

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

```
import pandas as pd
from sklearn.model_selection import GridSearchCV
from sklearn.preprocessing import StandardScaler
from sklearn.impute import SimpleImputer
from sklearn.pipeline import Pipeline
from sklearn.feature_selection import SelectKBest, f_classif
from sklearn.linear_model import LogisticRegression
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score, classification_report
```

```
data = pd.read_csv("heart.csv")

y = data["target"]
X = data.drop("target", axis=1)
```

```
pipeline = Pipeline([
    ("imputer", SimpleImputer(strategy="median")),
    ("scaler", StandardScaler())
])

X_processed = pipeline.fit_transform(X)
```

```
selector = SelectKBest(score_func=f_classif, k=8)
X_selected = selector.fit_transform(X_processed, y)
```

```
models = {
    "LogisticRegression": LogisticRegression(max_iter=500),
    "RandomForest": RandomForestClassifier()
}

params = {
    "LogisticRegression": {"C": [0.1, 1, 10]},
    "RandomForest": {"n_estimators": [100, 200], "max_depth": [5, 10]}
}

for name in models:
    grid = GridSearchCV(models[name], params[name], cv=5)
    grid.fit(X_selected, y)
    pred = grid.predict(X_selected)
    print(name)
    print(accuracy_score(y, pred))
    print(classification_report(y, pred))
```

```
LogisticRegression
0.8468292682926829
      precision    recall  f1-score   support

     0       0.89      0.79      0.83       499
     1       0.82      0.90      0.86       526

 accuracy          0.85          0.85          0.85       1025
 macro avg          0.85          0.85          0.85       1025
weighted avg          0.85          0.85          0.85       1025

RandomForest
0.9960975609756098
      precision    recall  f1-score   support

     0       1.00      0.99      1.00       499
     1       0.99      1.00      1.00       526

 accuracy          1.00          1.00          1.00       1025
 macro avg          1.00          1.00          1.00       1025
weighted avg          1.00          1.00          1.00       1025
```

```
final_model = GridSearchCV(RandomForestClassifier(), params["RandomForest"], cv=5)
final_model.fit(X_selected, y)

predictions = final_model.predict(X_selected)
```