", "Pacific  
Islander", "Caucasian"], p=[0.13, 0.18, 0.05, 0.05, 0.02, 0.57]),  
        "Pulmonary Function Tests (FEV1/FVC)": pft_fev1_fvc,  
        "Comorbidities": patient_comorbidities  
    }
```

Generate Random Comorbidities to Enhance Patient Profiles

```
comorbidities = {  
    "Neurological Disorders": ["Stroke", "Parkinson's disease", "Multiple sclerosis", "Epilepsy"],  
    "Psychiatric Conditions": ["Depression", "Anxiety disorders", "Bipolar disorder", "Schizophrenia"],  
    "Renal Disorders": ["Chronic kidney disease", "Acute kidney injury", "Nephrolithiasis"],  
    "Gastrointestinal Diseases": ["Chronic liver disease", "Inflammatory bowel disease", "GERD"],  
    "Musculoskeletal Disorders": ["Osteoarthritis", "Rheumatoid arthritis", "Osteoporosis"],  
    "Cancer": ["Lung cancer", "Breast cancer", "Prostate cancer", "Colorectal cancer"],  
    "Metabolic Disorders": ["Metabolic syndrome", "Obesity"]  
}  
  
def assign_comorbidities():  
    # Choose how many comorbidities (0 to 3)  
    num_comorbidities = np.random.choice([0, 1, 2, 3], p=[0.5, 0.3, 0.15, 0.05])  
    selected_comorbidities = []  
  
    if num_comorbidities > 0:  
        # Flatten and shuffle the list of possible comorbidities  
        all_comorbidities = [item for sublist in comorbidities.values() for item in sublist]  
        np.random.shuffle(all_comorbidities)  
  
        # Select the first 'num_comorbidities' from the shuffled list  
        selected_comorbidities = all_comorbidities[:num_comorbidities]  
  
    return selected_comorbidities
```

Generate Random Comorbidities to Enhance Patient Profiles

Using the consolidated list of factors, you asked ChatGPT+ to create Python code capable of generating distributions for these variables. This code uses statistical distributions to assign values to each factor for each synthetic patient, ensuring that the resulting profiles reflect realistic variations found in the general population.

- **Numerical variables** are generated by normal or log-normal distributions from ChatGPT+ depending on the variable type (e.g., BMI might be normally distributed, while something like blood glucose levels might have a skewed distribution).

- **Categorical variables** are generated based on probabilities ChatGPT+ derived from typical population demographics (e.g., the likelihood of being a smoker).

- We then use this Python code to create 50 synthetic patient profiles, presumably ensuring a diverse representation across the spectrum of each disease risk. (40 Training/5 Validation/5 Test)

```
{'PatientID': 17,  
  'Gender': 'Male',  
  'Age': 53,  
  'Blood Pressure': (132, 86),  
  'A1c Levels': 5.3,  
  'Total Cholesterol': 228,  
  'LDL Cholesterol': 133,  
  'HDL Cholesterol': 74,  
  'Smoking Status': 'Never smoked',  
  'BMI': 21.1,  
  'Physical Activity Level': 'Moderate',  
  'Exposure to Pollutants': 'Low',  
  'Family History of Chronic Diseases': 'Present',  
  'Ethnicity': 'Caucasian',  
  'Pulmonary Function Tests (FEV1/FVC)': 94.4,  
  'Comorbidities': ['Osteoarthritis']}
```


Generate Patient Disease Risk “Ground Truths” using GPT4-turbo as the “Expert”

4. Split into Training, Validation and Test Data Sets

You then used this Python code to create 50 synthetic patient profiles, presumably ensuring a diverse representation across the spectrum of each disease risk. The distribution of these profiles (40/5/5) suggests a focus on creating a larger base dataset with smaller subsets for more detailed analysis.

5. Generating Ground Truth Data

Prompt to generate 'ground truth' risk scores for each patient. This involves assessing each synthetic profile and assigning a risk level from 1 (low risk) to 5 (high risk) across the three disease categories. This step is crucial for training predictive models, as it provides a benchmark against which AI predictions can be compared.

Prompt used in ChatGPT+ (GPT4-turbo) for Ground Truths

```
prompt += """
    \nYou are an experienced internal medicine
specialist. Perform each of the following steps to
come to final conclusion on each patient. Do not
give explanations. Only your final conclusion.
Analyze how the patient's ethnicity and lifestyle
factors affect their risk of each of the
following: cardiovascular diseases, endocrine
disorders, and respiratory diseases.
Consider the patient's medical and family history
to evaluate further risk factors for each.
Interpret the patient's biomarker levels in the
context of risk to each disease.
Integrate the above analyses to assess the overall
risk of each disease for this patient and recommend
steps for risk reduction.
Only respond with the patient number and a rating
of 1 to 5 for each group of diseases for each of
the patients. 1 is low risk.
To assess the risk of cardiovascular diseases,
endocrine disorders, and respiratory diseases for
each patient in the dataset on a scale from 1 to 5,
consider several factors available from the data,
including:

Cardiovascular Risk:
Gender
Blood pressure levels (hypertension risk)
Total cholesterol and LDL cholesterol levels
(atherosclerosis risk)
Smoking status (increased risk for smokers)
BMI (obesity can increase risk)
Family history (genetic predisposition)

Endocrine Disorders Risk:
Gender
Alc levels (indicative of glucose metabolism issues,
potential diabetes)
Family history of chronic diseases like diabetes
BMI (as it relates to metabolic syndrome)

Respiratory Diseases Risk:
Gender
Smoking status (major risk factor for COPD and lung cancer)
Pulmonary function tests (FEV1/FVC ratios indicating
obstructive or restrictive lung disease)
Exposure to pollutants (potential risk factor for
respiratory conditions)
Comorbidities such as chronic lung diseases. """
```

In-Context Learning

In-Context Learning with Same Prompt in Open AI (GPT3.5-turbo) to Generate “Classifications” (Test Data Output Next Slide)

```
prompt += """
```

```
    \nYou are an experienced internal medicine specialist. Perform each of the following steps to come to final conclusion on each patient. Do not give explanations. Only your final conclusion. Analyze how the patient's ethnicity and lifestyle factors affect their risk of each of the following: cardiovascular diseases, endocrine disorders, and respiratory diseases. Consider the patient's medical and family history to evaluate further risk factors for each. Interpret the patient's biomarker levels in the context of risk to each disease. Integrate the above analyses to assess the overall risk of each disease for this patient and recommend steps for risk reduction. Only respond with the patient number and a rating of 1 to 5 for each group of diseases for each of the patients. 1 is low risk. To assess the risk of cardiovascular diseases, endocrine disorders, and respiratory diseases for each patient in the dataset on a scale from 1 to 5, consider several factors available from the data, including:
```

```
Cardiovascular Risk:
```

```
Gender
```

```
Blood pressure levels (hypertension risk)
```

```
Total cholesterol and LDL cholesterol levels (atherosclerosis risk)
```

```
Smoking status (increased risk for smokers)
```

```
BMI (obesity can increase risk)
```

```
Family history (genetic predisposition)
```

```
Endocrine Disorders Risk:
```

```
Gender
```

```
A1c levels (indicative of glucose metabolism issues, potential diabetes)
```

```
Family history of chronic diseases like diabetes
```

```
BMI (as it relates to metabolic syndrome)
```

```
Respiratory Diseases Risk:
```

```
Gender
```

```
Smoking status (major risk factor for COPD and lung cancer)
```

```
Pulmonary function tests (FEV1/FVC ratios indicating obstructive or restrictive lung disease)
```

```
Exposure to pollutants (potential risk factor for respiratory conditions)
```

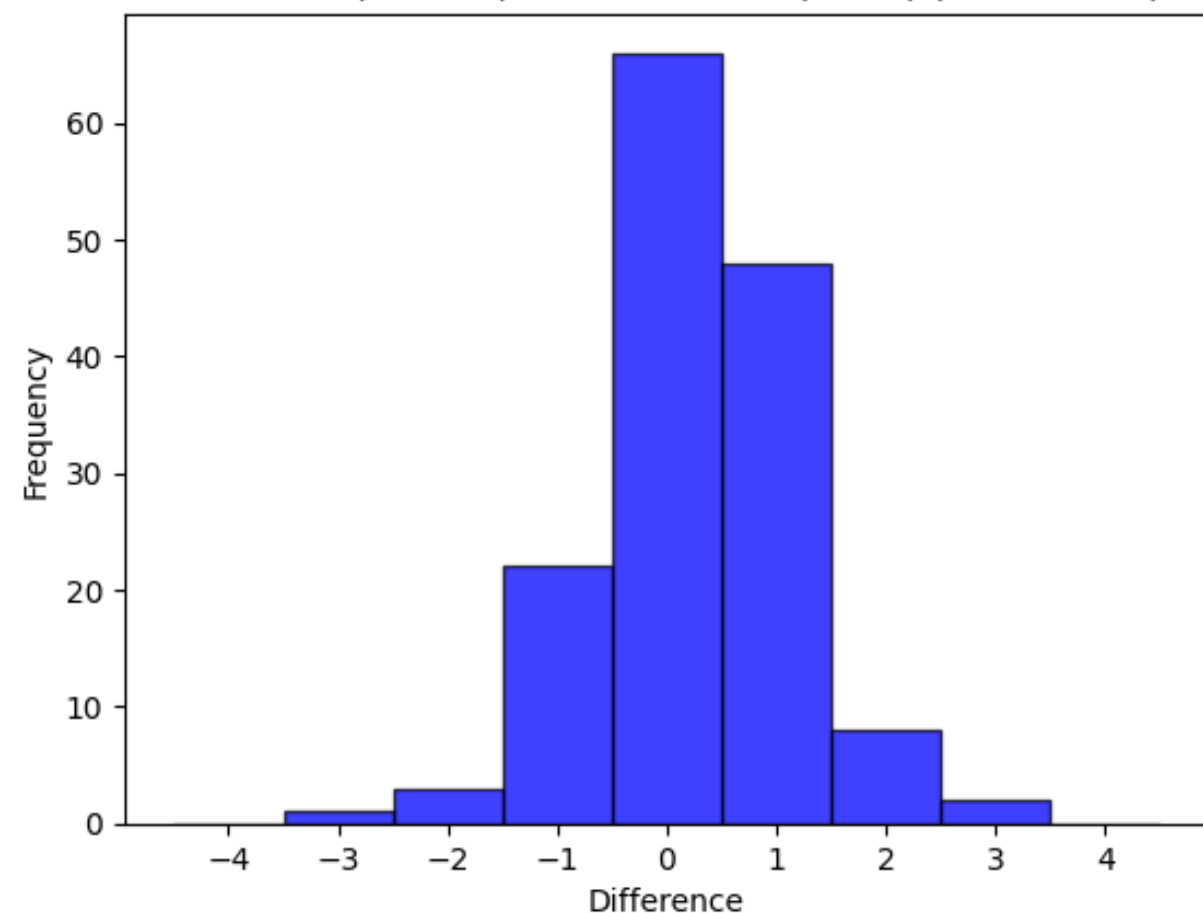
```
Comorbidities such as chronic lung diseases. """
```

In-Context Learning “Classifications” on Test Data

Patient 46:	Cardiovascular Risk: 3	Endocrine Disorders Risk: 2	Respiratory Diseases Risk: 2
Patient 47:	Cardiovascular Risk: 3	Endocrine Disorders Risk: 3	Respiratory Diseases Risk: 2
Patient 48:	Cardiovascular Risk: 3	Endocrine Disorders Risk: 2	Respiratory Diseases Risk: 3
Patient 49:	- Cardiovascular Risk: 3	- Endocrine Disorders Risk: 3	- Respiratory Diseases Risk: 2
Patient 50:	- Cardiovascular Diseases: 4	- Endocrine Disorders: 3	- Respiratory Diseases: 5

Difference Between Ground Truths and Predictions (All 50 Patients)

Predicted (GPT3.5) - Ground Truth (GPT4) (50 Patients)



The skew suggests that GPT3.5 assigns higher risk values to the 50 patients on average than GPT4 does

For the entire presentation:

Training Set – Patients 1-40

Validation Set – Patients 41-45

Test Set – Patients 46-50

More In-Context Learning on Test Data – Added to the end of the previous prompt (Output on next page)

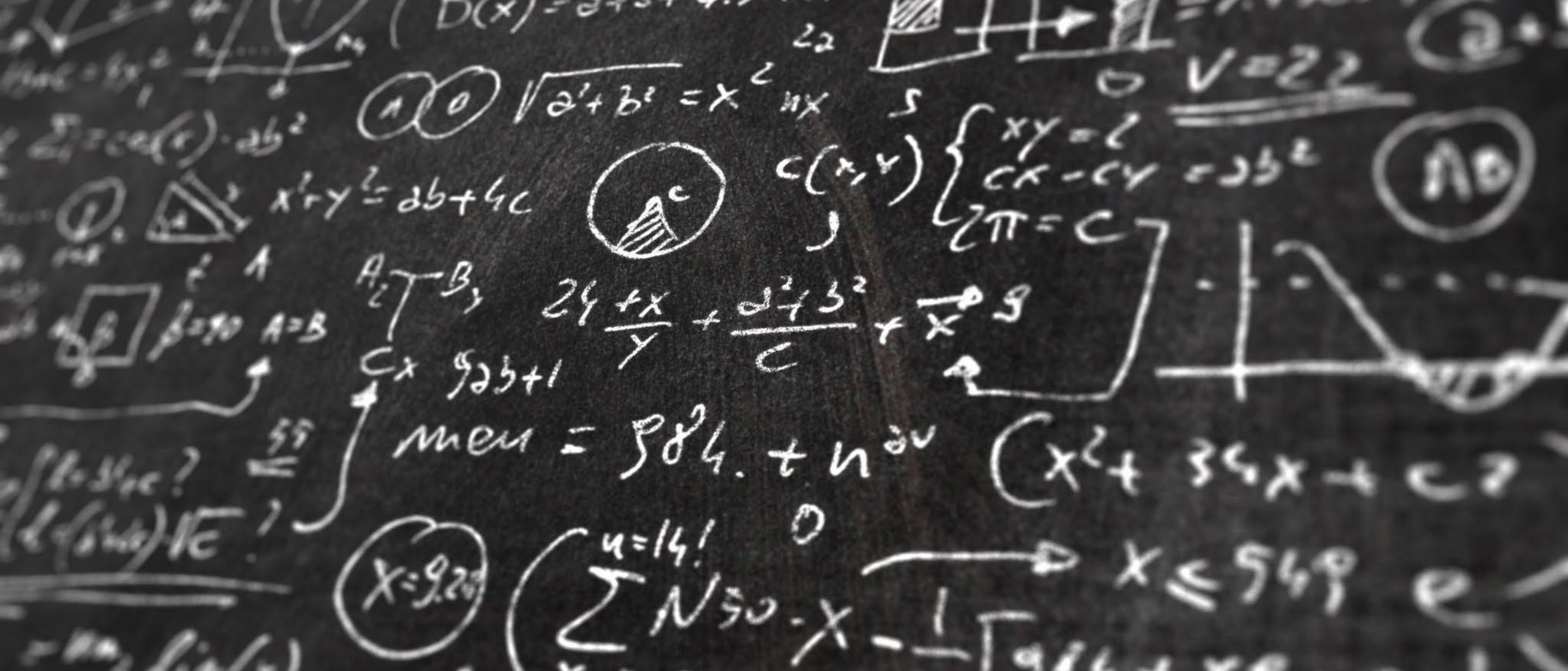
Now assume that each patient exercises 30 minutes a day for a year. What are your new predictions for each patient. Also, for each patient give two suggestions specific to their profile and how that could benefit their risk factors.

Evaluation of the results on next page:

- The recommendations are generic for each patient
 - Keep exercising
 - Improve diet
- Predicted Risk Factors after one year of exercise consistently decrease by one point for each patient for each disease

Impact of Exercise on Risk Factors + Two “Doctor” Recommendations from GPT3.5-turbo

- Patient 46: Cardiovascular Risk: 3 Endocrine Disorders Risk: 2 Respiratory Diseases Risk: 3 New Prediction
 - **After Exercising**: Cardiovascular Risk: 2 Endocrine Disorders Risk: 2 Respiratory Diseases Risk: 2
 - Suggestions: 1. Increase physical activity level from low to moderate intensity to further reduce cardiovascular and respiratory disease risk. 2. Consider consulting with a nutritionist to optimize diet and potentially lower cholesterol levels.
- Patient 47: Cardiovascular Risk: 3 Endocrine Disorders Risk: 4 Respiratory Diseases Risk: 2
 - **New Predictions after 1 year of exercise**: Cardiovascular Risk: 2 Endocrine Disorders Risk: 3 Respiratory Diseases Risk: 1
 - Suggestions: 1. Continue moderate physical activity as it can help improve overall cardiovascular health. 2. Maintain a healthy weight to reduce the risk of endocrine disorders like diabetes. Final Conclusion: Overall risk assessment: Moderate risk for cardiovascular diseases, higher risk for endocrine disorders, and low risk for respiratory diseases. Encourage continued physical activity and weight management to reduce risks.
- Patient 48: Cardiovascular Risk: 4 Endocrine Disorders Risk: 3 Respiratory Diseases Risk: 3
 - **New Predictions after 1 year of exercise**: Cardiovascular Risk: 3 Endocrine Disorders Risk: 2 Respiratory Diseases Risk: 2
 - Suggestions: 1. Focus on maintaining a healthy BMI through regular exercise and balanced diet. 2. Consider monitoring blood pressure levels regularly to track cardiovascular health.
- Patient 49: Cardiovascular Risk: 3 Endocrine Disorders Risk: 4 Respiratory Diseases Risk: 2
 - **New Predictions after a year of exercise**: - Cardiovascular Risk: 2 - Endocrine Disorders Risk: 3 - Respiratory Diseases Risk: 1
 - Suggestions: 1. Maintain a healthy diet rich in fruits, vegetables, and whole grains to help manage A1c levels and cholesterol. 2. Consider regular monitoring of blood pressure and cholesterol levels to track improvements.
- Patient 50: Cardiovascular Diseases Risk: 3 Endocrine Disorders Risk: 2 Respiratory Diseases Risk: 4
 - **New Predictions after exercising 30 minutes a day for a year**: Cardiovascular Diseases Risk: 2 Endocrine Disorders Risk: 1 Respiratory Diseases Risk: 3
 - Suggestions: 1. Consider incorporating more high-intensity interval training (HIIT) into the exercise routine to



Few-Shot Learning

Few-Shot Learning (5 Shots)

- We provide GPT3.5-turbo with five training patient profiles and their GPT4-turbo ground truth risk values and ask it to classify the risk values for the test patients.

****Risk Assessment for Cardiovascular, Endocrine, and Respiratory Diseases for the Last Five Patients:****

- Patient 46: - Cardiovascular: 3 - Endocrine: 2 - Respiratory: 2 - Analysis: Moderate risk for cardiovascular and endocrine diseases, low risk for respiratory diseases.
- Patient 47: - Cardiovascular: 4 - Endocrine: 3 - Respiratory: 2 - Analysis: High risk for cardiovascular diseases, moderate risk for endocrine diseases, and low risk for respiratory diseases.
- Patient 48: - Cardiovascular: 3 - Endocrine: 3 - Respiratory: 2 - Analysis: Moderate risk for cardiovascular and endocrine diseases, low risk for respiratory diseases.
- Patient 49: - Cardiovascular: 4 - Endocrine: 4 - Respiratory: 3 - Analysis: High risk for cardiovascular and endocrine diseases, moderate risk for respiratory diseases.
- Patient 50: - Cardiovascular: 3 - Endocrine: 4 - Respiratory: 4 - Analysis: Moderate risk for cardiovascular diseases, high risk for endocrine diseases, and high risk for respiratory diseases.

Based on the patient profiles and risk assessments, it is essential for the last five patients to focus on lifestyle modifications, regular monitoring, and potentially consider preventive measures to reduce the risk of developing cardiovascular, endocrine, and respiratory diseases. Regular follow-ups and consultations with healthcare providers are recommended to manage and mitigate these risks effectively.

Few-Shot Learning Results (5 Shots)

• In-Context, Few Shot, and Ground Truth Test Classifications

Patient	In-Context Cardiovascu lar	In-Context Endocrine	In-Context Respiratory	Few-Shot Cardiovascu lar	Few-Shot Endocrine	Few-Shot Respiratory	Ground Truth Cardiovascu lar	Ground Truth Endocrine	Ground Truth Respiratory
46	3	2	2	3	2	2	4	2	2
47	3	3	2	4	3	2	3	5	2
48	3	2	3	3	3	2	2	3	3
49	3	3	2	4	4	3	3	5	2
50	4	3	5	3	4	4	2	1	4

Few-Shot Learning Analysis (5 Shots)

General Observations

- There are **consistencies** within the AI-based predictions for respiratory diseases, with fewer discrepancies between predicted and actual values, suggesting that the factors influencing respiratory risk might be better represented or understood in the data used for modeling.
- The **variance in cardiovascular and endocrine predictions** could point to a need for better integration or representation of influencing factors in the AI models or perhaps a more nuanced understanding of how these factors play out differently in individual patient profiles.
- Few-Shot Learning seems to provide a slightly more accurate or consistent alignment with Ground Truth values, likely due to the priming effect of the examples included in the prompts. However, it still shows room for improvement, especially in capturing extreme risk values.

Few-Shot Learning Results (40 Shots)

• In-Context, Few Shot, and Ground Truth Test Classifications

Patient	5-Shot Cardiovascular	5-Shot Endocrine	5-Shot Respiratory	40-Shot Cardiovascular	40-Shot Endocrine	40-Shot Respiratory	Ground Truth Cardiovascular	Ground Truth Endocrine	Ground Truth Respiratory
46	3	2	2	2	1	1	4	2	2
47	4	3	2	3	3	2	3	5	2
48	3	3	2	3	2	2	2	3	3
49	4	4	3	4	3	2	3	5	2
50	3	4	4	3	2	5	2	1	4

Few-Shot Learning Analysis (40 Shots)

Analysis and Conclusion for the Last Five Patients Based on Their Profiles and Risk Assessment:

- **Patient 46:** 40-Shot Learning data suggest lower risks than previously indicated and the Ground Truth. This could indicate that with more training examples, the model's predictions become more conservative or better aligned with certain aspects of the patient's profile.
- **Patient 47:** The moderate risk assessment for cardiovascular and endocrine disorders aligns more closely between the 40-Shot Learning results and the Ground Truth compared to previous predictions, which suggested higher cardiovascular risk.
- **Patient 48:** Both Few-Shot Learning predictions show consistency in respiratory risk assessments and a better alignment with the Ground Truth for endocrine risk using the expanded training data.
- **Patient 49:** While cardiovascular risk remains consistently high in AI assessments, the 40-Shot Learning results adjust the endocrine risk closer to moderate, aligning slightly better with the Ground Truth.
- **Patient 50:** Notably, respiratory risk predictions vary significantly, with the 40-Shot model highlighting a high risk that aligns with the Ground Truth, potentially identifying specific risk factors more effectively with more training data.

Overall Conclusion:

The incorporation of more examples in Few-Shot Learning seems inconclusive with 5-shot being closer to ground truth sometimes and 40-shot being closer at other times.

The background image shows a blackboard covered in various mathematical notations and sketches. Visible elements include:

- Equations: $\sqrt{a^2 + b^2} = x^2 \ln x$, $x^2 + y^2 = ab + 4c$, $\frac{24+x}{y} + \frac{a^2+b^2}{c} + \frac{1}{x} = 9$, $\frac{1}{x} \sqrt{984 + x^2} = 9$, $x \leq 549$, $\sum N_{30} \cdot x$, $x = 9.20$, $x = 14!$, $x^2 + 34x + c = 0$, $\frac{1}{x} \sqrt{984 + x^2} = 9$.
- Diagrams: A triangle with a shaded area, a circle with a shaded sector, a coordinate plane with a curve, and a graph of a function.
- Symbols: π , ∞ , $\frac{1}{x}$, $\frac{1}{y}$, $\frac{1}{c}$, $\frac{1}{x}$, $\frac{1}{y}$, $\frac{1}{c}$, $\frac{1}{x}$, $\frac{1}{y}$, $\frac{1}{c}$.

Comparative Analysis: Chain-of-Thought on Test Data

You are an experienced internal medicine specialist. State, among all the patients' data I included above (Patients 46-50), rank the patients in order of most at risk to least at risk for each of the following and state your chain-of-thought as to why you chose that patient order: cardiovascular diseases, endocrine disorders, and respiratory diseases.

To rank the patients in order of most at risk to least at risk for each condition based on the provided data:

- 1. Cardiovascular Diseases:
 - Patient 49: This patient has the highest A1c levels (7.5) and total cholesterol (264) among the group. They are also a former smoker with moderate physical activity levels and a family history of obesity. These factors increase the risk of cardiovascular diseases.
 - Patient 47: This patient has elevated A1c levels (7.2) and total cholesterol (233), as well as a history of smoking. While their physical activity level is moderate, their cholesterol levels and smoking history put them at increased risk.
 - Patient 50: Although this patient has high LDL cholesterol (183), their A1c levels (4.5) are within the normal range, and they have a lower total cholesterol level compared to the other patients. However, the history of lung cancer could potentially impact cardiovascular health.
 - Patient 46: This patient has the lowest A1c levels (5.3) and total cholesterol (178) among the group. They have never smoked, have a normal BMI, and a family history of chronic diseases is absent, all of which are protective factors against cardiovascular diseases.
 - Patient 48: This patient has relatively good lipid profile and A1c levels, with a normal BMI, high physical activity level, and no family history of chronic diseases. However, the exposure to high pollutants may increase cardiovascular risk.

Comparative Analysis: Chain-of-Thought on Test Data

- 2. Endocrine Disorders:
 - Patient 49: With a higher A1c level (7.5) and a history of obesity, this patient is at the highest risk for endocrine disorders like diabetes.
 - Patient 47: This patient also has elevated A1c levels (7.2) and a history of smoking, which can contribute to the risk of endocrine disorders.
 - Patient 48: While this patient has a relatively good A1c level (5.5) and no family history of chronic diseases, the presence of bipolar disorder may indicate underlying endocrine issues that could increase the risk.
 - Patient 46: This patient has a normal A1c level (5.3) and no other significant risk factors for endocrine disorders.
 - Patient 50: With the lowest A1c level (4.5) among the group, this patient appears to be at the lowest risk for endocrine disorders.
- 3. Respiratory Diseases:
 - Patient 50: This patient has the lowest FEV1/FVC ratio (69.0), indicating potential respiratory issues. Additionally, the history of lung cancer puts them at the highest risk for respiratory diseases.
 - Patient 47: With a lower FEV1/FVC ratio (74.4) and a history of smoking, this patient is also at increased risk for respiratory diseases.
 - Patient 49: This patient has a lower FEV1/FVC ratio (75.1) and a history of obesity, which can impact respiratory function and increase the risk of respiratory diseases.
 - Patient 48: While this patient has a relatively good FEV1/FVC ratio (86.4) and no smoking history, the exposure to high pollutants may increase the risk of respiratory diseases.
 - Patient 46: This patient has the highest FEV1/FVC ratio (84.1) among the group and no other significant risk factors for respiratory diseases. Overall, Patient 49 appears to be at the highest risk for cardiovascular diseases and endocrine disorders, while Patient 50 is at the highest risk for respiratory diseases based on

Classification and Regression Models with Bio_ClinicalBERT Embeddings

Install Transformers, Load Patient Profiles, Bio_ClinicalBERT Model

```
!pip install transformers torch numpy pandas
```

```
import json  
import pandas as pd
```

```
# Load the JSON file
```

```
def load_data(filepath):  
    with open(filepath, 'r') as file:  
        data = json.load(file)  
    return data
```

```
# Assuming the JSON structure is a list of patient records
```

```
data = load_data('/content/drive/MyDrive/AI in Healthcare/Assignment  
5/patients_data.json')
```

```
# Convert the data to a DataFrame for easier manipulation
```

```
df = pd.json_normalize(data)
```

```
from transformers import AutoTokenizer, AutoModel
```

```
tokenizer  
= AutoTokenizer.from_pretrained("emilyalsentzer/Bio_ClinicalBERT")
```

```
def get_embeddings(texts):  
    encodings = tokenizer(texts, padding=True, truncation=True,  
max_length=512, return_tensors='pt')  
    with torch.no_grad():  
        outputs = model(**encodings)  
        embeddings = outputs.last_hidden_state[:, 0, :].numpy() # Extract  
embeddings of the [CLS] token  
    return embeddings
```

Create Prompts for Classification and Regression Tasks

```
def prepare_text(row):
    # Format the blood pressure reading
    bp = " and ".join(map(str, row['Blood Pressure']))
    # Handling the list of comorbidities
    if row['Comorbidities']:
        comorbidities = ', '.join(row['Comorbidities'])
    else:
        comorbidities = "No comorbidities reported"
    # Create the summary text
    text = "Use this patient profile to determine on a scale of 1 (low) to 5 (high) the risk to
cardiovascular disease, endocrine disorders, and respiratory diseases."
    text += (f"PatientID {row['PatientID']} is a {row['Age']} year
old {row['Gender']} {row['Ethnicity']} patient: blood pressure {bp}. "
           f"A1c {row['A1c Levels']}%. "
           f"Cholesterol: Total: {row['Total Cholesterol']}, LDL: {row['LDL Cholesterol']},
HDL: {row['HDL Cholesterol']}. "
           f"Smoking status: {row['Smoking Status']}, BMI: {row['BMI']}, Exercise Level: {row['Physical
Activity Level']}. "
           f"Exposure to pollutants: {row['Exposure to Pollutants']}, family history of
chronic diseases: {row['Family History of Chronic Diseases']}. "
           f"FEV1/FVC ratio: {row['Pulmonary Function Tests (FEV1/FVC)]}. "
           f"Comorbidities: {comorbidities}.")
    #print (text)
    return text
```

```
def prepare_text2(row):
    # Format the blood pressure reading
    bp = " and ".join(map(str, row['Blood Pressure']))
    # Handling the list of comorbidities
    if row['Comorbidities']:
        comorbidities = ', '.join(row['Comorbidities'])
    else:
        comorbidities = "No comorbidities reported"
    # Create the summary text
    text = "Use this patient profile to predict A1c level:."
    text += (f"PatientID {row['PatientID']} is a {row['Age']} year
old {row['Gender']} {row['Ethnicity']} patient: blood pressure {bp}. "
           f"Cardiovascular Risk: {row['CVD Risk']}, Endocrine Risk: {row['Endocrine Risk']},
Respiratory Risk: {row['Respiratory Risk']}. 1 means low risk, 5 means high."
           f"Cholesterol: Total: {row['Total Cholesterol']}, LDL: {row['LDL Cholesterol']},
HDL: {row['HDL Cholesterol']}. "
           f"Smoking status: {row['Smoking Status']}, BMI: {row['BMI']}, Exercise Level: {row['Physical
Activity Level']}. "
           f"Exposure to pollutants: {row['Exposure to Pollutants']}, family history of chronic diseases: {row['Family
History of Chronic Diseases']}. "
           f"FEV1/FVC ratio: {row['Pulmonary Function Tests (FEV1/FVC)]}. "
           f"Comorbidities: {comorbidities}.")
    #print (text)
    return text
```

```
# Apply the function to the DataFrame
risk_df = pd.DataFrame(ground_truth, columns=['CVD Risk', 'Endocrine Risk', 'Respiratory Risk'])
df = pd.concat([df, risk_df], axis=1)
df['summary_text'] = ""
df['summary_text'] = df.apply(prepare_text, axis=1)
df['Text_Length'] = df['summary_text'].str.len()
df['summary_text2'] = ""
df['summary_text2'] = df.apply(prepare_text2, axis=1)
df['Text_Length2'] = df['summary_text2'].str.len()
```


Create Train, Validation, and Test Sets

```
import torch

df['embeddings'] = df['summary_text'].apply(lambda x: get_embeddings(x))
df['embeddings2'] = df['summary_text2'].apply(lambda x: get_embeddings(x))

# Assuming 'df' is your main DataFrame and embeddings are correctly set up
train_df = df.iloc[:40] # First 40 records for training
valid_df = df.iloc[40:45] # Next 5 records for validation
test_df = df.iloc[45:50] # Last 5 records for testing
train_features = np.vstack(train_df['embeddings'].values)
train_labels_cvd = train_df['CVD Risk'].values
train_labels_endocrine = train_df['Endocrine Risk'].values
train_labels_respiratory = train_df['Respiratory Risk'].values
train_features2 = np.vstack(train_df['embeddings2'].values)
train_labels_a1c = train_df['A1c Levels'].values.astype(float) # Ensure the A1c levels are in a proper numeric format
valid_features = np.vstack(valid_df['embeddings'].values)
valid_labels_cvd = valid_df['CVD Risk'].values
valid_labels_endocrine = valid_df['Endocrine Risk'].values
valid_labels_respiratory = valid_df['Respiratory Risk'].values
valid_features2 = np.vstack(valid_df['embeddings2'].values)
valid_labels_a1c = valid_df['A1c Levels'].values.astype(float) # Ensure the A1c levels are in a proper numeric format
test_features = np.vstack(test_df['embeddings'].values)
test_labels_cvd = test_df['CVD Risk'].values
test_labels_endocrine = test_df['Endocrine Risk'].values
test_labels_respiratory = test_df['Respiratory Risk'].values
test_features2 = np.vstack(test_df['embeddings2'].values)
```

Fit 3 Disease Classification Models and 1 A1C Regression Model

```
from sklearn.linear_model import LogisticRegression, LinearRegression
from sklearn.preprocessing import StandardScaler
from sklearn.pipeline import make_pipeline

# Define the logistic regression model within a pipeline that includes scaling
model_cvd = make_pipeline(StandardScaler(),
LogisticRegression(multi_class='ovr'))
model_endocrine = make_pipeline(StandardScaler(),
LogisticRegression(multi_class='ovr'))
model_respiratory = make_pipeline(StandardScaler(),
LogisticRegression(multi_class='ovr'))

# Fit models
model_cvd.fit(train_features, train_labels_cvd)
model_endocrine.fit(train_features, train_labels_endocrine)
model_respiratory.fit(train_features, train_labels_respiratory)

# Setting up a pipeline with scaling and linear regression
model_a1c = make_pipeline(StandardScaler(), LinearRegression())

# Training the model
```

A1C Regression Model Results (Not Good Results)

```
from sklearn.metrics import mean_squared_error, r2_score
```

```
# Making predictions on the test set
```

```
labels_pred = model_a1c.predict(test_features2)
```

```
# Calculating the mean squared error and R^2 score
```

```
mse = mean_squared_error(test_labels_a1c, labels_pred)
```

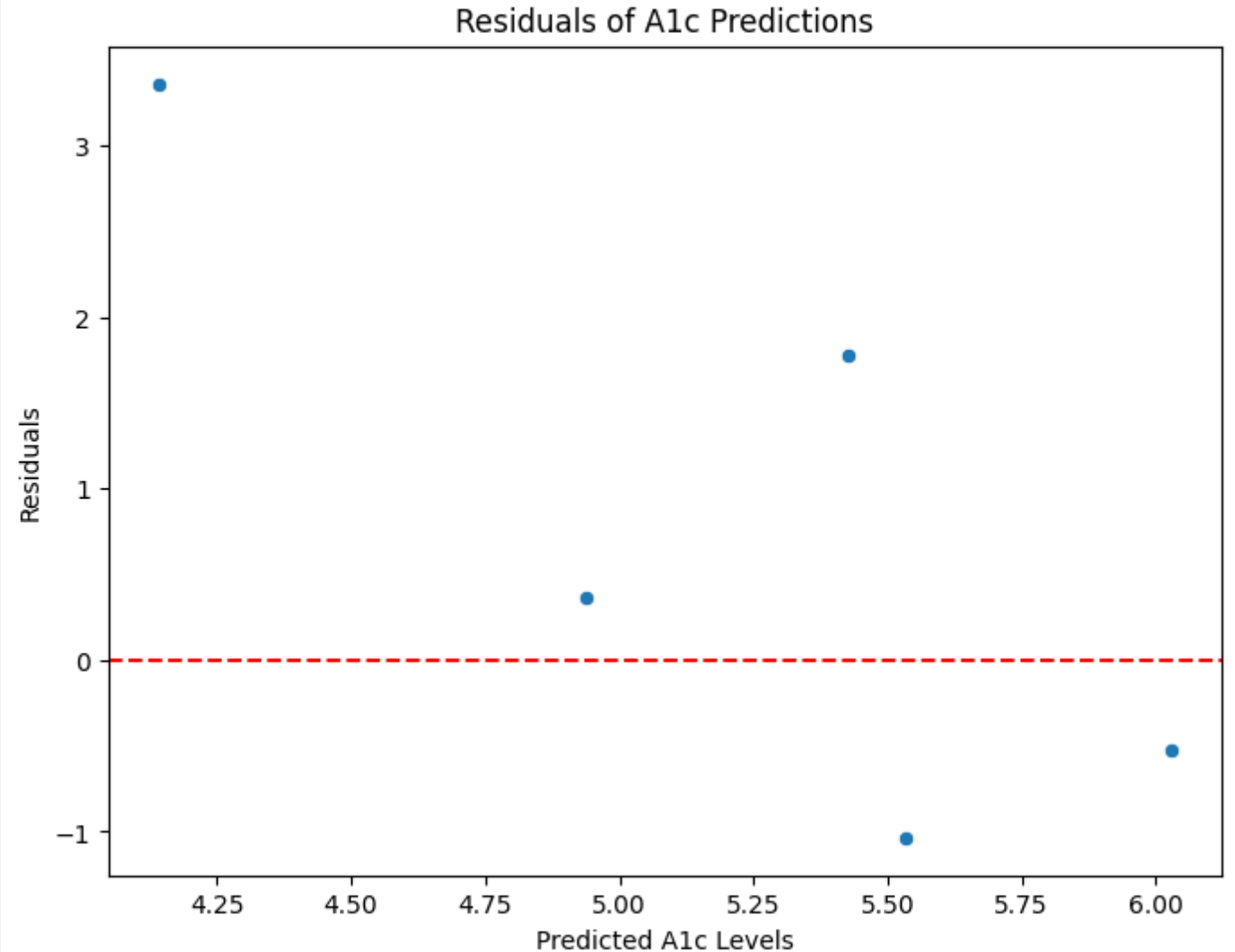
```
r2 = r2_score(test_labels_a1c, labels_pred)
```

```
print(f"Mean Squared Error: {mse}")
```

```
print(f"R^2 Score: {r2}")
```

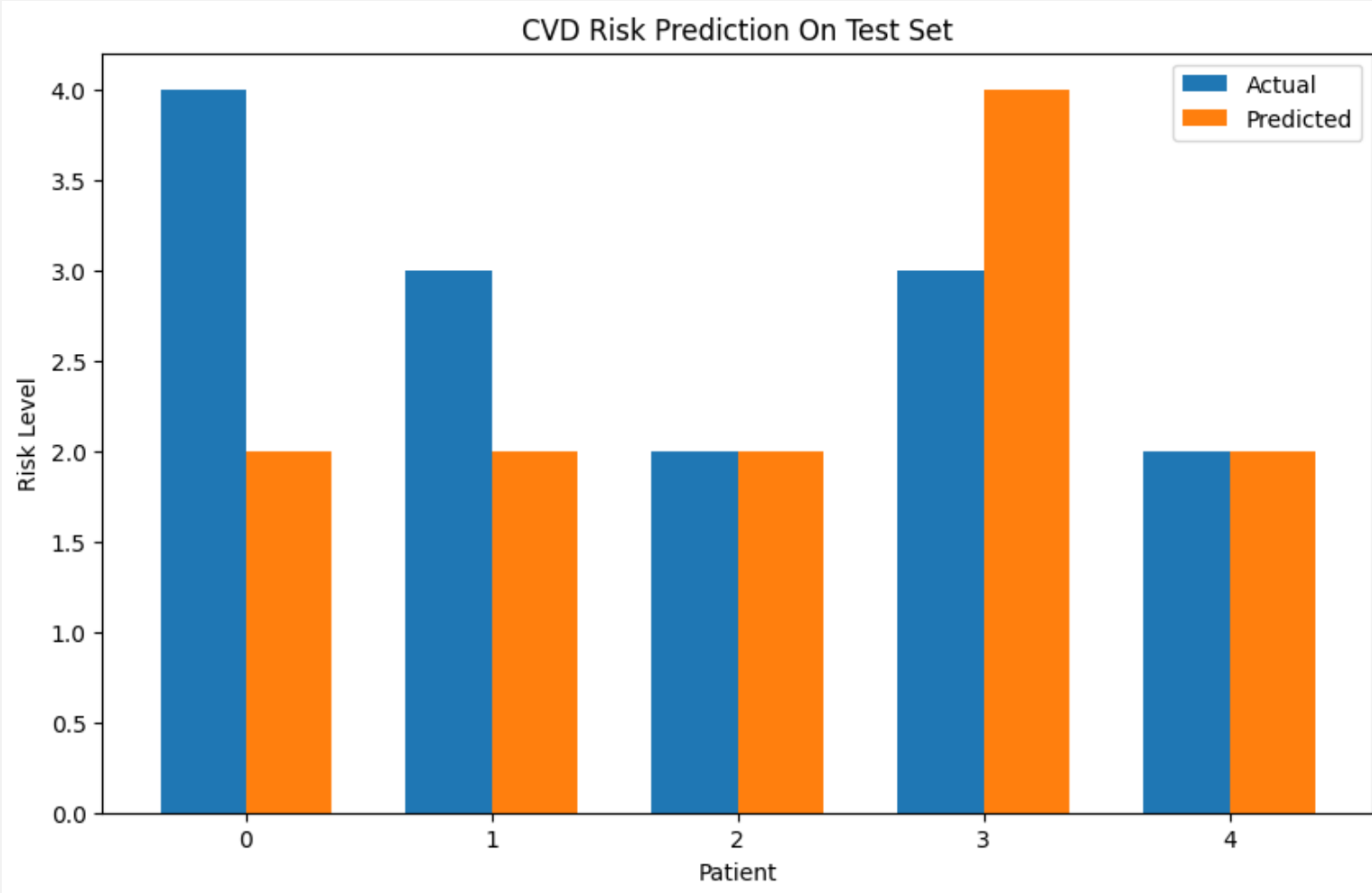
Mean Squared Error: 3.179935804793055

R^2 Score: -1.3801914706534837



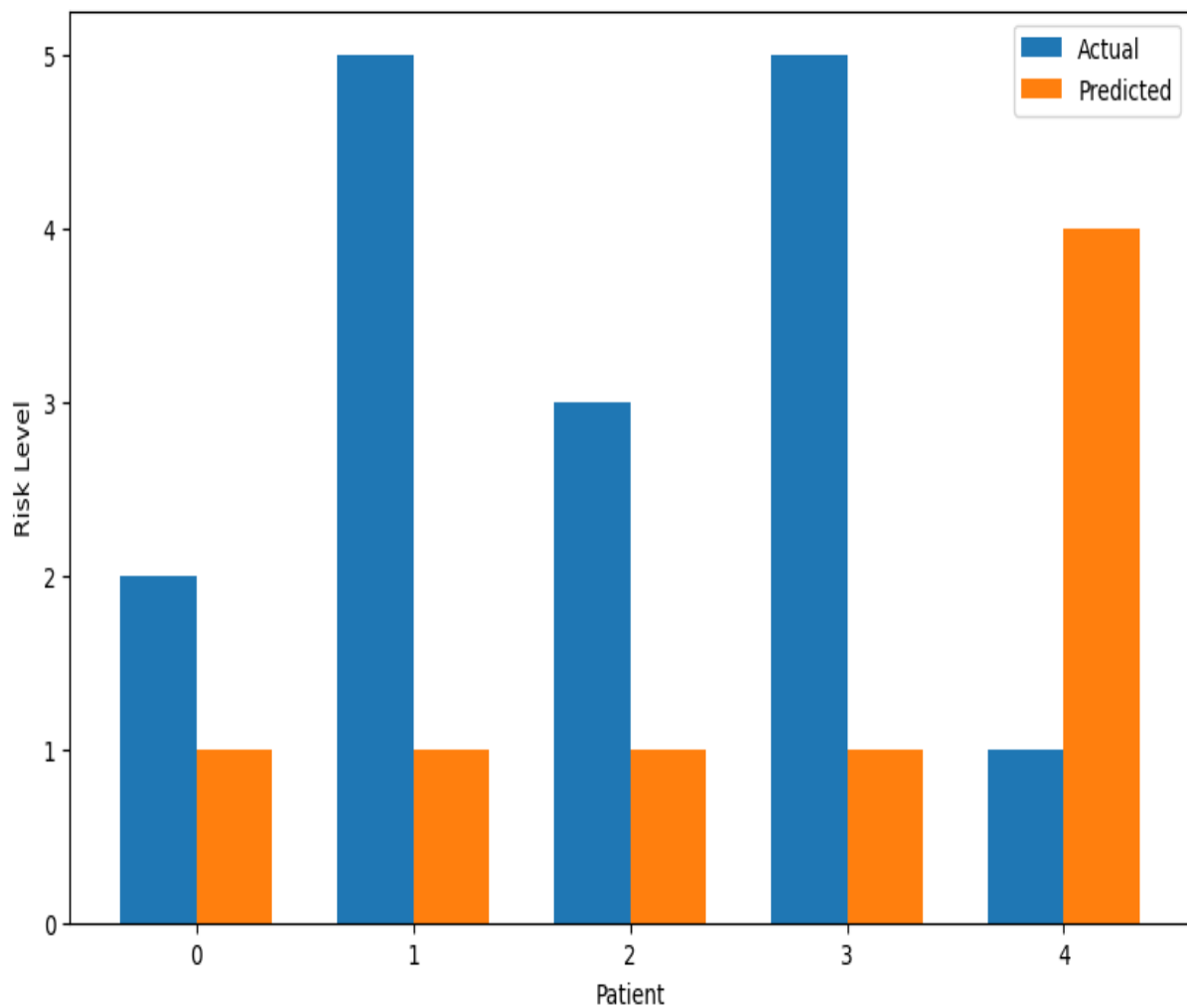
Classification Model Results (Mediocre Results)

Training Accuracy: CVD: 1.0, Endocrine: 1.0, Respiratory: 1.0
Validation Accuracy: CVD: 0.6, Endocrine: 0.2, Respiratory: 0.2
Test Accuracy: CVD: 0.4, Endocrine: 0.0, Respiratory: 0.4



Classification Model Results (Mediocre Results)

Endocrine Risk Prediction On Test Set



Respiratory Risk Prediction On Test Set

