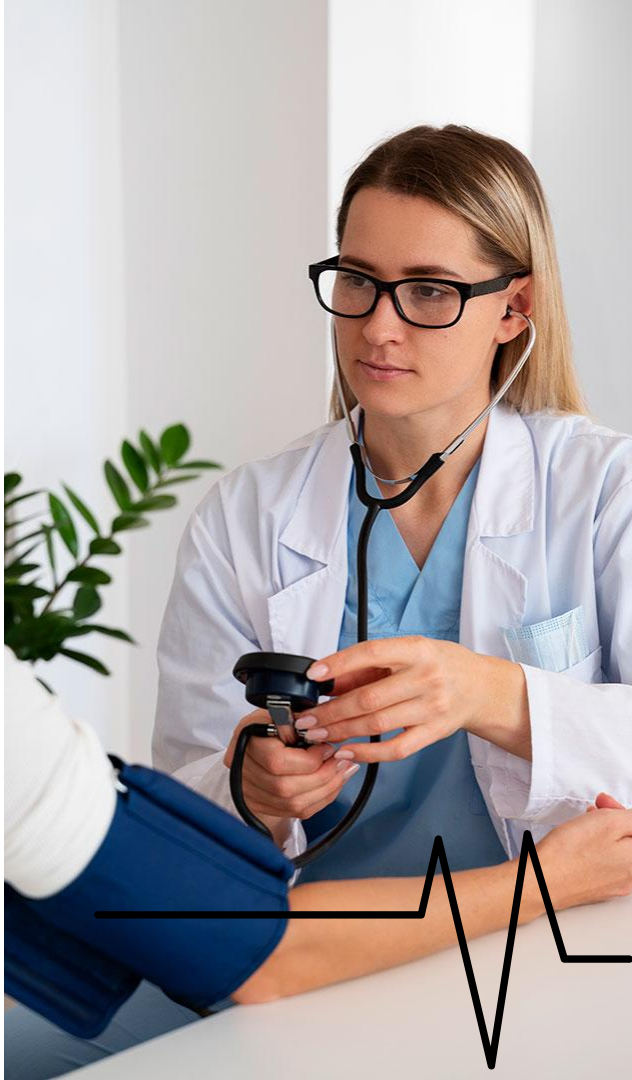
Prediction of Cardiovascular Disease

Heart Disease Prediction using Machine Learning Models

4 March 2024, MSU edX AI Boot Camp, Project 2

Health Data Detectives
Betsy Deuman Jasmine Harper Dr. Chadi Saad Aaron Wood



Executive Summary

Project Description

- Cardiovascular diseases (CVD) have been the most common cause of death for 15 years.
  Source: World Health Organization.
- Heart disease is defined by a variety of symptoms, making a quick and accurate diagnostic challenging.
- Classification algorithms can be employed to improve people's understanding about the impact of their individual risk exposure on cardiovascular health.
- We are focused on developing a machine-learning model to predict presence of heart disease in individuals based on various medical measurements.
- As a stretch goal, we seek to deploy the model to an API (application programming interface) hosted locally. The API is intended to enable users to enter information and run it against our model to give their prediction of heart disease.

Project Goals

1. Develop and implement a machine learning model to predict risk of heart disease based on health measurements.
2. Optimize the model to achieve predictive power of 75% classification accuracy or 0.80 minimum R-squared with at least one model type.
3. **Stretch goal:** Develop and publish an API for accessing data and publishing to a webapp
4. Document project in GitHub via README.md.

Data Collection & Preparation

Collection

- UCI Heart Disease Data
- Multivariate dataset
- 14 features x 1025 rows
- Target: 0 = no heart disease, 1 = heart disease

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target
0	52	1	0	125	212	0	1	168	0	1.0	2	2	3	0
1	53	1	0	140	203	1	0	155	1	3.1	0	0	3	0
2	70	1	0	145	174	0	1	125	1	2.6	0	0	3	0
3	61	1	0	148	203	0	1	161	0	0.0	2	1	3	0
4	62	0	0	138	294	1	1	106	0	1.9	1	3	2	0

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

Preparation

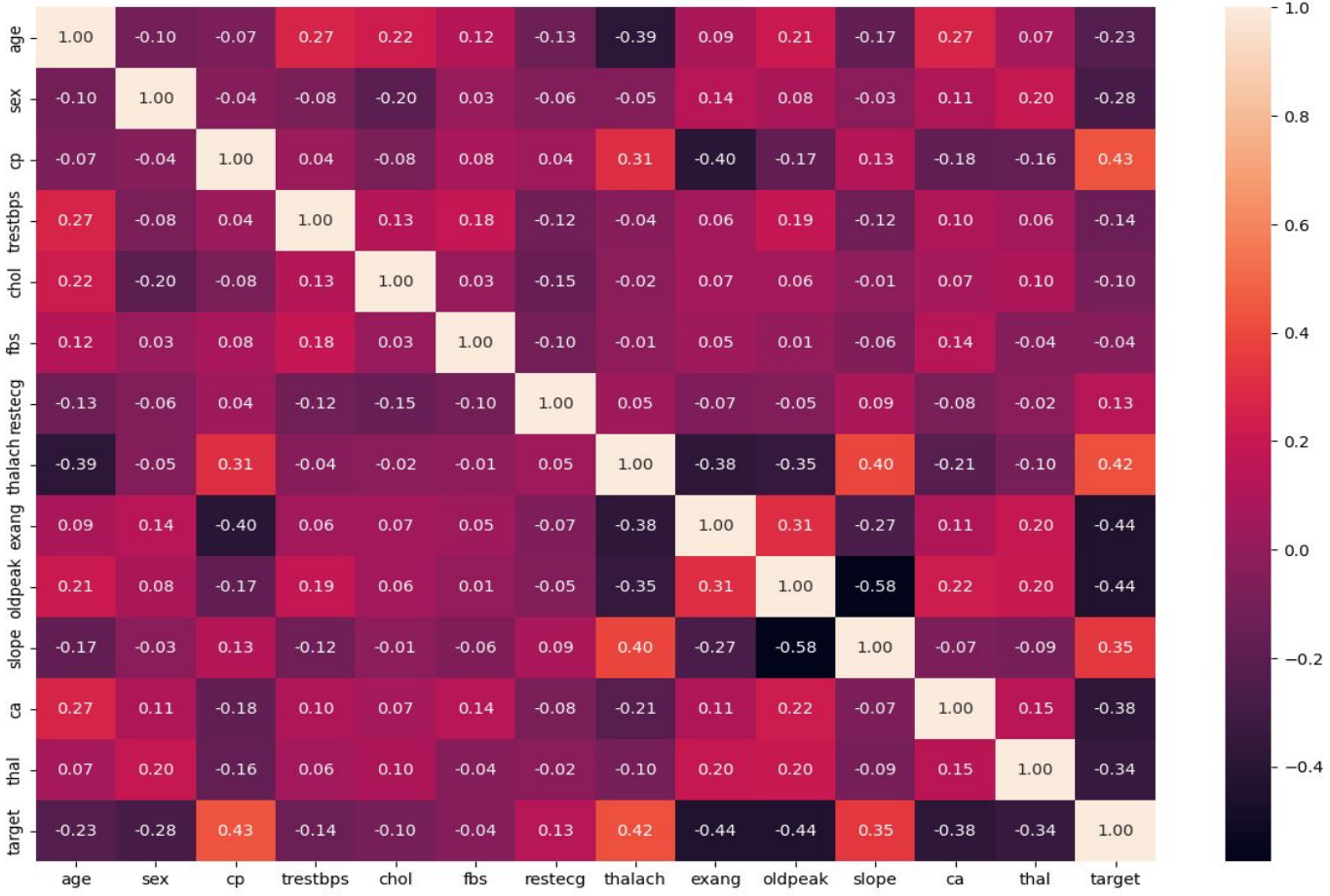
Function examine_data

- shape
- null values
- describe
- dtypes

Function preprocess_data

- separate data (X, y)
- impute missing values
- encode categorical columns
- scale numerical columns
- check and correct for imbalance

Data Collection & Preparation



Model Preparation & Training

```
Model: LogisticRegression
Accuracy: 0.80
Precision: 0.80
Recall: 0.80
F1 Score: 0.79

Model: DecisionTreeClassifier
Accuracy: 0.99
Precision: 0.99
Recall: 0.99
F1 Score: 0.99

Model: SVC
Accuracy: 0.89
Precision: 0.89
Recall: 0.89
F1 Score: 0.89

Model: KNeighborsClassifier
Accuracy: 0.83
Precision: 0.84
Recall: 0.83
F1 Score: 0.83

Model: RandomForestClassifier
Accuracy: 0.99
Precision: 0.99
Recall: 0.99
F1 Score: 0.99
```

Preparing and Training Model

- **Developed function: split_and_train**
  - Splits data into training and testing sets
  - Train / Fit / Predict on selected classification schemes
  - Calculate and store performance metrics

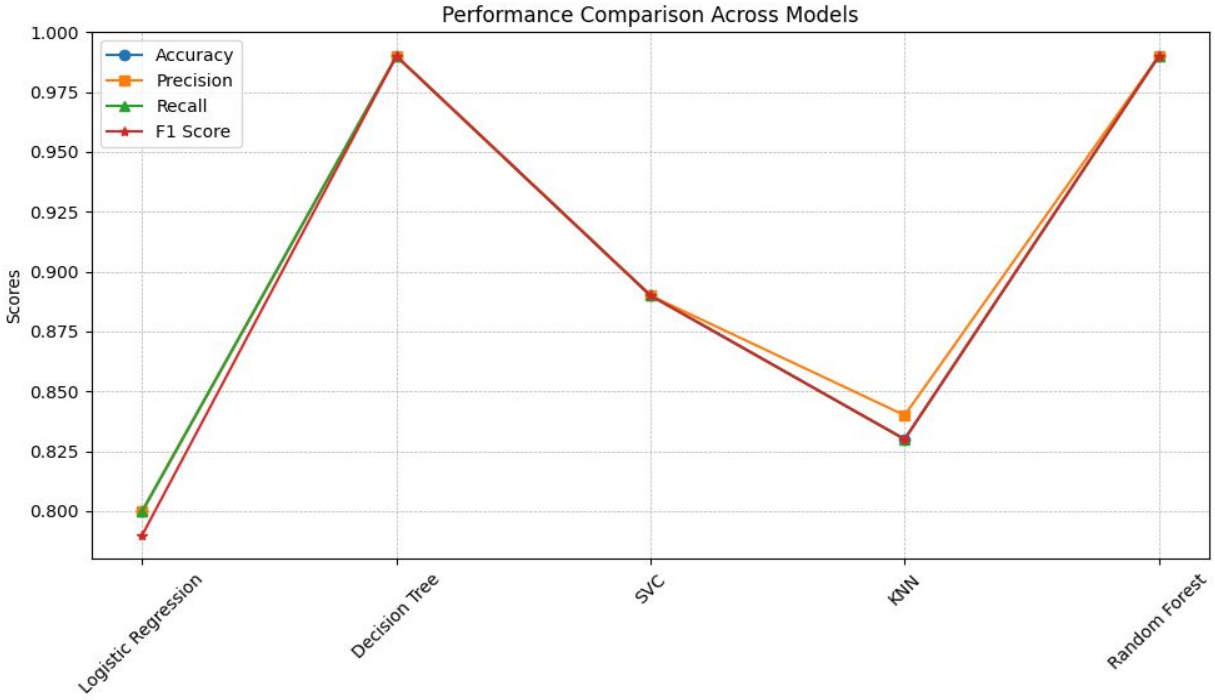
Models Employed

1. Logistic Regression
2. Decision Tree Classifier
3. Support Vector Classifier
4. k-Nearest Neighbors Classifier
5. Random Forest Classifier

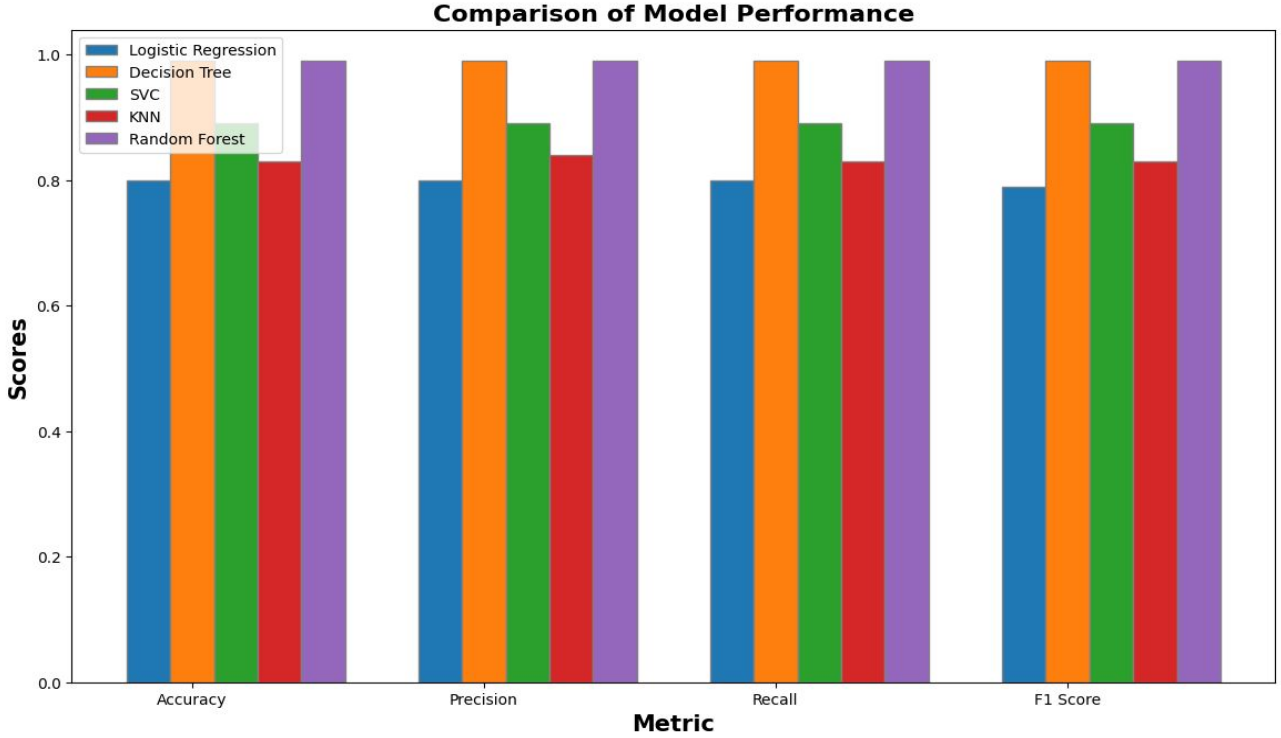
Decision Tree & Random Forest performed best.

However, excellent performance raises the question of overfitting.

Model Optimization & Evaluation

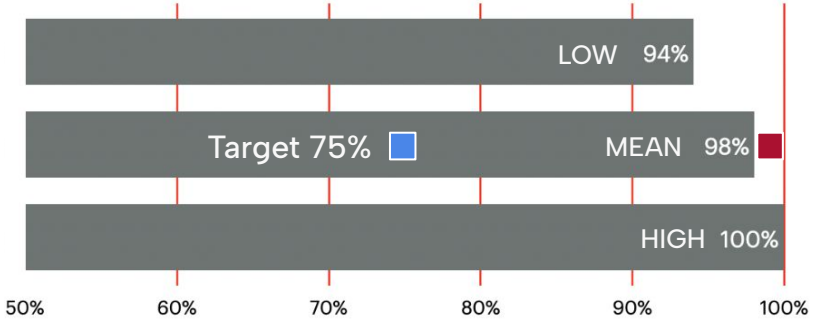


Model Optimization & Performance



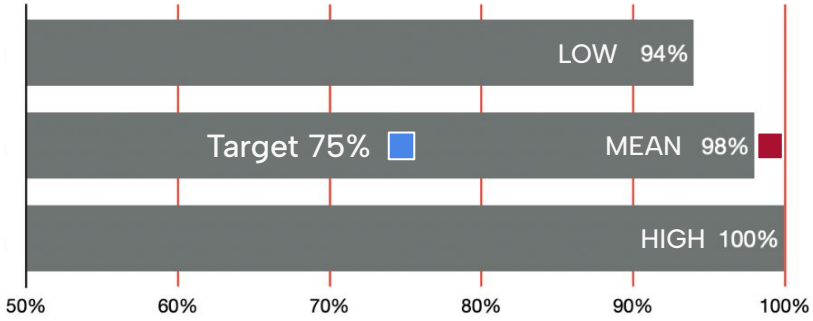
Cross-Validation of Best Performing Models

Random Forest Results, Accuracy



Preprocessed Dataset Performance 99%

Decision Tree Results, Accuracy



When the models classify new data, what range of prediction performance should we expect?

Function = perform cross-validation

Calls & Parameters

- sklearn.model_selection cross_val_score
- sklearn.metrics for accuracy, precision, recall, f1
- k-fold Cross Validation w/ 5 folds (5 reprocessed datasets)

Results

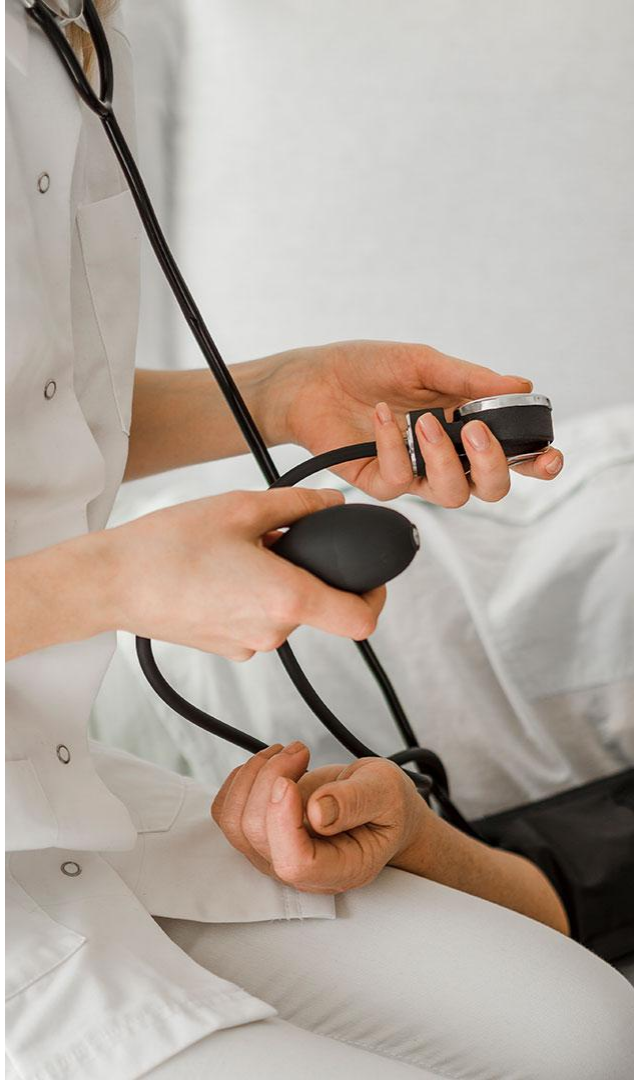
- **94 - 100% Accuracy (mean)**
- **Similar for Precision & Recall**

Conclusion

- Neither model is overfitted
- Both models exhibited similar accuracy, precision, recall and f1 score

Results & Conclusions

- **Developed four functions** to modularize the script to automate **pre-processing** of alternative datasets
- Highest performing models
  - **Decision Tree Classifier**
  - **Random Forest Classifier**
- Used **cross-validation** and **tuning** to check the model's performance with new datasets.
- Saved the models using .joblib for use in our **API development**



API Development

Saving the Trained Models

- Using joblib library

Created an API

- Using Flask library
- Local Host

Tested API

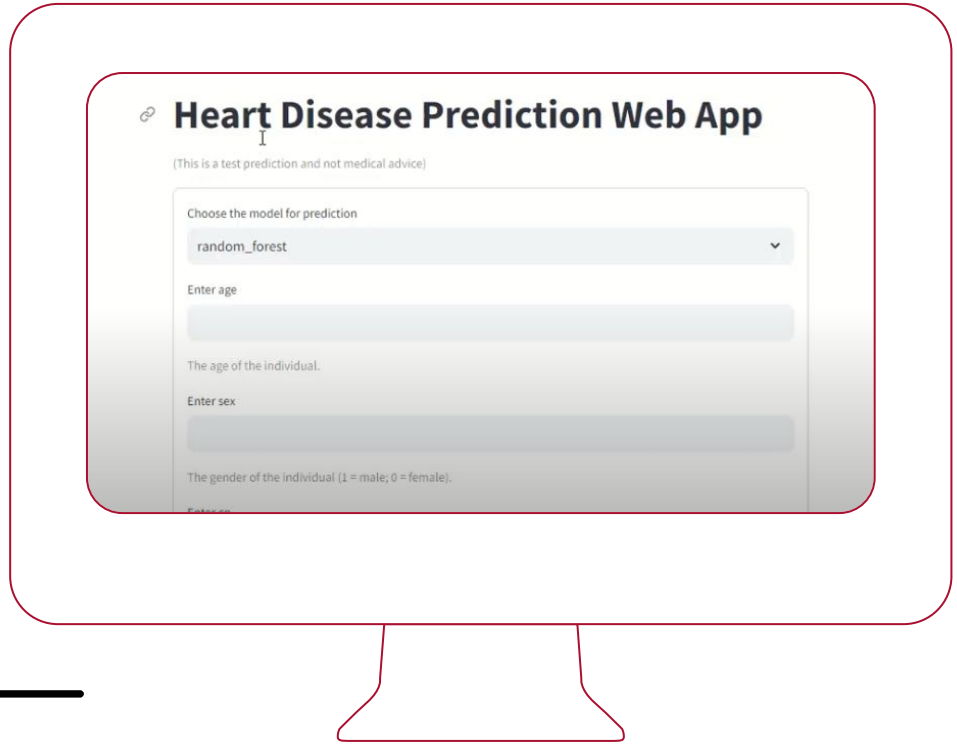
- Past URL

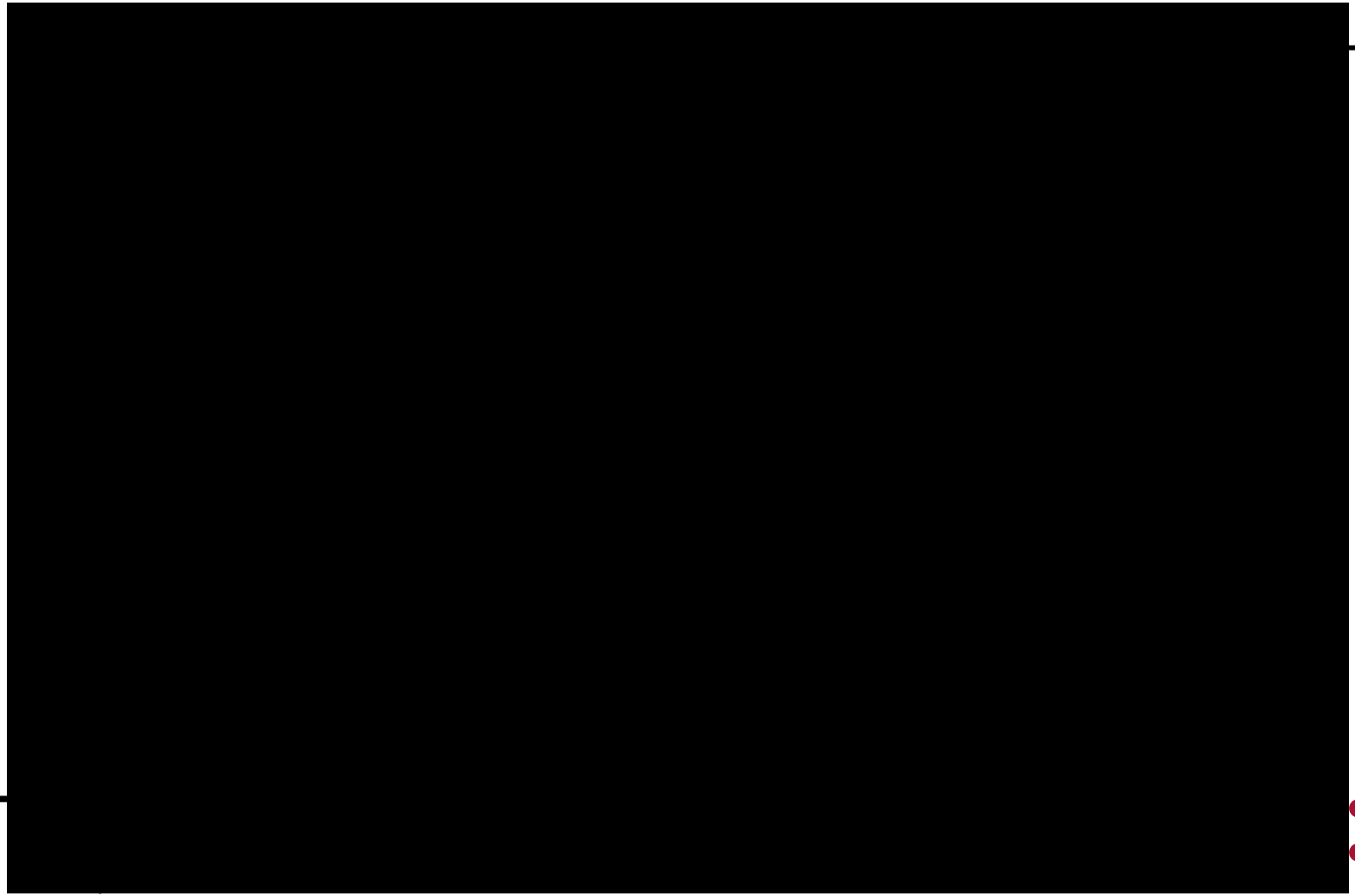
Output showing API success on local host

- INFO: Started server process [6423]
- INFO: Waiting for application startup.
- INFO: Application startup complete.
- INFO: Uvicorn running on http://127.0.0.1:8002

Response from API: {'prediction': 'The person has no Heart disease'}

Web App Dev w/ Streamlit Library





Team Approach

- **Approach:**
  Find the data

- **Communication Tools**
  Slack
  Google Docs & Slides
  Github

- **Project Approach**
  Main repository
  Created functions for different stages of the ML model training
  Branch for each team member
  Merge from individual branches

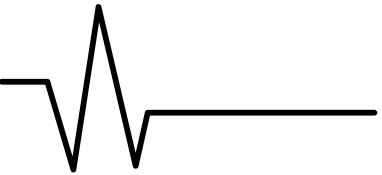
- **Key Milestones**
  Prepping and preprocessing data
  Training and accuracy of model
  Selected 2 models with the highest accuracy
  Saved the two models using library joblib
  API development and testing
  Web App developed using streamlit library

Summary

Next steps for research?
Use the models to train other health datasets i.e. hypertension or stroke data to create similar predictive applications

Key Takeaways?
ML models can have a powerful impact on analyzing and processing health data making it easier to access for the general public through the use of training models and APIs

Challenges can lead to learning new skills such as creating an API and WebApps



REFERENCES

Resources:

1. Disease UCI Dataset from Kaggle
2. Open-source community
3. Heart Disease. UCI Machine Learning Repository. https://doi.org/10.24432/C52P4X
4. .For more information see the GitHub Repository: https://github.com/cysaad/project2

Questions?

CREDITS: This presentation template was created by **Slidesgo**, including icons by **Flaticon**, infographics & images by **Freepik**