

**Project Report**

Statistics for Data Science

Semester – 2

**"Heart Attack Risk Prediction Using Machine Learning"**

By

Prem Panchal

Reg no: 2411021240011

GitHub link: [Prem2309/DATA-SCIENCE](https://github.com/Prem2309/DATA-SCIENCE)

Department of Computer Application

Alliance University Chandapura — Anekal Main Road,

Anekal Bengaluru — 562 106

April 2025

# Project Overview

This project aims to predict the likelihood of a heart attack in individuals based on various medical attributes and lifestyle factors. Using a dataset containing information about age, gender, cholesterol levels, blood pressure, and other relevant health indicators, we apply machine learning algorithms to build predictive models. Through exploratory data analysis, data cleaning, and model evaluation, this project demonstrates how data-driven approaches can aid in early diagnosis and risk assessment for heart disease.

# Project Goal

* To develop a machine learning model capable of accurately predicting the risk of a heart attack.
* To perform data preprocessing, visualization, and correlation analysis to understand key health factors contributing to heart disease.
* To compare different classification models like Logistic Regression and Random Forest to determine the best-performing algorithm.
* To raise awareness on how lifestyle and health indicators impact heart attack risks using real patient data.

# Challenges Faced

1. **Missing or Inconsistent Data**: Ensuring the dataset was clean and complete without null values or anomalies.
2. **Imbalanced Data**: Ensuring that the model did not become biased due to imbalance in the target variable (heart attack risk).
3. **Overfitting with Complex Models**: Avoiding overfitting when trying more powerful models like Random Forest.
4. **Model Accuracy**: Achieving satisfactory prediction accuracy while avoiding underfitting with simple models like Logistic Regression.
5. **Understanding Feature Impact**: Interpreting how different features such as cholesterol, age, or lifestyle habits affect heart attack risk.

import pandas as pd  
df=pd.read\_csv(r"C:\Users\Prem Panchal\Downloads\heart\_attack\_prediction\_india (2).csv")

df

Age Gender Chest Pain Type Resting Blood Pressure Cholesterol \  
0 69 0 4 109 487   
1 32 0 2 125 170   
2 89 0 2 186 543   
3 78 0 2 104 367   
4 38 1 1 151 158   
... ... ... ... ... ...   
6995 69 1 4 144 110   
6996 23 0 1 177 481   
6997 86 1 2 174 315   
6998 91 0 4 198 129   
6999 26 0 3 132 193   
  
 Fasting Blood Sugar (> 120 mg/dL) Resting ECG Results \  
0 1 1   
1 0 1   
2 1 0   
3 0 0   
4 1 0   
... ... ...   
6995 0 0   
6996 0 0   
6997 1 0   
6998 1 1   
6999 0 0   
  
 Max Heart Rate Achieved Exercise-Induced Angina \  
0 122 1   
1 113 1   
2 165 1   
3 117 1   
4 123 1   
... ... ...   
6995 105 1   
6996 135 1   
6997 203 0   
6998 194 0   
6999 210 1   
  
 Old Peak (ST Depression) ... Thalassemia Smoking History \  
0 3.51 ... 2 1   
1 4.74 ... 1 0   
2 2.70 ... 1 0   
3 0.62 ... 1 1   
4 2.00 ... 2 0   
... ... ... ... ...   
6995 2.08 ... 3 0   
6996 0.17 ... 3 0   
6997 2.77 ... 3 0   
6998 3.42 ... 3 1   
6999 1.49 ... 2 1   
  
 Obesity (BMI > 30) Hypertension History Diabetes History \  
0 1 0 0   
1 1 1 0   
2 0 1 0   
3 0 1 0   
4 1 1 0   
... ... ... ...   
6995 0 1 0   
6996 0 1 1   
6997 0 0 1   
6998 0 1 0   
6999 0 1 1   
  
 Alcohol Consumption Physical Activity Stress Levels \  
0 0 0 1   
1 0 0 2   
2 1 1 3   
3 1 0 1   
4 1 0 2   
... ... ... ...   
6995 1 0 1   
6996 1 0 1   
6997 0 0 3   
6998 1 0 2   
6999 0 0 1   
  
 Family History of Heart Disease Heart Attack Risk   
0 1 0   
1 0 1   
2 0 0   
3 0 1   
4 0 1   
... ... ...   
6995 0 0   
6996 0 1   
6997 0 1   
6998 1 0   
6999 1 1   
  
[7000 rows x 22 columns]

# Display the first few rows of the dataset as a sanity check  
df.head()

Age Gender Chest Pain Type Resting Blood Pressure Cholesterol \  
0 69 0 4 109 487   
1 32 0 2 125 170   
2 89 0 2 186 543   
3 78 0 2 104 367   
4 38 1 1 151 158   
  
 Fasting Blood Sugar (> 120 mg/dL) Resting ECG Results \  
0 1 1   
1 0 1   
2 1 0   
3 0 0   
4 1 0   
  
 Max Heart Rate Achieved Exercise-Induced Angina Old Peak (ST Depression) \  
0 122 1 3.51   
1 113 1 4.74   
2 165 1 2.70   
3 117 1 0.62   
4 123 1 2.00   
  
 ... Thalassemia Smoking History Obesity (BMI > 30) \  
0 ... 2 1 1   
1 ... 1 0 1   
2 ... 1 0 0   
3 ... 1 1 0   
4 ... 2 0 1   
  
 Hypertension History Diabetes History Alcohol Consumption \  
0 0 0 0   
1 1 0 0   
2 1 0 1   
3 1 0 1   
4 1 0 1   
  
 Physical Activity Stress Levels Family History of Heart Disease \  
0 0 1 1   
1 0 2 0   
2 1 3 0   
3 0 1 0   
4 0 2 0   
  
 Heart Attack Risk   
0 0   
1 1   
2 0   
3 1   
4 1   
  
[5 rows x 22 columns]

df.tail()

Age Gender Chest Pain Type Resting Blood Pressure Cholesterol \  
6995 69 1 4 144 110   
6996 23 0 1 177 481   
6997 86 1 2 174 315   
6998 91 0 4 198 129   
6999 26 0 3 132 193   
  
 Fasting Blood Sugar (> 120 mg/dL) Resting ECG Results \  
6995 0 0   
6996 0 0   
6997 1 0   
6998 1 1   
6999 0 0   
  
 Max Heart Rate Achieved Exercise-Induced Angina \  
6995 105 1   
6996 135 1   
6997 203 0   
6998 194 0   
6999 210 1   
  
 Old Peak (ST Depression) ... Thalassemia Smoking History \  
6995 2.08 ... 3 0   
6996 0.17 ... 3 0   
6997 2.77 ... 3 0   
6998 3.42 ... 3 1   
6999 1.49 ... 2 1   
  
 Obesity (BMI > 30) Hypertension History Diabetes History \  
6995 0 1 0   
6996 0 1 1   
6997 0 0 1   
6998 0 1 0   
6999 0 1 1   
  
 Alcohol Consumption Physical Activity Stress Levels \  
6995 1 0 1   
6996 1 0 1   
6997 0 0 3   
6998 1 0 2   
6999 0 0 1   
  
 Family History of Heart Disease Heart Attack Risk   
6995 0 0   
6996 0 1   
6997 0 1   
6998 1 0   
6999 1 1   
  
[5 rows x 22 columns]

# Data Cleaning and Preprocessing

# Check for missing values and data types  
print('Missing values:')  
print(df.isnull().sum())  
  
print('\nData types:')  
print(df.dtypes)

Missing values:  
Age 0  
Gender 0  
Chest Pain Type 0  
Resting Blood Pressure 0  
Cholesterol 0  
Fasting Blood Sugar (> 120 mg/dL) 0  
Resting ECG Results 0  
Max Heart Rate Achieved 0  
Exercise-Induced Angina 0  
Old Peak (ST Depression) 0  
Slope of Peak Exercise ST Segment 0  
Number of Major Vessels (0-3) 0  
Thalassemia 0  
Smoking History 0  
Obesity (BMI > 30) 0  
Hypertension History 0  
Diabetes History 0  
Alcohol Consumption 0  
Physical Activity 0  
Stress Levels 0  
Family History of Heart Disease 0  
Heart Attack Risk 0  
dtype: int64  
  
Data types:  
Age int64  
Gender int64  
Chest Pain Type int64  
Resting Blood Pressure int64  
Cholesterol int64  
Fasting Blood Sugar (> 120 mg/dL) int64  
Resting ECG Results int64  
Max Heart Rate Achieved int64  
Exercise-Induced Angina int64  
Old Peak (ST Depression) float64  
Slope of Peak Exercise ST Segment int64  
Number of Major Vessels (0-3) int64  
Thalassemia int64  
Smoking History int64  
Obesity (BMI > 30) int64  
Hypertension History int64  
Diabetes History int64  
Alcohol Consumption int64  
Physical Activity int64  
Stress Levels int64  
Family History of Heart Disease int64  
Heart Attack Risk int64  
dtype: object

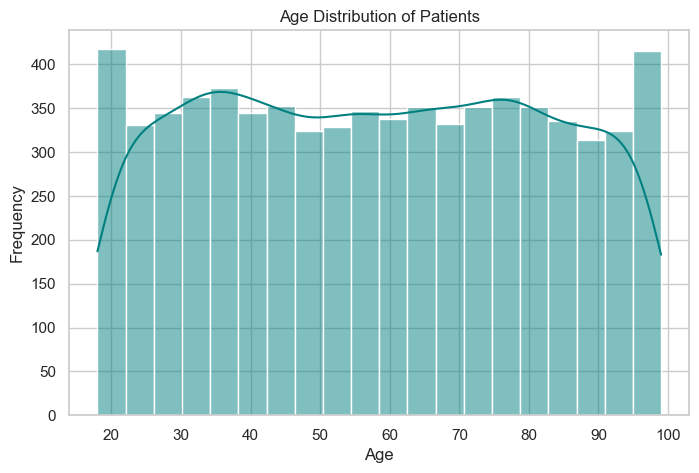
# A brief descriptive statistics overview  
df.describe()

Age Gender Chest Pain Type Resting Blood Pressure \  
count 7000.000000 7000.000000 7000.000000 7000.000000   
mean 58.227857 0.492571 2.507857 144.207857   
std 23.511403 0.499981 1.119491 31.682105   
min 18.000000 0.000000 1.000000 90.000000   
25% 38.000000 0.000000 2.000000 117.000000   
50% 58.000000 0.000000 3.000000 144.000000   
75% 78.000000 1.000000 4.000000 172.000000   
max 99.000000 1.000000 4.000000 199.000000   
  
 Cholesterol Fasting Blood Sugar (> 120 mg/dL) Resting ECG Results \  
count 7000.000000 7000.000000 7000.000000   
mean 351.351000 0.492429 0.489571   
std 146.303141 0.499978 0.499927   
min 100.000000 0.000000 0.000000   
25% 223.000000 0.000000 0.000000   
50% 354.000000 0.000000 0.000000   
75% 480.000000 1.000000 1.000000   
max 599.000000 1.000000 1.000000   
  
 Max Heart Rate Achieved Exercise-Induced Angina \  
count 7000.000000 7000.000000   
mean 144.431857 0.504571   
std 43.224027 0.500015   
min 70.000000 0.000000   
25% 107.000000 0.000000   
50% 144.000000 1.000000   
75% 182.000000 1.000000   
max 219.000000 1.000000   
  
 Old Peak (ST Depression) ... Thalassemia Smoking History \  
count 7000.000000 ... 7000.000000 7000.000000   
mean 2.997859 ... 1.986857 0.497714   
std 1.740264 ... 0.813937 0.500030   
min 0.000000 ... 1.000000 0.000000   
25% 1.500000 ... 1.000000 0.000000   
50% 2.990000 ... 2.000000 0.000000   
75% 4.530000 ... 3.000000 1.000000   
max 6.000000 ... 3.000000 1.000000   
  
 Obesity (BMI > 30) Hypertension History Diabetes History \  
count 7000.000 7000.000000 7000.000000   
mean 0.506 0.498000 0.493571   
std 0.500 0.500032 0.499994   
min 0.000 0.000000 0.000000   
25% 0.000 0.000000 0.000000   
50% 1.000 0.000000 0.000000   
75% 1.000 1.000000 1.000000   
max 1.000 1.000000 1.000000   
  
 Alcohol Consumption Physical Activity Stress Levels \  
count 7000.000000 7000.000000 7000.000000   
mean 0.489000 0.499571 2.010286   
std 0.499915 0.500036 0.812924   
min 0.000000 0.000000 1.000000   
25% 0.000000 0.000000 1.000000   
50% 0.000000 0.000000 2.000000   
75% 1.000000 1.000000 3.000000   
max 1.000000 1.000000 3.000000   
  
 Family History of Heart Disease Heart Attack Risk   
count 7000.000000 7000.000000   
mean 0.498286 0.500429   
std 0.500033 0.500036   
min 0.000000 0.000000   
25% 0.000000 0.000000   
50% 0.000000 1.000000   
75% 1.000000 1.000000   
max 1.000000 1.000000   
  
[8 rows x 22 columns]

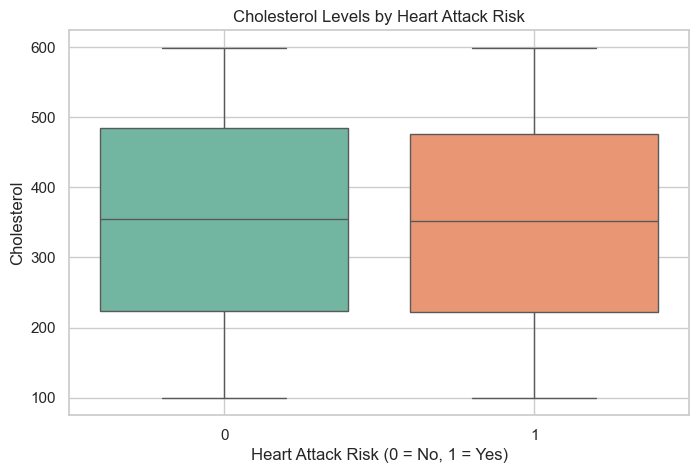
Data Visualization

import pandas as pd  
df = pd.read\_csv(r"C:\Users\Prem Panchal\Downloads\heart\_attack\_prediction\_india (2).csv")

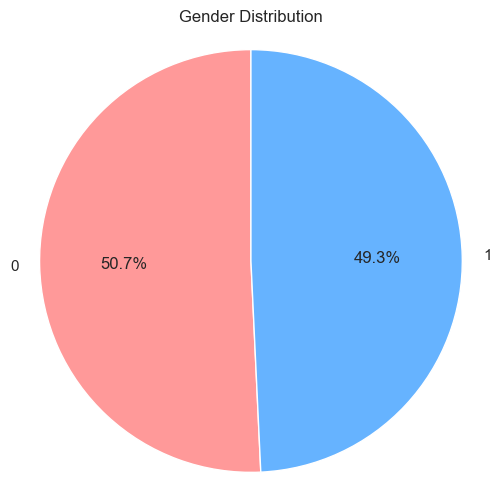
import matplotlib.pyplot as plt  
import seaborn as sns  
  
# Histogram of Age  
plt.figure(figsize=(8, 5))  
sns.histplot(df['Age'], bins=20, kde=True, color='teal')  
plt.title('Age Distribution of Patients')  
plt.xlabel('Age')  
plt.ylabel('Frequency')  
plt.grid(True)  
plt.show()



# BOX PLOT — Cholesterol by Heart Attack Risk  
  
plt.figure(figsize=(8, 5))  
sns.boxplot(x='Heart Attack Risk', y='Cholesterol', data=df, palette='Set2')  
plt.title('Cholesterol Levels by Heart Attack Risk')  
plt.xlabel('Heart Attack Risk (0 = No, 1 = Yes)')  
plt.ylabel('Cholesterol')  
plt.grid(True)  
plt.show()

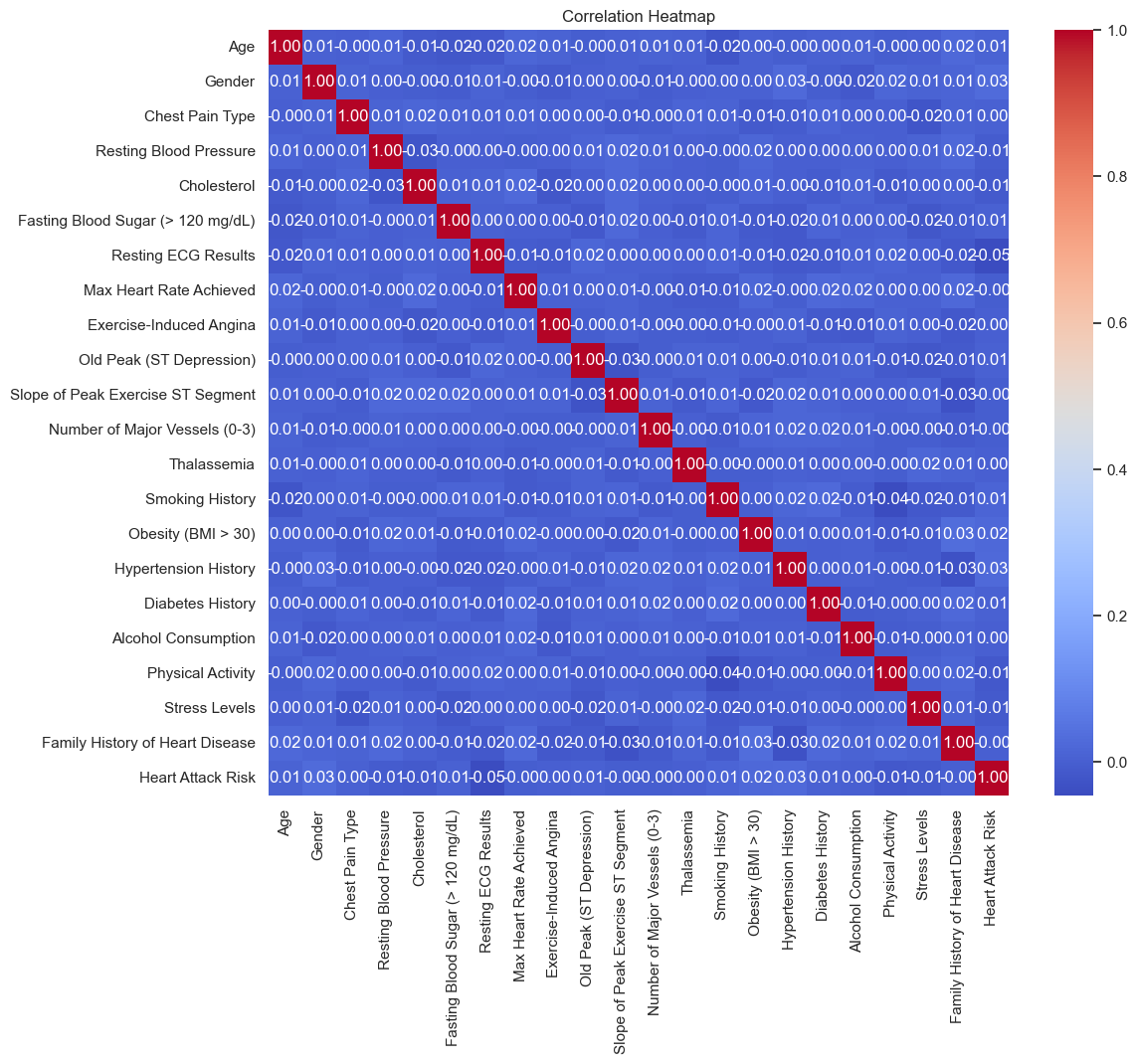


# PIE CHART — Gender Distribution  
plt.figure(figsize=(6, 6))  
gender\_counts = df['Gender'].value\_counts()  
plt.pie(gender\_counts, labels=gender\_counts.index, autopct='%1.1f%%', colors=['#ff9999', '#66b3ff'], startangle=90)  
plt.title('Gender Distribution')  
plt.axis('equal')  
plt.show()



# Correlation Analysis

# Select only numeric columns for correlation analysis  
numeric\_df = df.select\_dtypes(include=[np.number])  
  
# Check whether there are four or more numeric columns  
if numeric\_df.shape[1] >= 4:  
 plt.figure(figsize=(12, 10))  
 corr = numeric\_df.corr()  
 sns.heatmap(corr, annot=True, cmap='coolwarm', fmt='.2f')  
 plt.title('Correlation Heatmap')  
 plt.show()  
else:  
 print('Not enough numeric columns for a correlation heatmap.')



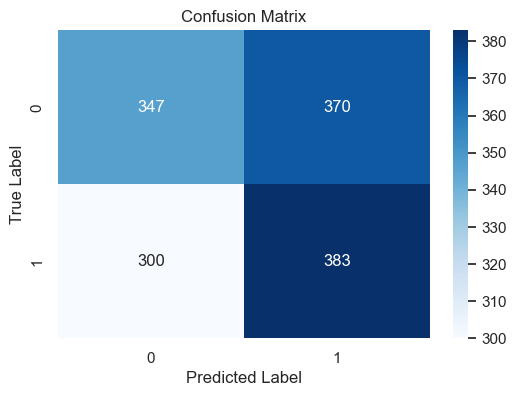
# Predictor (Model Building)

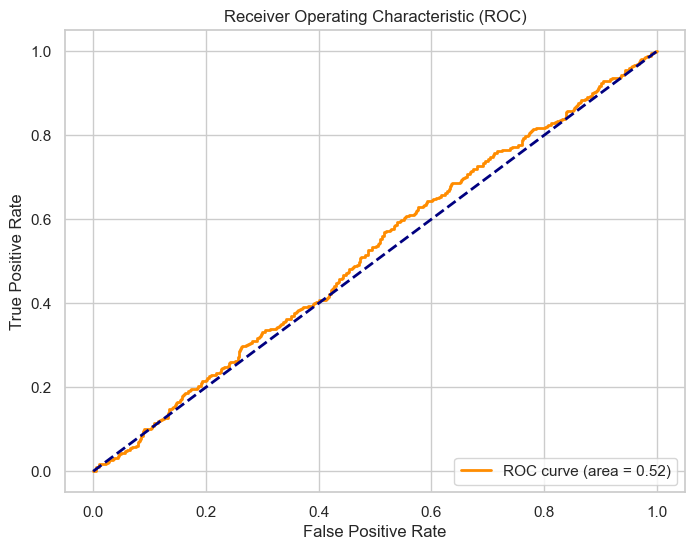
# Define features (X) and target (y)  
# For simplicity, we use all other variables as predictors, removing the target column  
X = df.drop(columns=['Heart Attack Risk'])  
y = df['Heart Attack Risk']  
  
# Split the data into training and testing sets (80-20 split)  
X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.20, random\_state=42)  
  
# Initialize and train logistic regression model  
model = LogisticRegression(max\_iter=1000)  
model.fit(X\_train, y\_train)  
  
# Make predictions and calculate prediction accuracy  
y\_pred = model.predict(X\_test)  
accuracy = accuracy\_score(y\_test, y\_pred)  
print(f'Prediction Accuracy Score: {accuracy:.4f}')

Prediction Accuracy Score: 0.5214

# Model Evaluation

# Confusion Matrix  
cm = confusion\_matrix(y\_test, y\_pred)  
plt.figure(figsize=(6, 4))  
sns.heatmap(cm, annot=True, fmt='d', cmap='Blues')  
plt.title('Confusion Matrix')  
plt.xlabel('Predicted Label')  
plt.ylabel('True Label')  
plt.show()  
  
# ROC Curve  
y\_pred\_prob = model.predict\_proba(X\_test)[:, 1]  
fpr, tpr, thresholds = roc\_curve(y\_test, y\_pred\_prob)  
roc\_auc = auc(fpr, tpr)  
  
plt.figure(figsize=(8, 6))  
plt.plot(fpr, tpr, color='darkorange', lw=2, label=f'ROC curve (area = {roc\_auc:.2f})')  
plt.plot([0, 1], [0, 1], color='navy', lw=2, linestyle='--')  
plt.xlabel('False Positive Rate')  
plt.ylabel('True Positive Rate')  
plt.title('Receiver Operating Characteristic (ROC)')  
plt.legend(loc='lower right')  
plt.show()





# Summary and Future Work

Our exploratory analysis of the heart attack risk dataset has given us several insights. I performed a range of visualizations and built a logistic regression model that achieved a reasonable accuracy. Although logistic regression is a good starting point, future work might include:

Testing more sophisticated models (e.g., Random Forest, Gradient Boosting, or Neural Networks) for improved accuracy. Feature engineering to incorporate interaction terms or non-linear transformations. Cross-validation to ensure that our model's performance is robust. We also observed a few outliers in the distributions, which may warrant further investigation or even the application of robust models. The methods used to resolve potential errors in data types and missing values are quite standard for anyone dealing with real-world data