

dataset: <https://www.kaggle.com/datasets/uciml/pima-indians-diabetes-database?resource=download>

```
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("diabetes.csv")

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

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

X_processed = pipeline.fit_transform(X)
```

```
selector = SelectKBest(score_func=f_classif, k=5)
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  
0.7760416666666666
```

|              | precision | recall | f1-score | support |
|--------------|-----------|--------|----------|---------|
| 0            | 0.79      | 0.89   | 0.84     | 500     |
| 1            | 0.73      | 0.56   | 0.64     | 268     |
| accuracy     |           |        | 0.78     | 768     |
| macro avg    | 0.76      | 0.73   | 0.74     | 768     |
| weighted avg | 0.77      | 0.78   | 0.77     | 768     |

```
RandomForest  
0.8294270833333334
```

|              | precision | recall | f1-score | support |
|--------------|-----------|--------|----------|---------|
| 0            | 0.84      | 0.91   | 0.87     | 500     |
| 1            | 0.80      | 0.68   | 0.74     | 268     |
| accuracy     |           |        | 0.83     | 768     |
| macro avg    | 0.82      | 0.80   | 0.81     | 768     |
| weighted avg | 0.83      | 0.83   | 0.83     | 768     |

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

```
final_predictions = final_model.predict(X_selected)
```