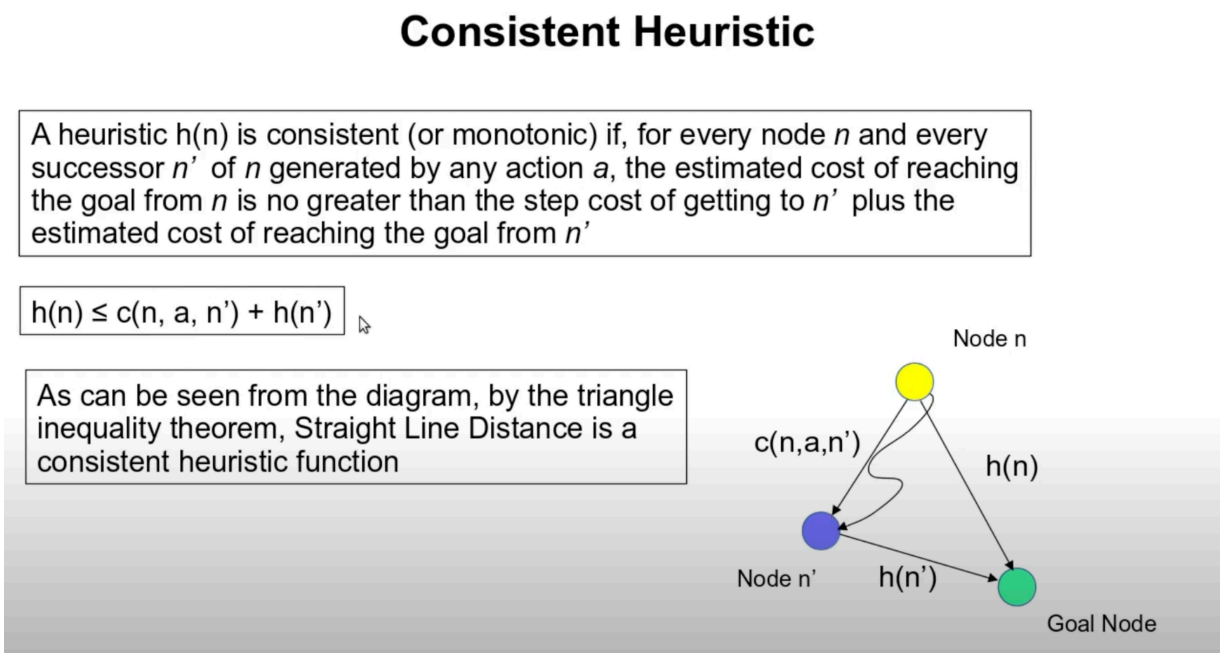
🧠

UCI_EDA AP

Course #	5100 - Intro AI
Task	Class Lecture
Parent Class	🧠 *Intro to AI Syllabus, 🧠 🔥🔥🔥 AI FINAL PROJECT 🔥🔥🔥
URL	https://rajagopalvenkat.com/teaching/resources/AI/
Progress	In progress
Algorithms, Formulas & Equations DB	*Artificial Intelligence, Artificial Intelligence Tutorial | AI Tutorial - GeeksforGeeks
Week	11
Office Hours	Khoury Office Hours App for all office hours (online and in-person) for efficient queue management this semester. Instructor - Raj • In-person, Meserve 303, Tuesday, 9:30 am - 11:30 am. • To schedule appointments outside of office hours, visit my appointments page. • If you decide to swing by on a whim and my office door is open, feel free to bug me
Reviewed?	☐
Textbook File	https://artint.info/3e/html/ArtInt3e.html

Masters CS Northeastern DB*

Task	Week	Class	Progress	Date	Due Date	Rea
Syllabus	1	🧠 Anaconda Setup	Submitted!			
Homework	1	🧠 Problem Set 0: Background	Submitted!		@January 17, 2025	



AI Review

Starting EDA PORJECT AP

RAW DATA LOCATIONS in Google Drive

```
df = pd.read_csv("/content/drive/MyDrive/Colab_Notebooks/FAI_Spring_2025_Project/framingham_original/framingham/ra
```

```
"/content/drive/MyDrive/Colab_Notebooks/FAI_Spring_2025_Project/UCI/raw_data_download/processed.cleveland.data"
```

▼ UCI Git Documentation - LINK

`ucimlrepo` **package**

Package to easily import datasets from the UC Irvine Machine Learning Repository into scripts and notebooks.

Current Version: 0.0.7

Installation

In a Jupyter notebook, install with the command

```
!pip3 install -U ucimlrepo
```

Restart the kernel and import the module `ucimlrepo`.

Example Usage

```
from ucimlrepo import fetch_ucirepo, list_available_datasets

# check which datasets can be imported
list_available_datasets()

# import dataset
heart_disease = fetch_ucirepo(id=45)
# alternatively: fetch_ucirepo(name='Heart Disease')

# access data
X = heart_disease.data.features
y = heart_disease.data.targets
# train model e.g. sklearn.linear_model.LinearRegression().fit(X, y)

# access metadata
print(heart_disease.metadata.uci_id)
print(heart_disease.metadata.num_instances)
print(heart_disease.metadata.additional_info.summary)

# access variable info in tabular format
print(heart_disease.variables)
```

`fetch_ucirepo`

Loads a dataset from the UCI ML Repository, including the dataframes and metadata information.

Parameters

Provide either a dataset ID or name as keyword (named) arguments. Cannot accept both.

- `id`: Dataset ID for UCI ML Repository
- `name`: Dataset name, or substring of name

Returns

- `dataset`
  - `data`: Contains dataset matrices as **pandas** dataframes
    - `ids`: Dataframe of ID columns
    - `features`: Dataframe of feature columns
    - `targets`: Dataframe of target columns
    - `original`: Dataframe consisting of all IDs, features, and targets
    - `headers`: List of all variable names/headers
  - `metadata`: Contains metadata information about the dataset
    - See Metadata section below for details
  - `variables`: Contains variable details presented in a tabular/dataframe format
    - `name`: Variable name
    - `role`: Whether the variable is an ID, feature, or target
    - `type`: Data type e.g. categorical, integer, continuous
    - `demographic`: Indicates whether the variable represents demographic data

    - `description`: Short description of variable
    - `units`: variable units for non-categorical data
    - `missing_values`: Whether there are missing values in the variable's column

`list_available_datasets`

Prints a list of datasets that can be imported via `fetch_ucirepo`

Parameters

- **`filter`**: Optional keyword argument to filter available datasets based on a category
  - Valid filters: `aim-ahead`
- **`search`**: Optional keyword argument to search datasets whose name contains the search query

Returns

none

Metadata

- `uci_id`: Unique dataset identifier for UCI repository
- `name`
- `abstract`: Short description of dataset
- `area`: Subject area e.g. life science, business
- `task`: Associated machine learning tasks e.g. classification, regression
- `characteristics`: Dataset types e.g. multivariate, sequential
- `num_instances`: Number of rows or samples
- `num_features`: Number of feature columns
- `feature_types`: Data types of features
- `target_col`: Name of target column(s)
- `index_col`: Name of index column(s)
- `has_missing_values`: Whether the dataset contains missing values
- `missing_values_symbol`: Indicates what symbol represents the missing entries (if the dataset has missing values)
- `year_of_dataset_creation`
- `dataset_doi`: DOI registered for dataset that links to UCI repo dataset page
- `creators`: List of dataset creator names
- `intro_paper`: Information about dataset's published introductory paper
- `repository_url`: Link to dataset webpage on the UCI repository
- `data_url`: Link to raw data file
- `additional_info`: Descriptive free text about dataset
  - `summary`: General summary
  - `purpose`: For what purpose was the dataset created?
  - `funding`: Who funded the creation of the dataset?
  - `instances_represent`: What do the instances in this dataset represent?
  - `recommended_data_splits`: Are there recommended data splits?
  - `sensitive_data`: Does the dataset contain data that might be considered sensitive in any way?
  - `preprocessing_description`: Was there any data preprocessing performed?

- `variable_info`: Additional free text description for variables
- `citation`: Citation Requests/Acknowledgements

- `external_url`: URL to external dataset page. This field will only exist for linked datasets i.e. not hosted by UCI

Links

- UCI Machine Learning Repository home page
- PyPi repository for this package
- Submit an issue

▼ UCI Dataset - 16 attributes - Link - LINK - UCI Github

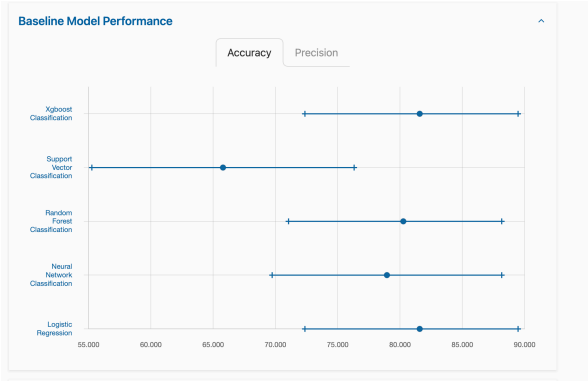
UCI Heart Disease Data Heart Disease Data Set from UCI data repository

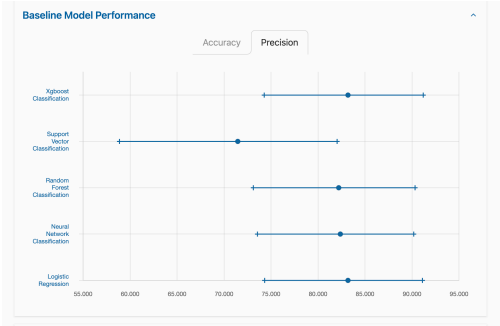
Data CardCode (151)Discussion (3)Suggestions (0)

```
pip install ucimlrepo


from ucimlrepo import fetch_ucirepo

# fetch dataset
heart_disease = fetch_ucirepo(id=45)

# data (as pandas dataframes)
X = heart_disease.data.features
y = heart_disease.data.targets

# metadata
print(heart_disease.metadata)

# variable information
print(heart_disease.variables)
```

About Dataset

This database contains 76 attributes, but all published experiments refer to using a subset of 14 of them. In particular, the Cleveland database is the only one that has been used by ML researchers to date. The "goal" field refers to the presence of heart disease in the patient. It is integer valued from 0 (no presence) to 4. Experiments with the Cleveland database have concentrated on simply attempting to distinguish presence (values 1,2,3,4) from absence (value 0).





Context

This is a multivariate type of dataset which means providing or involving a variety of separate mathematical or statistical variables, multivariate numerical data analysis. It is composed of 14 attributes which are age, sex, chest pain type, resting blood pressure, serum cholesterol, fasting blood sugar, resting electrocardiographic results, maximum heart rate achieved, exercise-induced angina, oldpeak — ST depression induced by exercise relative to rest, the slope of the peak exercise ST segment, number of major vessels and Thalassemia. This database includes 76 attributes, but all published studies relate to the use of a subset of 14 of them. The Cleveland database is the only one used by ML researchers to date. One of the major tasks on this dataset is to predict based on the given attributes of a patient that whether that particular person has heart disease or not and other is the experimental task to diagnose and find out various insights from this dataset which could help in understanding the problem more.

Content

Column Descriptions:

1. `id` (Unique id for each patient)
2. `age` (Age of the patient in years)
3. `origin` (place of study)
4. `sex` (Male/Female)
5. `cp` chest pain type ([typical angina, atypical angina, non-anginal, asymptomatic])
6. `trestbps` resting blood pressure (resting blood pressure (in mm Hg on admission to the hospital))
7. `chol` (serum cholesterol in mg/dl)
8. `fbs` (if fasting blood sugar > 120 mg/dl)
9. `restecg` (resting electrocardiographic results)- Values: [normal, stt abnormality, lv hypertrophy]
10. `thalach`: maximum heart rate achieved
11. `exang`: exercise-induced angina (True/ False)
12. `oldpeak`: ST depression induced by exercise relative to rest
13. `slope`: the slope of the peak exercise ST segment
14. `ca`: number of major vessels (0-3) colored by fluoroscopy
15. `thal`: [normal; fixed defect; reversible defect]
16. `num`: the predicted attribute

Acknowledgements

Creators:

1. Hungarian Institute of Cardiology. Budapest: Andras Janosi, M.D.
2. University Hospital, Zurich, Switzerland: William Steinbrunn, M.D.
3. University Hospital, Basel, Switzerland: Matthias Pfisterer, M.D.
4. V.A. Medical Center, Long Beach and Cleveland Clinic Foundation: Robert Detrano, M.D., Ph.D.

Relevant Papers:

- Detrano, R., Janosi, A., Steinbrunn, W., Pfisterer, M., Schmid, J., Sandhu, S., Guppy, K., Lee, S., & Froelicher, V. (1989). International application of a new probability algorithm for the diagnosis of coronary artery disease. American Journal of Cardiology, 64,304--310.Web Link
- David W. Aha & Dennis Kibler. "Instance-based prediction of heart-disease presence with the Cleveland database." Web Link
- Gennari, J.H., Langley, P, & Fisher, D. (1989). Models of incremental concept formation. Artificial Intelligence, 40, 11--61. Web Link

https://www.kaggle.com/datasets/rishidamarla/heart-disease-prediction

https://www.kaggle.com/datasets/mfarhaannazirkhan/heart-dataset

https://www.kaggle.com/datasets/johnsmith88/heart-disease-dataset/data

▼ Merged Dataset from 4 DBs, 76 attributes with 14 published ones - LINK

https://www.kaggle.com/datasets/johnsmith88/heart-disease-dataset/data

Context

This data set dates from 1988 and consists of four databases: Cleveland, Hungary, Switzerland, and Long Beach V. It contains 76 attributes, including the predicted attribute, but all published experiments refer to using a subset of 14 of them. The "target" field refers to the presence of heart disease in the patient. It is integer valued 0 = no disease and 1 = disease.

Content

Attribute Information:

> agesexchest pain type (4 values)resting blood pressureserum cholestoral in mg/dlfasting blood sugar > 120 mg/dlresting electrocardiographic results (values 0,1,2)maximum heart rate achievedexercise induced anginaoldpeak = ST depression induced by exercise relative to restthe slope of the peak exercise ST segmentnumber of major vessels (0-3) colored by flourosopythal: 0 = normal; 1 = fixed defect; 2 = reversable defectThe names and social security numbers of the patients were recently removed from the database, replaced with dummy values.

- From processed.cleveland.data, 14 columns

---

10 usability - LINK

Context

Cardiovascular diseases (CVDs) are the number 1 cause of death globally, taking an estimated 17.9 million lives each year, which accounts for 31% of all deaths worldwide. Four out of 5CVD deaths are due to heart attacks and strokes, and one-third of these deaths occur prematurely in people under 70 years of age. Heart failure is a common event caused by CVDs and this dataset contains 11 features that can be used to predict a possible heart disease.

People with cardiovascular disease or who are at high cardiovascular risk (due to the presence of one or more risk factors such as hypertension, diabetes, hyperlipidaemia or already established disease) need early detection and management wherein a machine learning model can be of great help.

Attribute Information

1. Age: age of the patient [years]
2. Sex: sex of the patient [M: Male, F: Female]
3. ChestPainType: chest pain type [TA: Typical Angina, ATA: Atypical Angina, NAP: Non-Anginal Pain, ASY: Asymptomatic]
4. RestingBP: resting blood pressure [mm Hg]
5. Cholesterol: serum cholesterol [mm/dl]
6. FastingBS: fasting blood sugar [1: if FastingBS > 120 mg/dl, 0: otherwise]
7. RestingECG: resting electrocardiogram results [Normal: Normal, ST: having ST-T wave abnormality (T wave inversions and/or ST elevation or depression of > 0.05 mV), LVH: showing probable or definite left ventricular hypertrophy by Estes' criteria]
8. MaxHR: maximum heart rate achieved [Numeric value between 60 and 202]
9. ExerciseAngina: exercise-induced angina [Y: Yes, N: No]
10. Oldpeak: oldpeak = ST [Numeric value measured in depression]
11. ST_Slope: the slope of the peak exercise ST segment [Up: upsloping, Flat: flat, Down: downsloping]
12. HeartDisease: output class [1: heart disease, 0: Normal]

FAI Project DB*

# Feature#	Aa Variable Name	Description	Dataset	Role	Type	Units	Missing Values	Dtype - raw	Demographic
1	age		Both	Feature - Main 14	Integer	years	☐	int64	Age

# Feature#	Aa Variable Name	Description	Dataset	Role	Type	Units	Missing Values	Dtype - raw	Demographic
2	sex		Both	Feature - Main 14	Categorical		☐	int64	Sex
3	cp	Chest Pain	UCI	Feature - Main 14	Categorical		☐	int64	
4	trestbps	resting blood pressure (on admission to the hospital)	Both	Feature - Main 14	Integer	mm Hg	☐	object	For Framingham, can use Sys/Diastolic BP
5	chol	serum cholestoral	Both	Feature - Main 14	Integer	mg/dl	☐	object	
6	fbs	fasting blood sugar > 120 mg/dl	Both	Feature - Main 14	Categorical		☐	object	
7	restecg	0: normal; 1: abnormality; 2: probably LV hypertrophy	Both	Feature - Main 14	Categorical		☐	object	
8	thalach	maximum heart rate achieved	Both	Feature - Main 14	Integer		☐	object	
9	exang	exercise induced angina	UCI	Feature - Main 14	Categorical		☐	object	
10	oldpeak	ST depression induced by exercise relative to rest	UCI	Feature - Main 14	Integer		☐	float64	
11	slope		UCI	Feature - Main 14	Categorical		☐	object	
12	ca	number of major vessels (0-3) colored by flourosopy	UCI	Feature - Main 14	Integer		☑	object	
13	thal		UCI	Feature - Main 14	Categorical		☑	object	
14	num/10YearCHD	diagnosis of heart disease	Both	Target	Integer		☐		

Source

This dataset was created by combining different datasets already available independently but not combined before. In this dataset, 5 heart datasets are combined over 11 common features which makes it the largest heart disease dataset available so far for research purposes. The five datasets used for its curation are:

- Cleveland: 303 observations

- Hungarian: 294 observations
- Switzerland: 123 observations
- Long Beach VA: 200 observations
- Stalog (Heart) Data Set: 270 observations

Total: 1190 observations

Duplicated: 272 observations

Final dataset: 918 observations

Every dataset used can be found under the Index of heart disease datasets from UCI Machine Learning Repository on the following link: https://archive.ics.uci.edu/ml/machine-learning-databases/heart-disease/

Citation

> fedesoriano. (September 2021). Heart Failure Prediction Dataset. Retrieved [Date Retrieved] from https://www.kaggle.com/fedesoriano/heart-failure-prediction.

https://www.kaggle.com/code/azizozmen/heart-failure-predict-8-classification-techniques

https://www.kaggle.com/code/fahadrehman07/heart-disease-prediction-using-9-models

https://www.kaggle.com/code/tanmay111999/heart-failure-prediction-cv-score-90-5-models

UCI Features

UCI Dataset

▼ Brandon Simmons paper

> For a number of unknown reasons to this researcher, the original raw source data became corrupted and was never re-uploaded to the repository.

> Using Random Forest and logistic regression, we determine the best dataset for machine learning and statistical analysis: the Cleveland data on a reduced set of six features.
>
> Three of which are statistically significant at explaining or classifying patients as "Heart Disease". They are thalach (maximum heart rate), oldpeak, and cp (chest pain).

> Our work shows how to fix the Kaggle data as it becomes essentially the Cleveland data. We determine if there is any relation between the Statlog data and the Cleveland data.
>
> Then, we identify both of these data from the most recently published (combined) data on heart disease.

Finally, we analyze these data using logistic regression along with random forest feature selections to determine the best models with the least AI criterion. We conclude that, the Cleveland data on a reduced (features) model is the best statistical model.

Approach

1. although the data is supposed to have originated form the University of California Irvine Machine Learning Repository archive, the predictor's encoding is backward. This lead to misleading interpretations and incorrect conclusions about the features relation with the target variable. However, this data is very similar to Cleveland data which is also found in the UCI repository. Except, the Cleveland data has 6 missing values and the target variable has 5 levels. With the right encoding, we conclude that Kaggle (with target 1=No disease; 0=Disease) is the Cleveland. Still, we found another complete dataset in another archive of the UCI repository, called Stalog which is similar to Cleveland except in size.
2. Compare the Cleveland data (303 obs.) to the Statlog data (270 obs.). We use "Pandas Profiling" to access overall data content such as size and variable names, types and their distributions. Then, we compare pairwise, the distribution of each feature, and each feature against the target variable. We see very similar associations between each feature and the target and upon further investigations, we found that the Statlog data is a proper subset of the Cleveland data.
3. Find the remaining three datasets that were corrupted in the UCI dataset repository. To do so, we needed to first identify the Cleveland (and thus the Statlog datasets) from the most recently published data, ORIG—We merge the Cleveland data with ORIG data and remove all duplicated records. In which case, any record that appears in both Cleveland, and ORIG is removed. The remaining dataset is called MISS and perhaps represents the (3) missing datasets that were deemed corrupt in the UCI repository. Unfortunately, we are unable to authenticate the origin of the ORIG data. Despite our attempt to reach out to the originator, we received no response. So, we proceed to further investigate the content of the MISS data. We found that about 25% (303 observations) are duplicated records.
4. Investigate the MISS data by exploring the distribution of its features, and the features associations with the target variable. We found that the data has skewed significantly the gender, and unusually high records of '0' values for cholesterol level.
5. Analyze each of the three datasets: CLEVELAND, MISS, and STATLOG. We build a logistic regression on each of the data, and using Random Forest algorithm we select the most important features from each data against the target. The process led to building some reduced models and determine the best model for statistical analysis.

1. Criteria 1: Blood Flow

- CAD is defined as the obstruction of the blood flow
  to the heart muscles through the coronary arteries by plaque accumulation on the walls which indefinitely leads to heart attacks[12]. The CAD mortality rate visually

> presents the stats for 35 and older for both sexes and all ethnicity [3]. North Carolina falls in the range 148.6 - 167.9 with the estimated value of 165.2.

> Criteria 2: Structure - Heart Valve Disease (HVD)
>
> - Heart Valve Disease (HVD) involves the issues occurring with the four values that renew and direct blood flow for a smooth transition within the circulatory system[12]. In essence, when one or more of the valves is improperly functions then the patient's condition is classified as HVD. Symptoms include stenosis, which is the limiting of muscle mobility due to valves fusing; Valve Leakage which causes blood to flow back into the heart; and lastly Atresia, which absence of valve opening

> Criteria 3: Overall Function - Arrhythmias & Heart Failure

Contributing Factors

> main criteria that is represent in the data sets are blood flow. It is apparent these following factors contribute to the number of heart disease cases within the United States in general. The factors of age, sex, and geographic location increase the risk of a heart attack or stroke[19]. It is also interesting that our heart age may be higher than our actual age due to the influences of diet, stress, activity and heritage.
>
> Until the age of 45 years old and older, medical personnel prescribe the public to have a routine heart health check at least every two years. The condition of participants within the data sets pertains to the data retrieved from testing. At the time of data collection traditional risk factors for CAD are cholesterol levels, blood pressure, blood sugar, and pain developed from exercise recorded during the thallium test.

UCI Features (Simmons)

Heart Disease Attributes		
Number	Name	Information
1	Age Label – (age)	Measured in years
2	Sex Label – (sex)	1 = male 0 = female
3	Chest Pain Label – (cp)	Value 1: typical angina Value 2: atypical angina Value 3: non-anginal pain Value 4: asymptomatic
4	Resting Blood Pressure Label – (trestbps)	Measured in mm Hg
5	Cholesterol Label – (chol)	Measured in mg/dl.
6	Fasting Blood Sugar Label – (fbs)	If greater than 120 mg/dl (normal): Value 1 = true. Value 0 = false
7	Resting Electrocardiogram Label – (restecg)	Value 0: normal. Value 1: abnormality (Stress Test elevation or depression of > 0.05 mV) Value 2: Probable or definite left ventricular hypertrophy by Estes' criteria
8	Maximum Heart Label – (thalach)	Heart rate achieved during the patient's Stress Testing
9	Exercise Induced Angina Label – (exang)	Value 1 = yes Value 0 = no
10	Stress Test Depression Label – (oldpeak)	Numeric value signifying the marker for adverse cardiac events.
11	Slope for Peak Exercise Label – (slope)	Value 1: up sloping Value 2: flat Value 3: down sloping
12	Number of major vessels Label – (ca)	Number of major vessels (0-3) colored by fluoroscopy
13	Thallium Heart Rate Label – (thal)	Value 3 = normal Value 6 = fixed defect Value 7 = reversable defect
14	Heart disease prediction Label – (target)	Value 0: < 50% diameter narrowing. (Absence) Value 1 to 4: > 50% diameter narrowing. (Presence)

Thesis Heart Disease Attributes		
Number	Name	Information
1	Age Label – (age)	Measured in years
2	Sex Labeled – (sex)	Value 1 = male Value 0 = female
3	Chest Pain Label – (cp)	Value 1: typical angina Value 2: atypical angina Value 3: non-anginal pain Value 4: asymptomatic
4	Resting Blood Pressure Label – (trestbps)	Measured in mm Hg
5	Cholesterol Label – (chol)	Measured in mg/dl.
6	Fasting Blood Sugar Label – (fbs)	If greater than 120 mg/dl (normal): Value 1 = true. Value 0 = false
7	Resting Electrocardiogram Label – (restecg)	Value 0: normal. Value 1: abnormality (Stress Test elevation or depression of > 0.05 mV) Value 2: Probable or definite left ventricular hypertrophy by Estes' criteria
8	Maximum Heart Label – (thalach)	Heart rate achieved during the patient's Stress Testing
9	Exercise Induced Angina Label – (exang)	Value 1 = yes Value 0 = no
10	Stress Test Depression Label – (oldpeak)	Numeric value signifying the marker for adverse cardiac events.
11	Slope for Peak Exercise Label – (slope)	Value 1: up sloping Value 2: flat Value 3: down sloping
12	Heart disease prediction Label – (target)	Value 0: No Heart Disease Value 1: Heart Disease

2.2.3 Chest Pain

The most important indicator of heart disease is the chest discomfort or chest angina [12]. Because discomfort provides the first indication of beginning symptoms for many disease, chest angina establishes there is a problem present. The data tables below angina levels may be case by case due to pain tolerance [12]. Although this is one of the most dangerous symptoms of many diseases, this attribute can act as a control for initial diagnoses verse actual diagnoses.

Kaggle				
Chest Pain	Pain	Female	Male	Total
Value 0	Asymptomatic	39	104	143
Value 1	Non-anginal Pain	18	32	50
Value 2	Atypical Angina	35	52	87
Value 3	Typical Angina	4	19	23
		96	207	303

2.2.6 Fasting Blood Sugar

Blood Glucose or commonly recognized as Blood Sugar Diabetes is a disease that occurs when your blood glucose, also called blood sugar, is too high [15]. Milligrams per deciliter, is a measurement that indicates the amount of glucose in a specific amount of blood. According to the CDC, about 1 in 4 people with diabetes don't know they have the disease. For the data sets, blood sugar is distinguished whether the patient's blood sugar is higher than 120 mg/dl or not.

Kaggle			
Fasting Blood Sugar Levels	Female	Male	Total
Normal: < 120 mg/dL	84	174	258
Risk: 120 ≤ mg/dL	12	33	45
	96	207	303

2.2.1 Age

Because heart health risk increases with age, the screening age range for participants is greater than of equal to 35. Although heart risk is not typical for younger age groups, Heart Foundation(HF) doesn't detour younger participants especially family's that have a history with heart health risks [7].

Minimum	Q1	Median	Q3	Maximum	Mean	Range	Standard Divaition
29	48	56	61	77	54.49	48	9.04

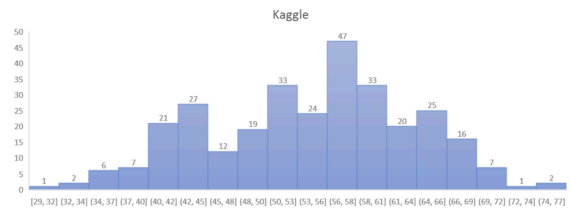


Figure 2.1: Bin Width Kaggle Age Summary

Kaggle				
Blood Pressure Levels	Systolic mm Hg (Higher Number)	Female	Male	Total
Normal	< 120	19	41	60
Elevated	120 – 129	17	58	75
Stage 1	130 – 139	28	43	71
Stage 2	140 – 180	31	64	95
Critical	180 <	1	1	2
		96	207	303

Table 2.7: Resting blood pressure gender comparison

Kaggle			
Fasting Blood Sugar Levels	Female	Male	Total
Normal: < 120 mg/dL	84	174	258
Risk: 120 ≤ mg/dL	12	33	45
	96	207	303

Kaggle			
Total Cholesterol Levels	Female	Male	Total
Normal: < 200	14	36	50
Risk: 200 -239	24	74	98
High Risk: 240 >	58	97	155
	96	207	303

Kaggle			
Resting ECG Results	Female	Male	Total
Value 0: Normal	44	103	147
Value 1: Abnormality	49	103	152
Value 2: Hypertrophy	3	1	4
	96	207	303

2.2.8 Heart Rate

The maximum heart rate was recorded during Thallium stress test. The data set showcase the optimal maximum healthy heart rate depends on the age (220 - age). Thus, higher rates tend to be from younger patients.

2.2.9 Exercise Pain

This attribute is the patient's level of angina or induced pain during exercise which is necessary input for the presence of heart disease.

2.2.10 Old Peak

The resting Stress Test segment depression is the marker for adverse cardiac events. The old peak monitors a certain level in a normal heart beat which indicates a displacement for the presence of a heart disease.

2.2.11 Slope

The part of the Stress Test for indicators of exercise. The slope by itself can help determine whether there is heart disease or not if it is flat or ascending. Adding a third variable where we can see if the slope is descending, the depression of the ST segment can help to determine if the patient has a heart disease.

"Thallium Stress Test" also known as "Nuclear Stress Test" or "Cardiac Test"
benefits heart disease research by analyzing the condition of blood flow [14]. The
gamma camera through nuclear imaging tracks the participant's blood flow which
carries a sample amount of Thallium, radioactive isotope

2.2.12 Target

The target is designated as the condition of the patient for heart disease after the conducting stress testing indicators. The results of the testing had an extensive range that determined for the disease presence. For the sake of simplicity and design of approach, the patients which had any indication of disease present then in this analysis it was considered diagnosed as heart disease.

Cleveland vs Kaggle Data (Simmons)

The original Cleveland data is called processed.cleveland.data, which can be downloaded from the UCI Machine Learning repository site (in "Data Folder") and exported as a txt file. It was donated in July 1988 and currently has 1,471,444 web hits. This data has no header, and yet the headers information were provided on the website (under Attribute Information)—There are 303 records and 14 features (5 numerical, 9 categorical) information were provided which helped determine each column header in the order listed in the Table B, below. Although the Cleveland data features share similar distributions to those of Kaggle (see Appendix I), there are a couple of differences that we list below:
(a) Contrary to the Kaggle data, the original Cleveland data has 6 missing values (recorded as '?'): 4 missing values are found for the number of blood vessels (ca) feature and 2 missing values for the thalassemia (thal) feature. Because each feature is categorical, it appears that, each of their missing values was replaced in the Kaggle data with a closest value generated by the Nearest Neighbor Algorithm. However, since these two features do not appear in a larger dataset (ORIG) which we explore later, for practical comparative analysis, decided to drop them.

AP Analysis

cleaned_heart.csv

```
[19]  1  df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 918 entries, 0 to 917
Data columns (total 12 columns):
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   Age             918 non-null    int64
 1   Sex             918 non-null    object
 2   ChestPainType   918 non-null    object
 3   RestingBP       918 non-null    int64
 4   Cholesterol     918 non-null    int64
 5   FastingBS       918 non-null    int64
 6   RestingECG      918 non-null    object
 7   MaxHR           918 non-null    int64
 8   ExerciseAngina  918 non-null    object
 9   Oldpeak         918 non-null    float64
 10  ST_Slope        918 non-null    object
 11  HeartDisease    918 non-null    int64
dtypes: float64(1), int64(6), object(5)
memory usage: 86.2+ KB
```

```
1  df.shape
```

```
(918, 12)
```

▼ raw_merged_heart_dataset.csv (81kb) Analysis

[21]
```python
df = pd.read_csv("/content/drive/MyDrive/Colab_Notebooks/FAI_Spring_2025_Project/UCI/data_raw/raw_merged_heart_dataset.csv")
#check if the dataset is loaded
df.head()
```

	age	sex	cp	trestbps	chol	fbs	restecg	thalachh	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

Next steps: Generate code with df | View recommended plots | New interactive sheet

```python
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2181 entries, 0 to 2180
Data columns (total 14 columns):
 #   Column    Non-Null Count  Dtype
---  ------    --------------  -----
 0   age       2181 non-null   int64
 1   sex       2181 non-null   int64
 2   cp        2181 non-null   int64
 3   trestbps  2181 non-null   object
 4   chol      2181 non-null   object
 5   fbs       2181 non-null   object
 6   restecg   2181 non-null   object
 7   thalachh  2181 non-null   object
 8   exang     2181 non-null   object
 9   oldpeak   2181 non-null   float64
 10  slope     2181 non-null   object
 11  ca        2181 non-null   object
 12  thal      2181 non-null   object
 13  target    2181 non-null   int64
dtypes: float64(1), int64(4), object(9)
memory usage: 238.7+ KB
```

```python
df.shape
```

```
(2181, 14)
```

```python
print(df.shape)
print('-'*20)
print(df.isnull().sum())
print('-'*20)
print(df.duplicated().sum())
```

```
(2181, 14)
--------------------
age         0
sex         0
cp          0
trestbps    0
chol        0
fbs         0
restecg     0
thalachh    0
exang       0
oldpeak     0
slope       0
ca          0
thal        0
target      0
dtype: int64
--------------------
1287
```

```
df.describe()
```

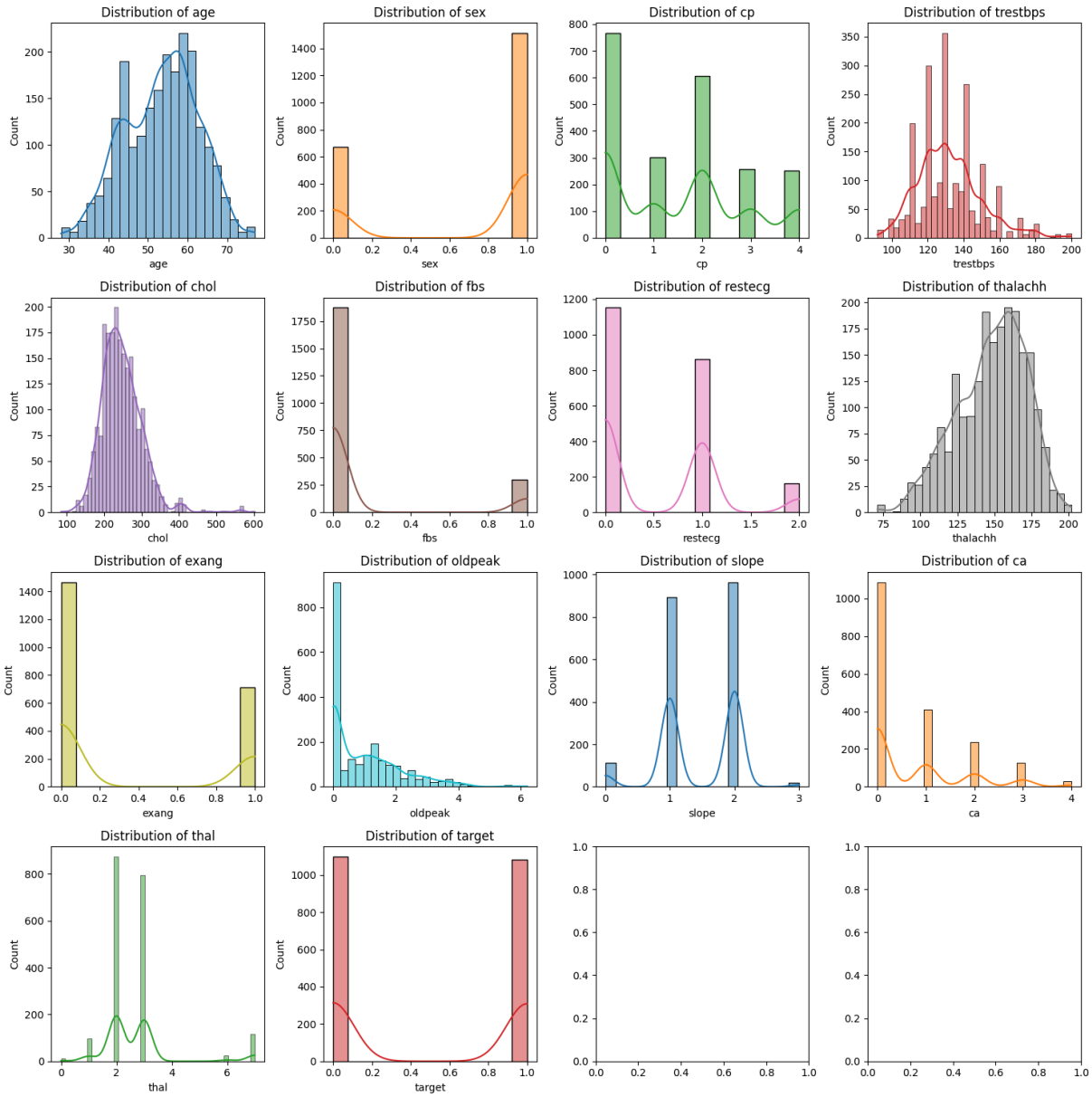
	age	sex	cp	oldpeak	target
count	2181.000000	2181.000000	2181.000000	2181.000000	2181.000000
mean	53.477762	0.693260	1.507565	0.990509	0.496103
std	9.194787	0.461246	1.371587	1.141851	0.500099
min	28.000000	0.000000	0.000000	0.000000	0.000000
25%	46.000000	0.000000	0.000000	0.000000	0.000000
50%	54.000000	1.000000	2.000000	0.600000	0.000000
75%	60.000000	1.000000	2.000000	1.600000	1.000000
max	77.000000	1.000000	4.000000	6.200000	1.000000

index	age	sex	cp	oldpeak	target
count	2181	2181	2181	2181	2181
mean	53.4777625	0.69325997	1.50756534	0.99050894	0.49610271
std	9.1947874	0.46124628	1.3715867	1.14185057	0.50009947
min	28	0	0	0	0
25%	46	0	0	0	0
50%	54	1	2	0.6	0
75%	60	1	2	1.6	1
max	77	1	4	6.2	1

▼ Visualization of Raw Dataset - 2181 rows

```
# Define features to plot
features = ['age', 'sex', 'cp', 'trestbps', 'chol', 'fbs', 'restecg', 'thalachh',
        'exang', 'oldpeak', 'slope', 'ca', 'thal', 'target']

# Use Seaborn's tab10 palette to generate colors (up to 10 colors)
colors = sns.color_palette("tab10", n_colors=len(features))  # Ensures you get enough colors for all features

# Create subplots (4×4 grid to match 16 features)
fig, axes = plt.subplots(4, 4, figsize=(15, 15))  # 4×4 grid

# Flatten the 2D axes array to 1D for easy iteration
```

```
axes = axes.flatten()

# Loop through features and plot each one
for i, feature in enumerate(features):
    sns.histplot(df[feature], kde=True, color=colors[i], ax=axes[i])
    axes[i].set_title(f"Distribution of {feature}")

plt.tight_layout()  # Adjust layout
plt.show()
```



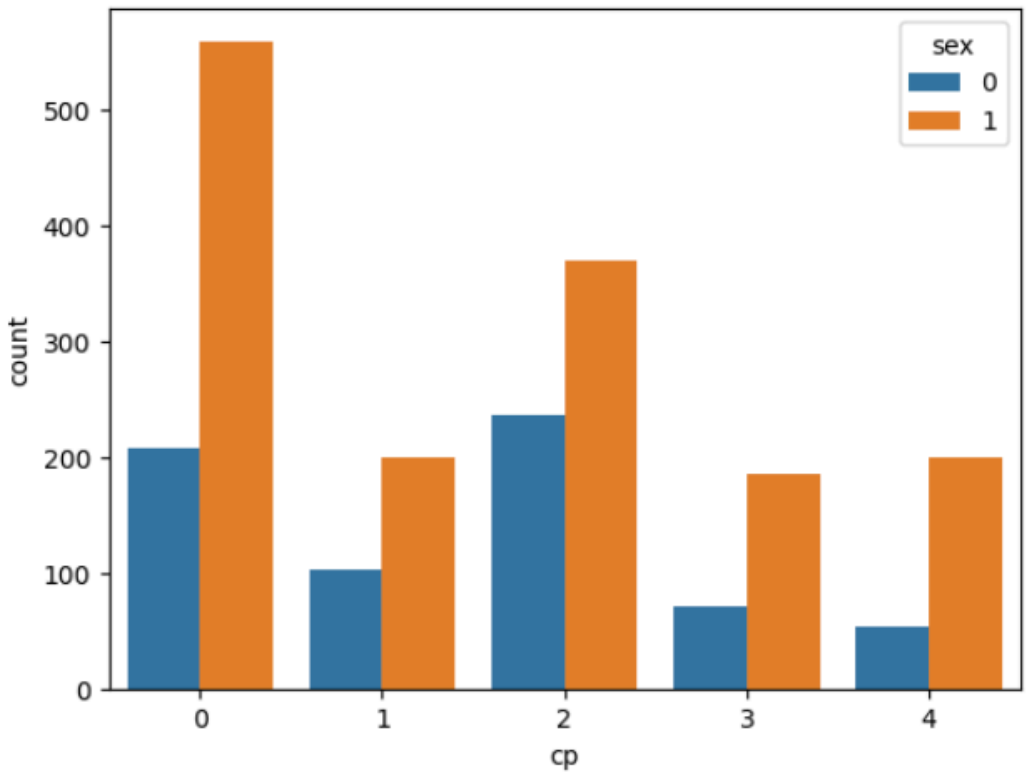
```python
# value count of cp column
df['cp'].value_counts()
```

cp	count
0	766
2	605
1	302
3	256
4	252

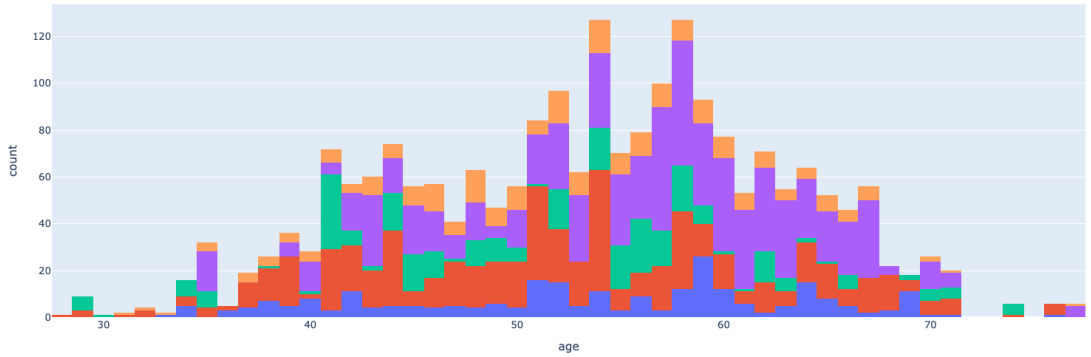
dtype: Int64

```python
# count plot of cp column by sex column
sns.countplot(df, x='cp', hue= 'sex')
```

```
<Axes: xlabel='cp', ylabel='count'>
```



```python
# Draw the plot of age column group by cp column

fig = px.histogram(data_frame=df, x='age', color='cp')
fig.show()
```



▼ Impute Missing Values

```
# Dealing with Missing values in trestbps column.
# find the percentage of misssing values in trestbps column
print(f"Percentage of missing values in trestbps column: {df['trestbps'].isnull().sum() /len(df) *100:.2f}%")
Percentage of missing values in trestbps column: 6.41%
Removing missing values using Iterative imputer

# Impute the missing values of trestbps column using iterative imputer
# create an object of iteratvie imputer
imputer1 = IterativeImputer(max_iter=10, random_state=42)

# Fit the imputer on trestbps column
imputer1.fit(df[['trestbps']])

# Transform the data
df['trestbps'] = imputer1.transform(df[['trestbps']])

# Check the missing values in trestbps column
print(f"Missing values in trestbps column: {df['trestbps'].isnull().sum()}")
```

```python
(df.isnull().sum()/ len(df)* 100).sort_values(ascending=False)
```

	0
ca	13.342503
thal	12.196240
slope	8.711600
chol	1.054562
fbs	0.366804
trestbps	0.045851
restecg	0.045851
thalachh	0.045851
exang	0.045851
age	0.000000
sex	0.000000
cp	0.000000
oldpeak	0.000000
target	0.000000

dtype: float64

Impute missing values using iterative imputer for selected columns.

columns are selected based on data types (floating data type) because imputer only works with the floating data types.

selected columns are:

- slope
- thal
- chol
- thalch

```
# create an object of iterative imputer
imputer2 = IterativeImputer(max_iter=10, random_state=42)

# fit transform on ca,oldpeak, thal,chol and thalch columns
df['ca'] = imputer2.fit_transform(df[['ca']])
df['thal'] = imputer2.fit_transform(df[['thal']])
df['slope'] = imputer2.fit_transform(df[['slope']])
df['chol'] = imputer2.fit_transform(df[['chol']])
```

After Imputation

```python
# let's check again for missing values
(df.isnull().sum()/ len(df)* 100).sort_values(ascending=False)
```

	0
fbs	0.366804
trestbps	0.045851
restecg	0.045851
thalachh	0.045851
exang	0.045851
age	0.000000
sex	0.000000
cp	0.000000
chol	0.000000
oldpeak	0.000000
slope	0.000000
ca	0.000000
thal	0.000000
target	0.000000

dtype: float64