Internship Project 1: Predicting Mortality of Heart Failure Patients

Abstract

This project aims to analyze and predict the mortality of heart failure patients using a dataset from Kaggle. The dataset includes clinical features such as age, blood pressure, ejection fraction, and other vital health indicators. The project demonstrates the use of Exploratory Data Analysis (EDA) and machine learning models to predict patient outcomes. The results and visualizations help identify the most critical risk factors for mortality.

Objective

- To perform exploratory data analysis on the heart failure dataset.
- To identify important features contributing to patient mortality.
- To implement predictive models for mortality risk.
- To evaluate and interpret model performance.

Introduction

Heart failure is a chronic condition where the heart is unable to pump blood effectively, leading to severe health complications. Understanding the clinical risk factors and predicting mortality risk can help in providing timely treatment. With the availability of medical datasets on Kaggle, data-driven models can be built to analyze trends, predict mortality, and provide actionable insights for healthcare providers.

Methodology

1. Data Collection: Dataset sourced from Kaggle containing clinical features of heart failure patients.
2. Data Cleaning: Handle missing values, normalize numeric features, and encode categorical features.
3. Exploratory Data Analysis (EDA): Visualize data distribution, correlations, and mortality patterns.
4. Feature Selection: Identify the most relevant clinical features affecting mortality.
5. Model Building: Implement classification models such as Logistic Regression, Random Forest, and SVM.
6. Model Evaluation: Evaluate accuracy, precision, recall, F1-score, and ROC-AUC.
7. Insights: Summarize patterns and critical risk indicators for heart failure mortality.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	age	sex	ejection_fi	serum_cre	serum_soc	high_blooc	anaemia	diabetes	smoking	platelets	creatinine_	time	death	
2	78	0	19	0.62	135	0	0	0	1	493633	5513	50	1	
3	68	0	62	3.14	134	1	0	1	0	407309	633	254	0	
4	54	1	33	4.73	142	0	0	1	1	267107	444	37	1	
5	82	1	18	3.09	134	0	0	1	0	268543	5233	129	1	
6	47	1	49	2.25	124	0	0	1	0	395033	3838	269	0	
7	60	1	63	3.39	140	0	0	0	0	195511	2818	116	1	
8	78	0	31	2.56	133	1	1	1	1	144190	4217	43	1	
9	58	1	58	2.96	121	0	1	1	0	241849	3910	31	0	
10	62	0	42	4.74	130	1	1	1	1	214896	323	70	1	
11	50	1	44	2.24	138	1	1	0	1	218523	2063	140	1	
12	50	0	43	4.83	126	0	1	0	0	193443	5489	43	1	
13	63	1	60	4.57	125	1	0	1	0	116837	4483	260	1	
14	75	1	67	1.38	147	1	0	0	0	107150	1439	113	1	
15	79	1	20	0.81	121	1	1	1	1	495089	3616	159	1	
16	63	1	49	0.95	125	0	0	0	0	271109	4752	260	1	
17	42	0	55	0.58	137	1	1	0	1	253731	215	174	0	
18	61	1	51	0.92	121	1	1	0	1	371859	1617	57	1	
19	41	0	38	3.57	137	1	0	0	0	187302	6204	95	0	
20	63	0	43	0.82	149	0	0	1	0	479984	5470	192	1	
21	83	0	63	1.94	134	1	0	1	1	414538	778	258	0	
22	69	0	60	4.3	146	0	1	1	1	135764	6994	290	1	
23	77	1	67	0.6	138	0	0	0	1	267032	2093	193	0	
24	41	0	45	4.17	121	0	1	1	0	451647	4247	39	0	
25	60	0	49	1.77	139	0	1	1	1	477893	3546	224	0	
26	72	0	47	1.03	144	0	1	1	0	286961	5561	147	1	

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
27	51	1	66	3.64	125	0	0	0	1	345365	828	299	0	
28	61	1	35	3.33	120	1	0	0	1	166814	263	109	1	
29	83	1	46	4.45	143	0	1	0	0	496467	3812	122	0	
30	64	1	37	3.81	134	1	0	1	1	192669	5257	295	1	
31	88	0	47	4.12	147	1	0	1	0	477093	1626	286	1	
32	66	0	17	1.77	149	0	0	1	0	359859	2498	162	1	
33	81	1	32	1.3	142	1	1	0	1	343095	6781	293	1	
34	67	0	39	3.88	129	1	1	0	0	305075	2986	200	1	
35	55	0	56	4.13	138	1	0	1	0	192268	3167	30	1	
36	54	0	45	4.96	136	1	1	0	1	170611	5841	93	1	
37	86	1	68	2.36	124	0	0	0	0	188194	7680	150	0	
38	83	1	17	2.17	148	1	0	0	0	174575	641	136	1	
39	42	0	54	3.99	123	0	0	0	0	411834	5479	204	0	
40	76	1	60	2.03	129	1	0	1	1	240050	5255	233	0	
41	46	1	38	4.69	143	1	1	0	0	123137	6285	194	1	
42	60	1	64	4.36	136	0	1	0	1	487641	7877	149	0	
43	48	1	46	2.43	129	1	1	0	1	453514	2770	73	1	
44	78	1	61	3.88	136	1	0	0	0	471101	763	185	1	
45	57	0	36	3.9	147	1	0	1	0	497963	985	236	1	
46	43	1	37	0.96	139	0	0	1	0	169558	6097	137	1	
47	64	0	16	4.56	143	1	1	0	1	258497	4387	280	1	
48	53	0	41	2.77	124	0	0	0	0	403295	6189	159	1	
49	89	1	56	4.22	121	1	1	0	0	378408	51	49	1	
50	48	0	16	1.94	125	0	1	0	0	161558	6418	30	1	
51	65	0	40	4.53	140	0	1	0	0	426333	2898	148	1	
52	41	0	31	2.25	121	1	1	0	0	189776	5914	64	1	

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
53	59	1	54	0.55	132	1	1	1	1	189527	4094	227	1	
54	67	1	47	4.57	149	0	1	0	1	314790	1999	274	1	
55	86	0	23	0.91	130	0	0	1	0	337176	6782	88	1	
56	46	1	57	1.94	130	1	0	1	0	332034	6309	190	0	
57	83	1	68	4.78	135	0	1	0	0	136595	7883	208	1	
58	47	1	62	4.78	130	1	1	1	0	450984	967	235	1	
59	86	1	53	3.08	138	1	0	1	1	206240	3224	41	1	
60	74	0	43	3.34	134	1	1	1	0	151806	2291	210	1	
61	53	0	56	2.52	144	1	1	1	1	455499	7383	271	0	
62	56	1	69	1.82	142	0	0	1	1	482261	7064	245	0	
63	75	1	40	1.98	142	1	1	1	0	444851	139	280	0	
64	89	0	49	3.53	135	0	1	0	1	423806	1759	155	0	
65	79	0	64	3.89	130	1	1	1	1	362097	2219	246	1	
66	43	1	39	4.06	141	0	0	1	1	320343	4288	214	1	
67	41	0	38	4.05	135	1	0	1	1	134795	3091	68	1	
68	45	1	27	0.91	127	0	0	0	0	263381	155	252	0	
69	81	1	21	2.72	123	0	1	1	0	249075	936	213	0	
70	43	0	50	0.76	127	1	1	0	1	203902	477	207	0	
71	68	0	59	2.97	123	1	0	1	0	389368	5474	180	0	
72	57	1	34	2.49	143	1	1	0	0	298350	1945	155	1	
73	65	1	15	4.49	144	1	1	0	0	132418	6140	263	1	
74	83	0	22	2.08	122	0	1	1	0	188073	366	37	1	
75	73	1	60	1.03	122	1	1	1	0	373304	6132	158	0	
76	49	0	30	1.14	146	0	1	0	1	130452	684	293	1	
77	75	1	28	3.93	148	0	0	1	0	440483	1975	186	1	
78	53	0	26	3.28	137	1	1	0	0	298059	2106	34	0	
79	70	0	65	0.96	148	0	0	0	0	292235	6967	115	1	
80	87	0	37	0.88	145	0	1	0	0	336963	3194	204	1	
81	54	1	29	3.65	138	0	0	0	1	429872	7576	222	1	
82	47	1	42	0.83	138	0	1	0	1	239124	2320	148	0	
83	53	0	48	4.2	140	1	0	1	1	371206	1489	247	1	
84	62	1	16	3.68	124	1	1	0	1	326293	602	143	1	
85	79	0	46	0.87	137	1	0	0	0	206811	7587	174	0	
86	60	0	37	0.88	147	1	1	1	1	451452	765	55	1	
87	55	1	36	4.94	129	0	0	0	0	418970	5433	165	1	
88	84	1	65	2.18	148	0	1	1	0	363381	3961	261	1	
89	57	0	39	2.17	141	1	0	1	0	440233	1763	105	1	
90	86	1	36	4.16	140	1	0	1	0	446918	6029	30	1	
91	63	1	36	4.76	125	1	1	0	0	383345	3439	176	1	
92	65	0	63	4.94	120	0	1	0	1	434805	1157	171	0	
93	64	0	66	3.89	124	1	0	0	1	378989	5487	82	1	
94	84	1	56	2.19	128	1	1	1	1	372056	4256	98	0	
95	80	0	20	0.88	131	1	0	0	1	347445	300	119	0	
96	68	0	29	4	145	0	0	1	0	401087	2480	249	1	
97	54	1	68	3.01	133	0	0	0	1	163442	2104	221	1	
98	84	0	57	2.41	121	0	1	1	0	452348	508	92	1	
99	40	0	51	4.58	136	0	0	0	1	448737	5253	102	1	
100	64	1	47	1	133	0	0	0	0	111699	257	190	0	
101	46	1	22	2.72	145	0	1	0	0	430327	949	198	0	
102	48	0	67	0.55	132	0	1	1	1	151548	3511	118	1	
103	63	0	58	2.61	146	0	1	1	0	234048	3435	205	1	
104	40	1	58	0.75	128	1	1	0	0	397403	71	298	1	

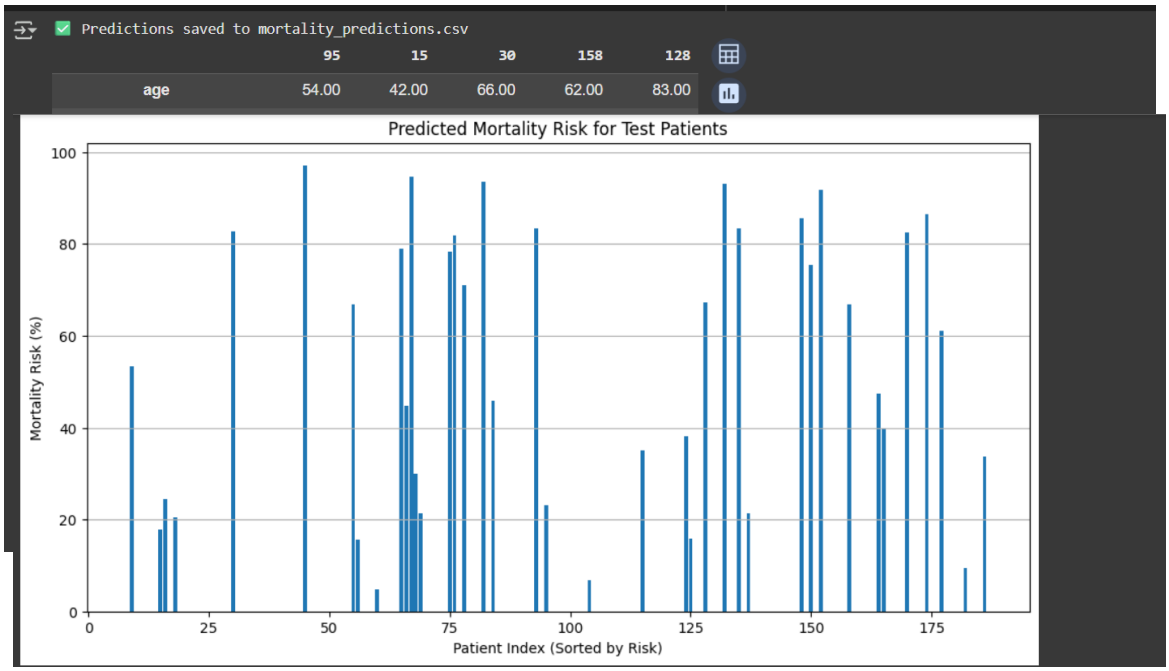
Code

- **Input**

```
pipe.fit(X_train, y_train)

# ==== 5. Predict mortality probability in % ====
y_pred_proba = pipe.predict_proba(X_test)[:, 1] * 100

# ==== 6. Save predictions ====
pred_df = X_test.copy()
pred_df["true_outcome"] = y_test.values
pred_df["mortality_risk_%"] = np.round(y_pred_proba, 2)

pred_df.to_csv("mortality_predictions.csv", index=False)
print("✅ Predictions saved to mortality_predictions.csv")
display(pred_df.head().T)

# ==== 7. Visualize predictions with a bar graph ====
# Sort the predictions by mortality risk for better visualization
pred_df_sorted = pred_df.sort_values(by="mortality_risk_%")

plt.figure(figsize=(12, 6))
plt.bar(pred_df_sorted.index, pred_df_sorted["mortality_risk_%"]) # Changed to bar plot
plt.xlabel("Patient Index (Sorted by Risk)")
plt.ylabel("Mortality Risk (%)")
plt.title("Predicted Mortality Risk for Test Patients")
plt.grid(axis='y') # Add grid lines to y-axis for bar graph
plt.show()

import os
print(os.listdir('/tmp'))
```

```
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.linear_model import LogisticRegression
from sklearn.pipeline import Pipeline
from sklearn.impute import SimpleImputer # Corrected import
import matplotlib.pyplot as plt # Import matplotlib

# ==== 1. Load dataset ====
df = pd.read_csv("/content/heart_failure_200.csv")

# ==== 2. Features and target ====
target = "death"
X = df.drop(columns=[target])
y = df[target]

# ==== 3. Train-test split ====
X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.2, random_state=42
)

# ==== 4. Pipeline: Imputation + Scaling + Logistic Regression ====
pipe = Pipeline([
    ("imputer", SimpleImputer(strategy="median")),
    ("scaler", StandardScaler()),
    ("clf", LogisticRegression(max_iter=1000, class_weight="balanced"))
])
```

- **Output**



Conclusion

The project successfully analyzed the heart failure dataset and identified significant clinical factors associated with mortality. Machine learning models were implemented to predict the mortality risk of patients. The findings highlight the importance of clinical indicators such as age, ejection fraction, and serum creatinine in predicting outcomes. This work demonstrates how data-driven healthcare analytics can support clinical decision-making and improve patient care.