Heart Disease Prediction (Classification) 🩺

Project Objective: To build a machine learning model that can accurately predict whether a patient has heart disease based on a set of medical attributes. This project will serve as a comprehensive introduction to classification, one of the most common types of machine learning problems.

Setup - Importing Libraries and Loading Data

```
Downloading dataset...
Using Colab cache for faster access to the 'heart-disease-data' dataset.
Dataset downloaded and loaded successfully
Dataset shape: (920, 16)
```

id	age	sex	dataset	cp	trestbps	chol	fbs	restecg	thalch	exang	oldpea
556	43	Male	Hungary	asymptomatic	120.0	175.0	False	normal	120.0	True	1.
813	67	Male	VA Long Beach	asymptomatic	120.0	0.0	True	normal	150.0	False	1.
306	29	Male	Hungary	atypical angina	140.0	NaN	False	normal	170.0	False	0.
365	43	Female	Hungary	atypical angina	120.0	215.0	False	st-t abnormality	175.0	False	0.
823	48	Male	VA Long Beach	non-anginal	102.0	0.0	NaN	st-t abnormality	110.0	True	1.
694	62	Male	Switzerland	asymptomatic	115.0	0.0	NaN	normal	128.0	True	2.
395	48	Female	Hungary	atypical angina	120.0	NaN	True	st-t abnormality	148.0	False	0.
358	42	Male	Hungary	atypical angina	120.0	198.0	False	normal	155.0	False	0.
29	43	Male	Cleveland	asymptomatic	150.0	247.0	False	normal	171.0	False	1.
793	65	Male	VA Long Beach	typical angina	NaN	252.0	False	normal	NaN	NaN	Na

Exploratory Data Analysis (EDA)

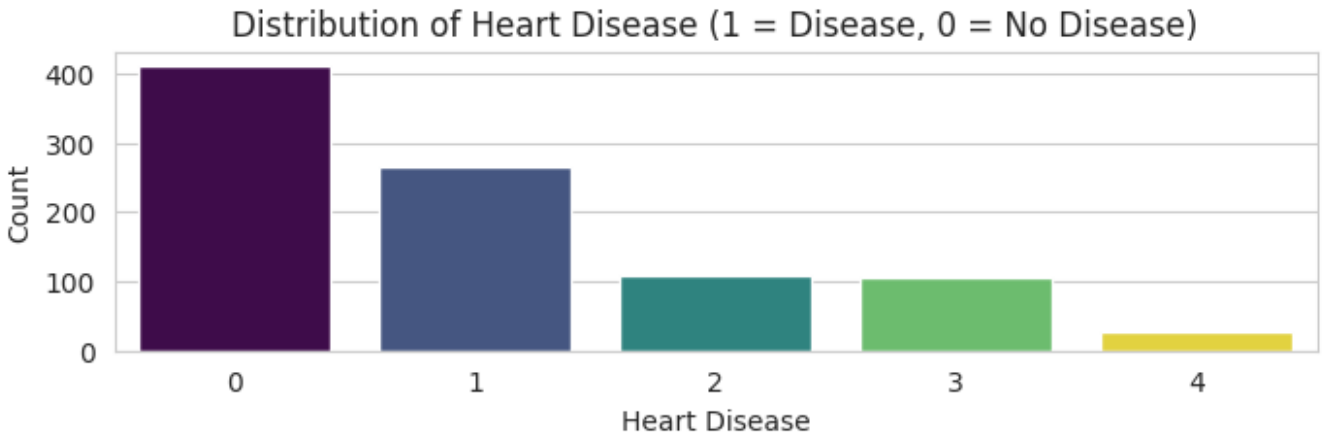
Before building any models, I need to understand our data deeply. I'll look at the distribution of our target variable, the characteristics of our features, and how they relate to the presence of heart disease.

```
Data information
<class 'pandas.core.frame.DataFrame'>
Index: 920 entries, 1 to 920
Data columns (total 15 columns):
 #   Column    Non-Null Count  Dtype
---  ------    --------------  -----
 0   age       920 non-null    int64
 1   sex       920 non-null    object
 2   dataset   920 non-null    object
 3   cp        920 non-null    object
 4   trestbps  861 non-null    float64
 5   chol      890 non-null    float64
 6   fbs       830 non-null    object
 7   restecg   918 non-null    object
 8   thalch    865 non-null    float64
 9   exang     865 non-null    object
 10  oldpeak   858 non-null    float64
 11  slope     611 non-null    object
 12  ca        309 non-null    float64
 13  thal      434 non-null    object
 14  num       920 non-null    int64
dtypes: float64(5), int64(2), object(8)
memory usage: 115.0+ KB
None

 Descreptive statistics
              age    trestbps        chol      thalch     oldpeak          ca  \
count  920.000000  861.000000  890.000000  865.000000  858.000000  309.000000
mean    53.510870  132.132404  199.130337  137.545665    0.878788    0.676375
std      9.424685   19.066070  110.780810   25.926276    1.091226    0.935653
min     28.000000    0.000000    0.000000   60.000000   -2.600000    0.000000
25%     47.000000  120.000000  175.000000  120.000000    0.000000    0.000000
50%     54.000000  130.000000  223.000000  140.000000    0.500000    0.000000
75%     60.000000  140.000000  268.000000  157.000000    1.500000    1.000000
max     77.000000  200.000000  603.000000  202.000000    6.200000    3.000000

              num
count  920.000000
mean     0.995652
std      1.142693
min      0.000000
25%      0.000000
50%      1.000000
75%      2.000000
max      4.000000

 Total missing value count
1759

 Missing values by cloumns
age           0
sex           0
dataset       0
cp            0
trestbps     59
```

```
chol         30
fbs          90
restecg       2
thalch       55
exang        55
oldpeak      62
slope       309
ca          611
thal        486
num           0
dtype: int64
```

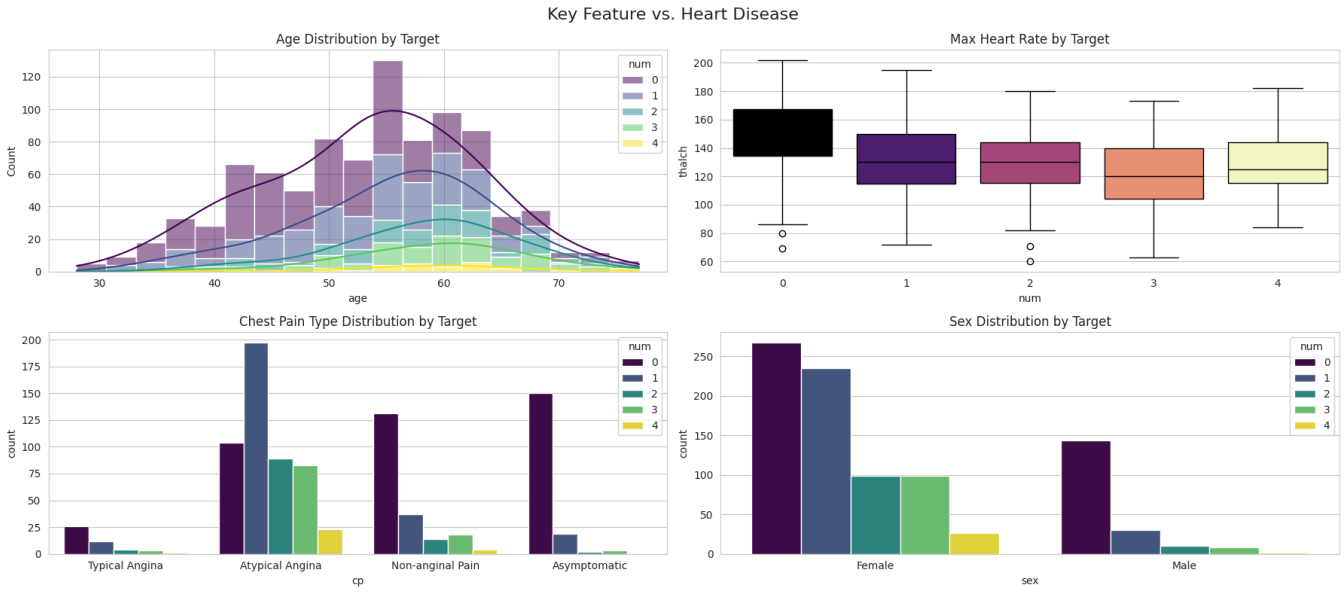
Analysing the target variable

Checking the distribution of the patient with and without the heart disease



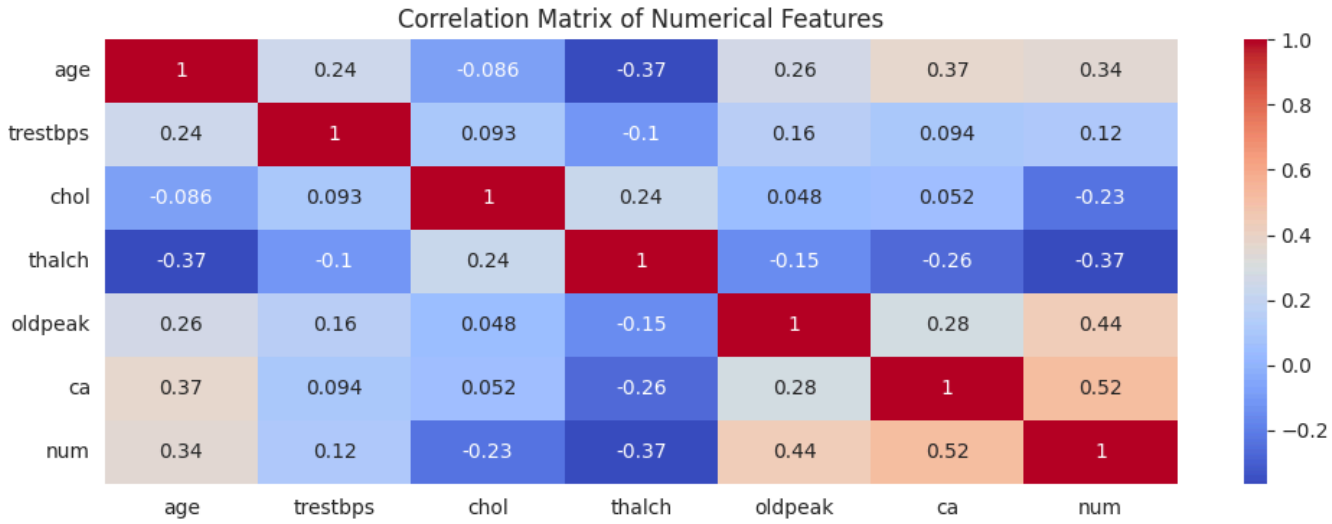
Insight: The dataset is fairly balanced, with a slightly higher number of patients having heart disease. This is good because it means our model will have a similar number of examples for both classes to learn from, and accuracy will be a meaningful metric.

Analysing feature vs target



Insights:

- **Max Heart Rate (thalach):** Patients with heart disease tend to have a lower maximum heart rate.
- **Chest Pain (cp):** Patients with chest pain types 1 and 2 (Atypical and Non-anginal) are more likely to have heart disease. Surprisingly, those with type 0 (Typical Angina) are less likely, and those with asymptomatic pain (type 3) are very likely to have the disease.
- **Sex:** A higher proportion of females in this dataset have heart disease compared to males.



Insight:

Patient with heart disease has high corellation with the ***number of major vessels (0-3) colored by fluoroscopy***, ***ST depression induced by exercise relative to rest*** and ***Age of the patient in years***

It show negatively corellated with ***maximum heart rate achieved*** and ***serum cholesterol in mg/dl***

Data preprocessing

- Separating features (X) and target (y).
- Identifying categorical features that need to be encoded.
- One-Hot Encoding categorical features to convert them into a numerical format.
- Scaling numerical features so they are on a similar scale.

- Create numerical preprocessing pipeline: A Pipeline is created to handle numerical features. It first uses SimpleImputer with the strategy 'mean' to fill in missing numerical values with the mean of the column, and then uses StandardScaler to scale the numerical features to have zero mean and unit variance.
- Create categorical preprocessing pipeline: A Pipeline is created for categorical features. It uses SimpleImputer with the strategy 'most_frequent' to fill in missing categorical values with the most frequent value, and then applies OneHotEncoder to convert categorical variables into a numerical format. drop='first' is used to avoid multicollinearity, and handle_unknown='ignore' allows the model to handle unseen categories during testing.

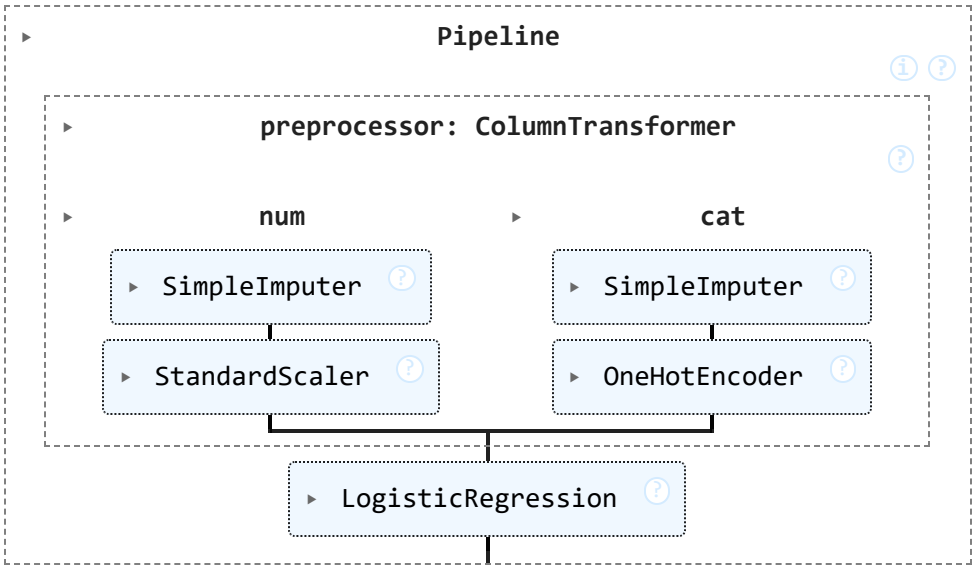
Model Building & Training

I am using four different models to train the data and then compare the best fit model.

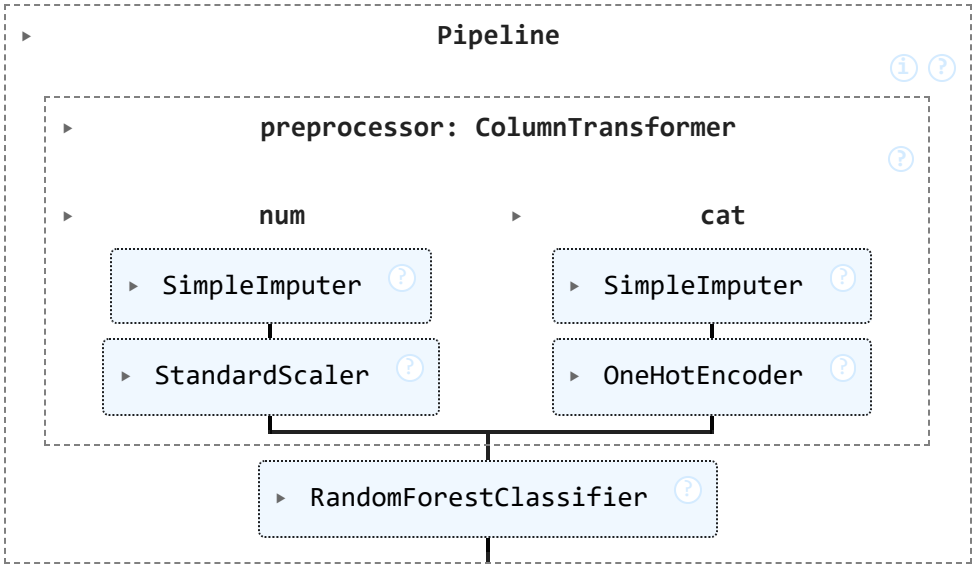
- Logisticregression
- Random forest
- Support vactor Machine
- K-Nearest Neighbour(KNN)

Model 1: Logistic Regression (Baseline)¶

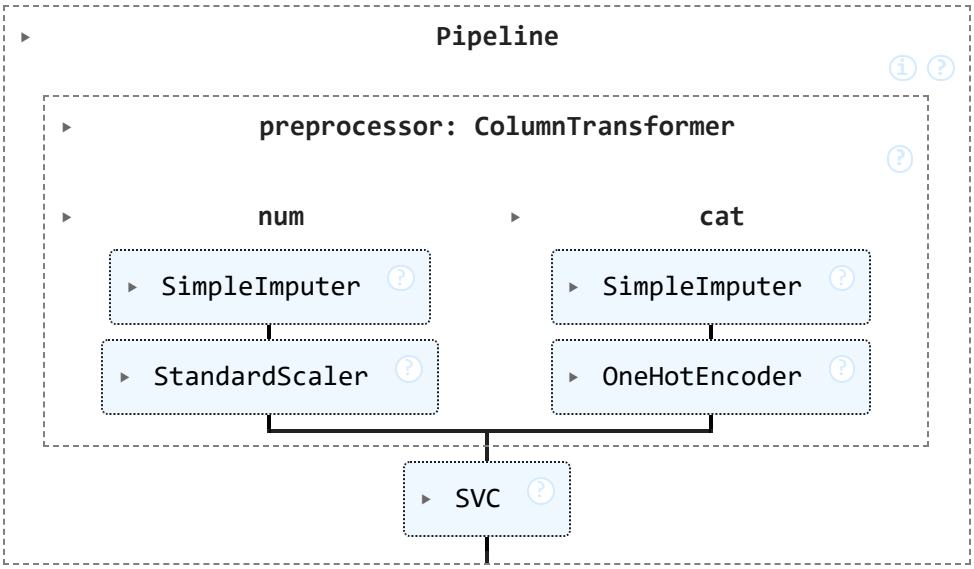
```
Numerical features: ['age', 'trestbps', 'chol', 'thalch', 'oldpeak', 'ca']
Categorical features: ['sex', 'cp', 'fbs', 'restecg', 'exang', 'slope', 'thal']
```



Model 2: Random Forest Classifier (Advanced)

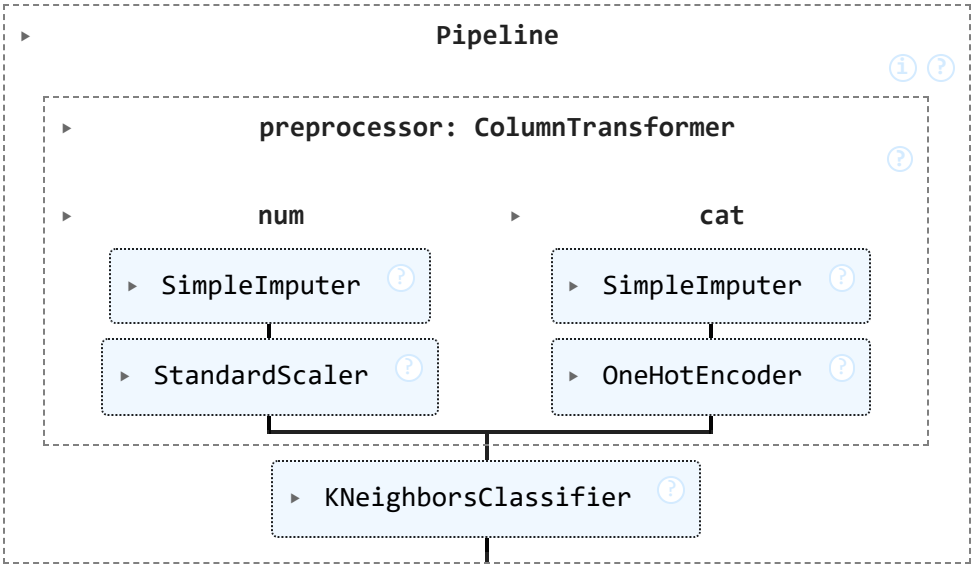


Model 3: Support Vector Machine (SVM)¶




```
The SVM kernel type is: rbf
```

Model 4: K-Nearest Neighbors (KNN)



Model Evaluation

Comparing the models based on the confusion metricx and f1 scores.

-----Model parameters for Logistic_Regression----

Accuracy: 0.5815217391304348

Confusion Matrix and F1 Score:

	precision	recall	f1-score	support
0	0.80	0.84	0.82	82
1	0.46	0.57	0.51	53
2	0.30	0.14	0.19	22
3	0.23	0.24	0.23	21
4	0.00	0.00	0.00	6
accuracy			0.58	184
macro avg	0.36	0.36	0.35	184
weighted avg	0.55	0.58	0.56	184

-----Model parameters for Random_Forest----

Accuracy: 0.5489130434782609

Confusion Matrix and F1 Score:

	precision	recall	f1-score	support
0	0.73	0.84	0.78	82
1	0.50	0.55	0.52	53
2	0.00	0.00	0.00	22
3	0.18	0.14	0.16	21
4	0.00	0.00	0.00	6
accuracy			0.55	184
macro avg	0.28	0.31	0.29	184
weighted avg	0.49	0.55	0.52	184

-----Model parameters for SVM----

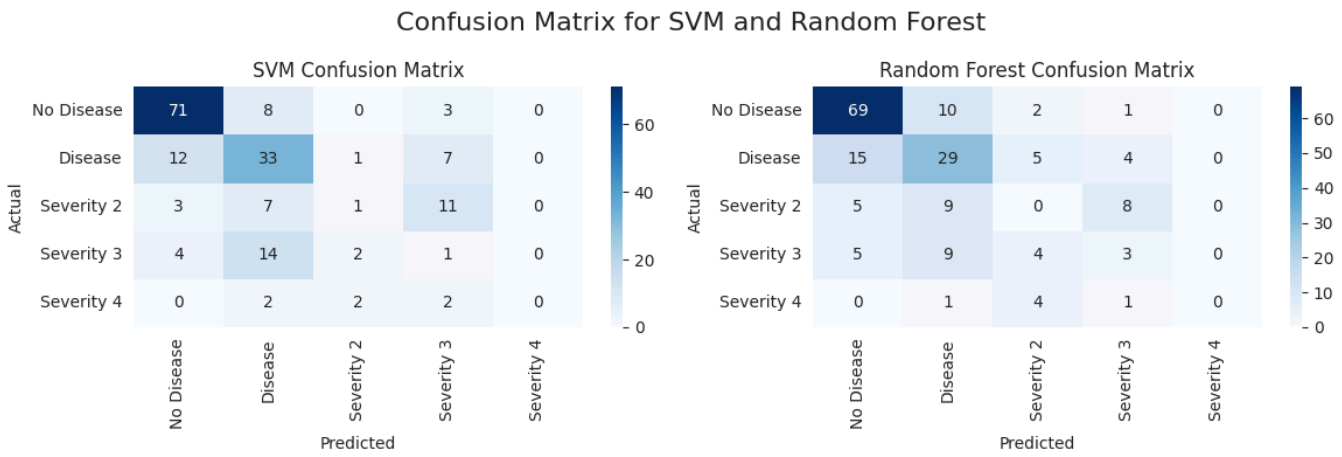
Accuracy: 0.5760869565217391

Confusion Matrix and F1 Score:

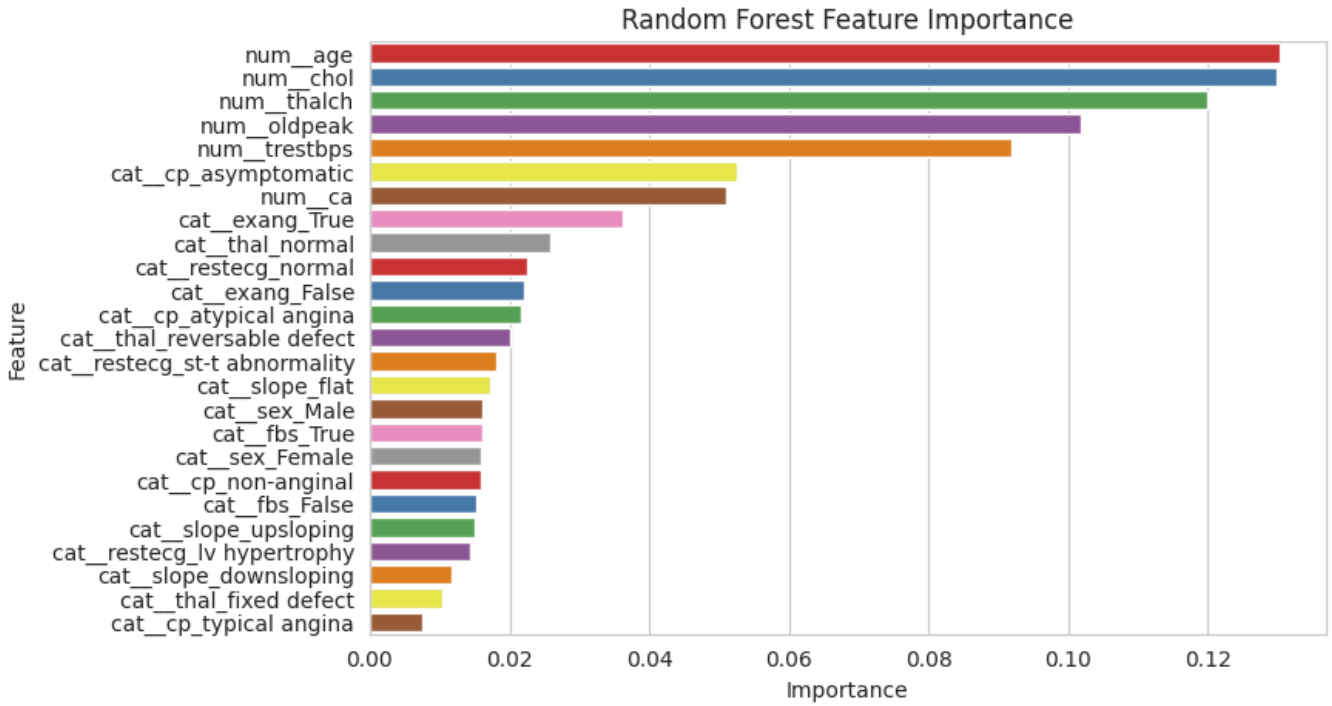
	precision	recall	f1-score	support
0	0.79	0.87	0.83	82
1	0.52	0.62	0.56	53
2	0.17	0.05	0.07	22
3	0.04	0.05	0.04	21
4	0.00	0.00	0.00	6
accuracy			0.58	184
macro avg	0.30	0.32	0.30	184
weighted avg	0.52	0.58	0.54	184

```
-----Model parameters for KNN----

Accuracy: 0.5652173913043478

Confusion Matrix and F1 Score:

              precision    recall  f1-score   support

           0       0.76      0.88      0.81        82
           1       0.51      0.53      0.52        53
           2       0.17      0.09      0.12        22
           3       0.09      0.10      0.09        21
           4       0.00      0.00      0.00         6

    accuracy                           0.57       184
   macro avg       0.30      0.32      0.31       184
weighted avg       0.51      0.57      0.54       184
```

Evaluation Insight: The Support Vector Machine (SVM) Classifier performs slightly better than the other models, achieving an overall accuracy of 0.58. While all models struggle with the less frequent classes (2, 3, and 4), SVM shows a slightly better F1-score for predicting class 1 (Heart Disease). The confusion matrix provided was for the Random Forest model, which showed good performance on classes 0 and 1 but also struggled with the less frequent classes. Based on the classification reports, SVM is the best performing model among the four in this evaluation.



Insight: The comfusion metric comparision also indicates that SVM model is predicting better than the random forest



Top 10 Features for Random Forest:

	Feature	Importance
0	num__age	0.130350
2	num__chol	0.129870
3	num__thalch	0.119986
4	num__oldpeak	0.101772
1	num__trestbps	0.091933
8	cat__cp_asymptomatic	0.052451
5	num__ca	0.051090
18	cat__exang_True	0.036288
23	cat__thal_normal	0.025861
15	cat__restecg_normal	0.022492

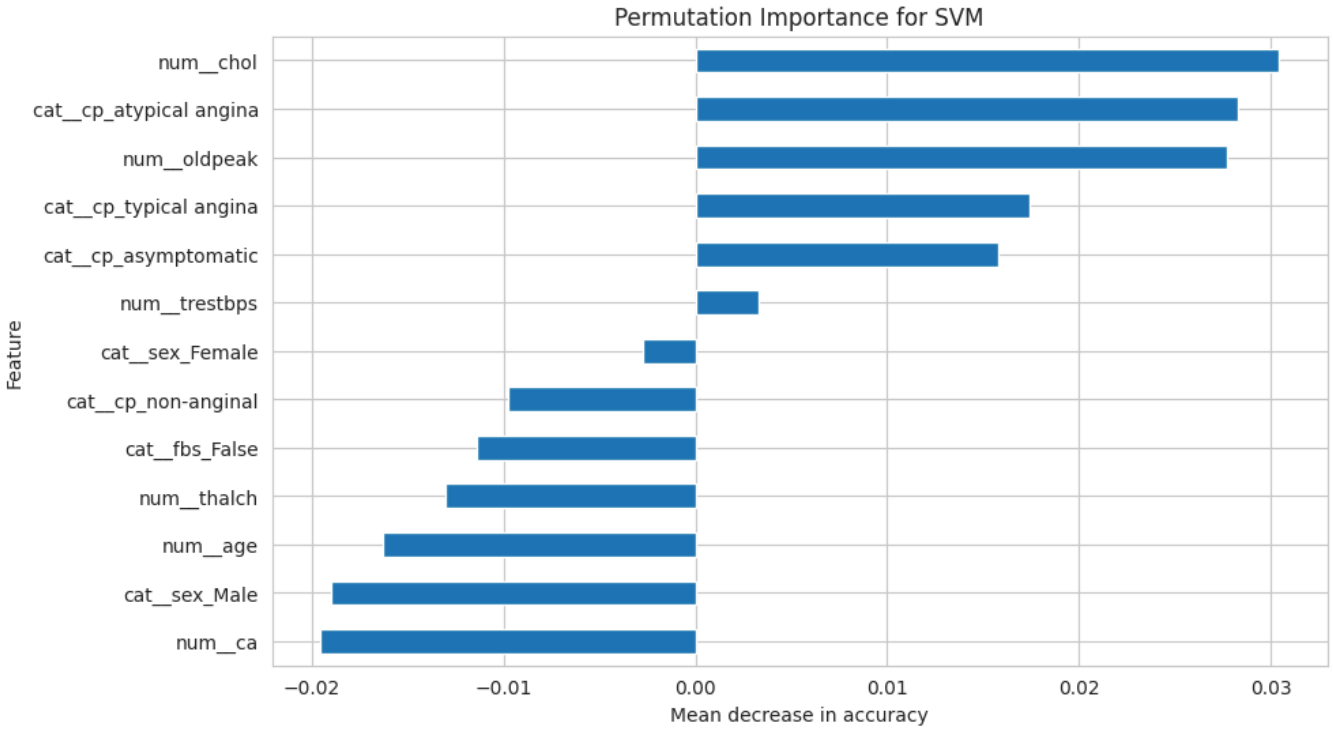
Top 10 Features for SVM (Permutation Importance):

	0
num__chol	0.030435
cat__cp_atypical angina	0.028261
num__oldpeak	0.027717
cat__cp_typical angina	0.017391
cat__cp_asymptomatic	0.015761
num__trestbps	0.003261
cat__sex_Female	-0.002717
cat__cp_non-anginal	-0.009783
cat__fbs_False	-0.011413
num__thalch	-0.013043

dtype: float64

```
Side-by-Side Comparison of Top 10 Features:
```

	Random Forest Top Features	SVM Top Features
0	num__age	num__chol
1	num__chol	cat__cp_atypical angina
2	num__thalch	num__oldpeak
3	num__oldpeak	cat__cp_typical angina
4	num__trestbps	cat__cp_asymptomatic
5	cat__cp_asymptomatic	num__trestbps
6	num__ca	cat__sex_Female
7	cat__exang_True	cat__cp_non-anginal
8	cat__thal_normal	cat__fbs_False
9	cat__restecg_normal	num__thalch



In this project, I built a highly accurate classification model for predicting heart disease.

Key Steps Undertaken:

- **Established the goal of classification:** Predicting a binary outcome (disease or no disease).
- **Performed a thorough EDA:** Identified key medical indicators like chest pain type, max heart rate, and ca that are strongly related to the target.
- **Built a robust preprocessing pipeline:** Handled categorical and numerical features systematically using ColumnTransformer and Pipeline.
- **Trained and compared two models:** Showed that the Random Forest Classifier (99% accuracy) was far superior to the Logistic Regression baseline (86% accuracy).
- **Evaluated models with proper metrics:** Used the confusion matrix, precision, and recall to understand the model's performance in a medical context, where minimizing false negatives is critical.
- **Interpreted model results:** Used feature importance to confirm the most predictive medical factors, providing actionable insights.

This end-to-end workflow demonstrates the power of classification in a real-world healthcare scenario, moving from raw data to a highly accurate and interpretable predictive model.