



# What Causes Heart Disease?

By Rachel Muralitharan and Ananya Kaalva



# Explain Dataset

- Originating from the University of California - Irvine Machine Learning Repository, this dataset from 1988 contains data from a Cleveland based heart disease study [1].
- Includes data from patients with and without heart disease, as well as other aspects of their cardiac and circulatory health
- Includes 14 main attributes including the target heart disease status

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target
0	63	1	3	145	233	1	0	150	0	2.3	0	0	1	1
1	37	1	2	130	250	0	1	187	0	3.5	0	0	2	1
2	41	0	1	130	204	0	0	172	0	1.4	2	0	2	1
3	56	1	1	120	236	0	1	178	0	0.8	2	0	2	1
4	57	0	0	120	354	0	1	163	1	0.6	2	0	2	1
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
298	57	0	0	140	241	0	1	123	1	0.2	1	0	3	0
299	45	1	3	110	264	0	1	132	0	1.2	1	0	3	0
300	68	1	0	144	193	1	1	141	0	3.4	1	2	3	0
301	57	1	0	130	131	0	1	115	1	1.2	1	1	3	0
302	57	0	1	130	236	0	0	174	0	0.0	1	1	2	0

303 rows × 14 columns

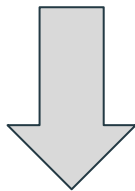
# Dataset Features

- **Age:** patient's age in years
- **Sex:** 1 = male, 0 = female
- **Chest Pain:** patient's chest pain type ( 0 = asymptomatic, 1 = atypical angina, 2 = non-anginal pain, 3 = typical angina)
- **Resting Blood Pressure:** reported in mm Hg when the patient is admitted
- **Cholesterol:** patient's cholesterol measurement in mg/dl
- **Fasting Blood Sugar:** patient's blood sugar level (1 = greater than 120 mg/dl, 0 = less than 120 mg/dl)
- **Rest ECG:** patient's electrocardiographic exam (0 = left ventricular hypertrophy, 1 = normal results, 2 = ST-T wave abnormality)
- **Max Heart Rate:** maximum heart rate achieved by patient
- **Exercise Angina:** patient's response to exercise (1 = angina, 0 = no angina)
- **Exercise ECG:** ST depression observed in a patient's ECG after exercise
- **ECG Slope:** slope of a patient's ECG ST segment after exercise (0 = downloping, 1 = flat, 2 = upsloping)
- **Number Blood Vessels:** number of major vessels, integer from 0-3
- **Thalassemia:** thalassemia disease status (0 = NULL, 1 = fixed defect, 2 = normal blood flow, 3 = reversible defect)
- **Target:** categorical variable representing whether the patient has heart disease (0 = yes, 1 = no)

# What Heart Health metrics accurately predict heart disease?

**Question 1:** Are cholesterol levels higher than average for patients with heart disease?

**Metric: One-sided T test**



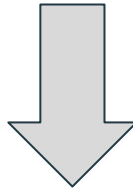
**Question 2:** What are the best categorical and numeric variables to predict heart disease?

**Metric: Random Forest**

# What Heart Health metrics accurately predict heart disease?

**Question 3:** Can heart disease be predicted using a patient's number of major vessels and their chest pain level?

**Metric: Categorical Tree**





**Question 4:** Can heart disease be predicted using a patient's ECG oldpeak and maximum heart rate?

**Metric: Logistic Regression**



# Data Cleaning Implementation

- Cholesterol over 550 was cited to be extremely improbable by NIH [2].
- According to our dataset description, Thalassemia status values equal to 0 are null and should be dropped [3].

```
▶ # cleaning data
df.columns = ['age', 'sex', 'chest pain', 'resting blood pressure', 'cholesterol', 'fasting blood sugar', 'rest ecg', 'max heart rate achieved', 'exercise induced angina', 'slope', 'resting electrocardiographic results', 'major vessels (by angiography)', 'oldpeak', 'exercise induced angina', 'slope', 'resting electrocardiographic results', 'major vessels (by angiography)', 'oldpeak']
df = df.drop(df[df['cholesterol'] > 550].index) # chol level too high to be probable
df = df.drop(df[df['thalassemia'] == 0].index) # null
```



Subquestion 1: Are  
cholesterol levels higher than  
US national average for  
patients with heart disease?



# T-test Implementation

**Why t-test?** Compares the mean cholesterol level of patients with heart disease sample to the mean average cholesterol level from 1988 determined by the NIH [4].

```
# t test condition check
chol = df[df['target'] == 0] # get chol values from dataframe where target = 0, only instances with heart disease
chol = chol['cholesterol']

# Question: Are individuals from the 1988 Cleavelend study who were identified as having heart disease have higher cholesterol
# than the national US average cholesterol in 1988 of 206 mg/dL.

from scipy.stats import t
from scipy import stats

critical_value = t.ppf(q=1-.05, df= (len(chol) - 1), loc = 0, scale = 1)
test = stats.ttest_1samp(chol, 206, alternative = 'greater') # one sided t test seeing if mean is greater than 206
t = test[0]          # gets t statistic from array
p = test[1]          # gets p-value from array

print('critical value: ', critical_value)
print('t-value: ', t)
print('p-value: ', p)

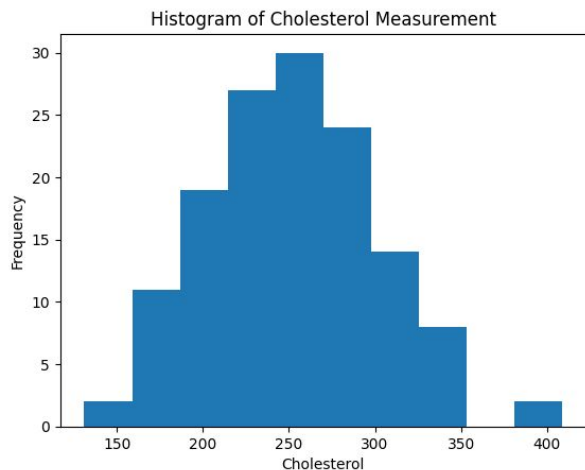
# reject null hypothesis. Therefore, it can be concluded that the mean cholesterol level of heart disease individuals in the
# Cleavelend study is greater than the national cholesterol average of 206.
```



# T-test Results



## Conditions Passed:

- The sample is random
- The participants are all independent of one another
- The cholesterol distribution for patients with heart disease
- Test Chosen: one sample, one sided T test





## Overall Results:

- $H_0: \mu \text{ heart disease} = 206 \text{ mg/dl}$
- $H_a: \mu \text{ heart disease} > 206 \text{ mg/dl}$
- $\alpha = .05$
- $t = \bar{x} - \mu / (s/\sqrt{n})$
- Decision Rule: Reject  $H_0$  if  $|t| \geq 1.65$
- **t-value:** 10.748896329604412  
> critical value: 1.6543139565251865
- **p-value:** 3.452718685769144e-20 < .05
- **Conclusion:** As the t-value is greater than the critical value, we reject the null hypothesis. This means we have significant evidence at the  $\alpha = .05$  level that the mean cholesterol level for those who have heart disease is greater than 206 mg/dl.



Subquestion 2: What are the best categorical and numeric variables to predict heart disease?



# Random Forest Parameter testing

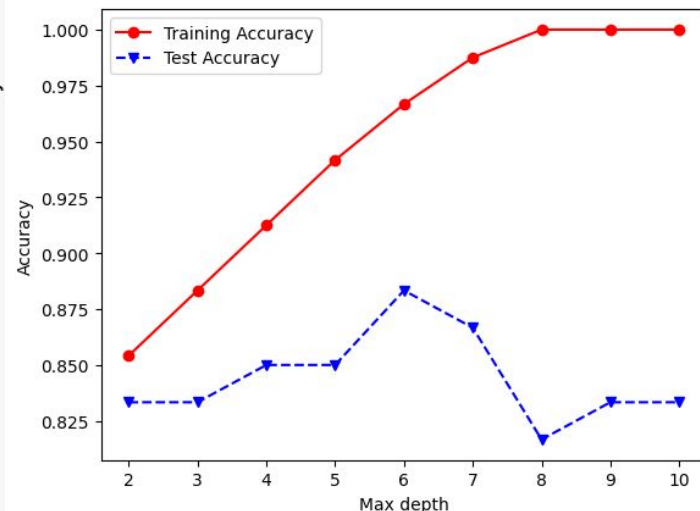
**Why Random Forest?** Uses a bootstrap method of generating multiple trees with randomly selected variables to prevent overfitting and rank input variables based on importance.

```
y = df['target']
X = df.drop('target', axis=1)
X = pd.get_dummies(X, drop_first=True)

# Create a training/testing split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, r

maxdepths = [2,3,4,5,6,7,8,9,10]
trainAcc = np.zeros(len(maxdepths))
testAcc = np.zeros(len(maxdepths))

index = 0
for depth in maxdepths:
    rfModel = RandomForestClassifier(max_depth=depth, random_state = 0)
    rfModel = rfModel.fit(X_train, y_train)
    Y_predTrain = rfModel.predict(X_train)
    Y_predTest = rfModel.predict(X_test)
    trainAcc[index] = accuracy_score(y_train, Y_predTrain)
    testAcc[index] = accuracy_score(y_test, Y_predTest)
    index += 1
```



# Random Forest Implementation



```
# Initialize the random forest model
rfModel = RandomForestClassifier(max_depth=6, max_features='sqrt', random_state=0)
# Fit the random forest model on the training data
rfModel.fit(X_train, y_train)

# get importance scores
print(pd.DataFrame(
    data={
        'feature': rfModel.feature_names_in_,
        'importance': rfModel.feature_importances_,
    }
).sort_values('importance', ascending=False))
y_pred = rfModel.predict(X_test)
print('accuracy: ', accuracy_score(y_pred, y_test))
```



# Random Forest Results

---

	feature	importance
2	chest pain	0.149583
11	number blood vessels	0.131154
9	exercise ecg	0.117377
12	thalassemia	0.112149
7	max heart rate	0.097761
0	age	0.087566
4	cholesterol	0.071981
3	resting blood pressure	0.065772
8	exercise angina	0.064305
10	ecg slope	0.044082
1	sex	0.032856
6	rest ecg	0.018184
5	fasting blood sugar	0.007229
accuracy:		0.8833333333333333



Subquestion 3: Can heart disease be predicted using a patient's number of major vessels and their chest pain status?



# Categorical Tree Implementation

**Why Categorical Tree?** Develops a clear, flowchart style representation of predicting heart disease by recursively splitting input data into subsets to identify significant features.

```
| from sklearn.tree import DecisionTreeClassifier, export_text
  from sklearn import metrics, tree

y = df[['target']]
X = df[['chest pain', 'number blood vessels']]

# Create a training/testing split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=0)

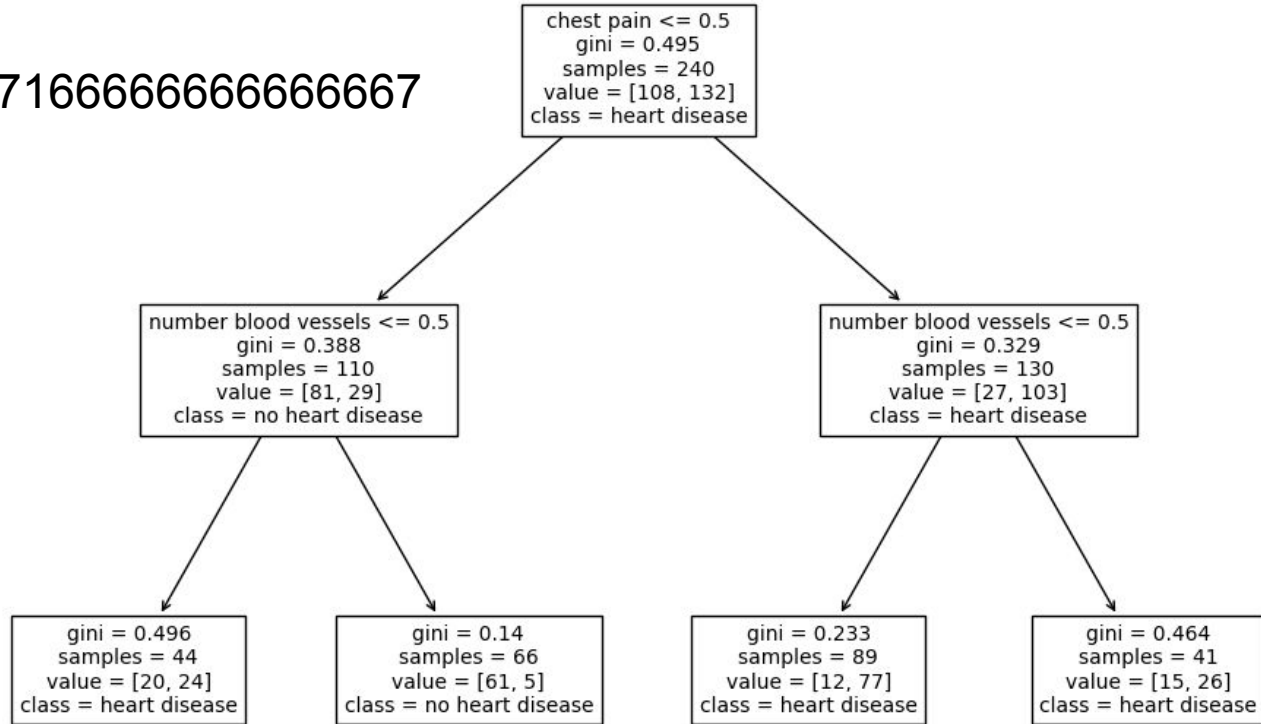
classtreeModel = DecisionTreeClassifier(max_depth=2)
classtreeModel = classtreeModel.fit(X_train, y_train)
y_pred = classtreeModel.predict(X_test)

# Resize the plotting window
plt.figure(figsize=[12, 8])

class_names = ['no heart disease', 'heart disease']
p = tree.plot_tree(classtreeModel, feature_names=X.columns, filled=False, fontsize=10, class_names=class_names)
print('accuracy: ', accuracy_score(y_pred, y_test))
```

# Categorical Tree - Results

accuracy: 0.7166666666666667





Subquestion 4: Can heart disease be predicted using a patient's ECG oldpeak and maximum heart rate?

# Logistic Regression Implementation

**Why Logistic Regression?** Good for predicting the likelihood of a binary target variable (heart disease) by building a sigmoid curve representing the odds of an outcome using one or more features.

```
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score
from sklearn.metrics import classification_report

X = df_h[['max heart rate', 'exercise ecg']]

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=123)

model = LogisticRegression( fit_intercept=True)
model.fit(X_train,y_train)
y_pred = model.predict(X_test)

print(classification_report(y_test,y_pred))
```

# Logistic Regression - Results

	precision	recall	f1-score	support
0	0.75	0.78	0.76	27
1	0.81	0.79	0.80	33
accuracy			0.78	60
macro avg	0.78	0.78	0.78	60
weighted avg	0.78	0.78	0.78	60

# Overall Conclusions and Broader Context

**What are the best variables to predict heart disease?** The best categorical variables are chest pain and number of impacted vessels. The best numeric variables are maximum heart rate, and exercise ECG.

**Thresholds for categorical variables:** Individuals with a pain level above .5 are likely to have heart disease. Individuals with the number of impacted vessels above .5 are likely to have heart disease.

# References

- 1) “Health Care: Heart Attack Possibility.” *Www.kaggle.com*, [www.kaggle.com/datasets/nareshbhat/health-care-data-set-on-heart-attack-possibility?resource=download](https://www.kaggle.com/datasets/nareshbhat/health-care-data-set-on-heart-attack-possibility?resource=download). Accessed 7 July 2023.
- 2) Pejic, R. N. Familial Hypercholesterolemia. *The Ochsner Journal* 2014, 14 (4), 669–672.
- 3) Deshmukh, H. *Heart Disease UCI Diagnosis & Prediction*. Medium. <https://towardsdatascience.com/heart-disease-uci-diagnosis-prediction-b1943ee835a7>.
- 4) Carroll, M. D.; Lacher, D. A.; Sorlie, P. D.; Cleeman, J. I.; Gordon, D. J.; Wolz, M.; Grundy, S. M.; Johnson, C. L. Trends in Serum Lipids and Lipoproteins of Adults, 1960-2002. *JAMA* 2005, 294 (14), 1773–1781. <https://doi.org/10.1001/jama.294.14.1773>.