


```
from google.colab import files
```

```
uploaded = files.upload()
```

 heart disease.csv
 • heart disease.csv(text/csv) - 36840 bytes, last modified: 5/10/2025 - 100% done
 Saving heart disease.csv to heart disease.csv

```
import pandas as pd
```

```
df = pd.read_csv('heart disease.csv')
```

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from sklearn import datasets, model_selection, linear_model, neural_network
from mlxtend.plotting import plot_decision_regions
```


```
# Load the dataset
data = datasets.load_breast_cancer()
X = data.data
y = data.target
```

```
# Split the dataset into train and test sets
X_train, X_test, y_train, y_test = model_selection.train_test_split(
    X, y, test_size=0.2, random_state=42)
```


```
# Logistic Regression
lr = linear_model.LogisticRegression(max_iter=10000)
lr.fit(X_train, y_train)
y_pred_lr = lr.predict(X_test)
```

```
# Neural Network
mlp = neural_network.MLPClassifier(hidden_layer_sizes=(50, 50), max_iter=1000)
mlp.fit(X_train, y_train)
y_pred_nn = mlp.predict(X_test)
```

```
# Compare accuracy
print(f"Logistic Regression: {lr.score(X_test, y_test)}")
print(f"Neural Network: {mlp.score(X_test, y_test)}")
```

 Logistic Regression: 0.956140350877193
 Neural Network: 0.9298245614035088

```
!pip uninstall scikit-learn
```

 Found existing installation: scikit-learn 1.6.1
 Uninstalling scikit-learn-1.6.1:
 Would remove:
 /usr/local/lib/python3.11/dist-packages/scikit_learn-1.6.1.dist-info/*
 /usr/local/lib/python3.11/dist-packages/scikit_learn.libs/libgomp-a34b3233.so.1.0.0
 /usr/local/lib/python3.11/dist-packages/sklearn/*
 Proceed (Y/n)? y
 Successfully uninstalled scikit-learn-1.6.1

```
!pip install matplotlib-venn
```

 Requirement already satisfied: matplotlib-venn in /usr/local/lib/python3.11/dist-packages (1.1.2)
 Requirement already satisfied: matplotlib in /usr/local/lib/python3.11/dist-packages (from matplotlib-venn) (3.10.0)
 Requirement already satisfied: numpy in /usr/local/lib/python3.11/dist-packages (from matplotlib-venn) (2.0.2)
 Requirement already satisfied: scipy in /usr/local/lib/python3.11/dist-packages (from matplotlib-venn) (1.15.3)
 Requirement already satisfied: contourpy>=1.0.1 in /usr/local/lib/python3.11/dist-packages (from matplotlib->matplotlib-venn) (1.3.2)
 Requirement already satisfied: cycler>=0.10 in /usr/local/lib/python3.11/dist-packages (from matplotlib->matplotlib-venn) (0.12.1)
 Requirement already satisfied: fonttools>=4.22.0 in /usr/local/lib/python3.11/dist-packages (from matplotlib->matplotlib-venn) (4.58.0)
 Requirement already satisfied: kiwisolver>=1.3.1 in /usr/local/lib/python3.11/dist-packages (from matplotlib->matplotlib-venn) (1.4.8)
 Requirement already satisfied: packaging>=20.0 in /usr/local/lib/python3.11/dist-packages (from matplotlib->matplotlib-venn) (24.2)
 Requirement already satisfied: pillow>=8 in /usr/local/lib/python3.11/dist-packages (from matplotlib->matplotlib-venn) (11.2.1)
 Requirement already satisfied: pyparsing>=2.3.1 in /usr/local/lib/python3.11/dist-packages (from matplotlib->matplotlib-venn) (3.2.3)
 Requirement already satisfied: python-dateutil>=2.7 in /usr/local/lib/python3.11/dist-packages (from matplotlib->matplotlib-venn) (2.9.0)
 Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.11/dist-packages (from python-dateutil>=2.7->matplotlib->matplotlib-ve

```
import matplotlib_venn as vn
```

```
# Compare accuracy
print(f"Logistic Regression: {lr.score(X_test, y_test)}")
print(f"Neural Network: {mlp.score(X_test, y_test)}")
```

```
↳ Logistic Regression: 0.956140350877193
   Neural Network: 0.9298245614035088
```

```
!pip install eli5
```

```
↳ Collecting eli5
   Downloading eli5-0.16.0-py2.py3-none-any.whl.metadata (18 kB)
   Requirement already satisfied: attrs>17.1.0 in /usr/local/lib/python3.11/dist-packages (from eli5) (25.3.0)
   Requirement already satisfied: Jinja2>=3.0.0 in /usr/local/lib/python3.11/dist-packages (from eli5) (3.1.6)
   Requirement already satisfied: numpy>=1.9.0 in /usr/local/lib/python3.11/dist-packages (from eli5) (2.0.2)
   Requirement already satisfied: scipy in /usr/local/lib/python3.11/dist-packages (from eli5) (1.15.3)
   Collecting scikit-learn>=1.6.0 (from eli5)
     Downloading scikit_learn-1.6.1-cp311-cp311-manylinux_2_17_x86_64.manylinux2014_x86_64.whl.metadata (18 kB)
     Requirement already satisfied: graphviz in /usr/local/lib/python3.11/dist-packages (from eli5) (0.20.3)
     Requirement already satisfied: tabulate>=0.7.7 in /usr/local/lib/python3.11/dist-packages (from eli5) (0.9.0)
     Requirement already satisfied: MarkupSafe>=2.0 in /usr/local/lib/python3.11/dist-packages (from Jinja2>=3.0.0->eli5) (3.0.2)
     Requirement already satisfied: joblib>=1.2.0 in /usr/local/lib/python3.11/dist-packages (from scikit-learn>=1.6.0->eli5) (1.5.0)
     Requirement already satisfied: threadpoolctl>=3.1.0 in /usr/local/lib/python3.11/dist-packages (from scikit-learn>=1.6.0->eli5) (3.6.0)
     Downloading eli5-0.16.0-py2.py3-none-any.whl (108 kB)
       108.4/108.4 kB 4.7 MB/s eta 0:00:00
   Downloading scikit_learn-1.6.1-cp311-cp311-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (13.5 MB)
       13.5/13.5 MB 66.8 MB/s eta 0:00:00
   Installing collected packages: scikit-learn, eli5
   Successfully installed eli5-0.16.0 scikit-learn-1.6.1
```

```
import matplotlib_venn as vn
```

```
# Compare accuracy
print(f"Logistic Regression: {lr.score(X_test, y_test)}")
print(f"Neural Network: {mlp.score(X_test, y_test)}")
```

```
↳ Logistic Regression: 0.956140350877193
   Neural Network: 0.9298245614035088
```

```
# Calculate scores
lr_score = lr.score(X_test, y_test)
mlp_score = mlp.score(X_test, y_test)
```

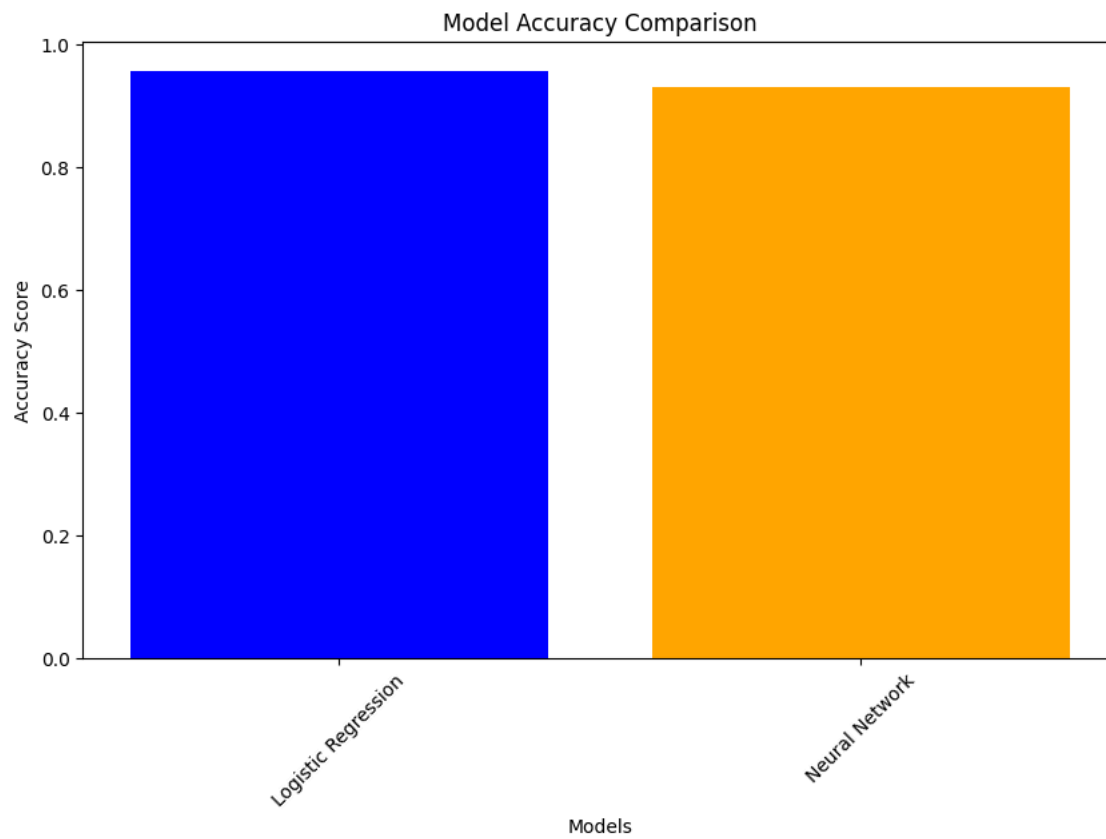
```
# Compare accuracy
print(f"Logistic Regression: {lr_score}")
print(f"Neural Network: {mlp_score}")
```

```
↳ Logistic Regression: 0.956140350877193
   Neural Network: 0.9298245614035088
```

```
import matplotlib.pyplot as plt
```

```
# Create a list of model names
models = ['Logistic Regression', 'Neural Network']
```

```
# Plot the accuracy comparison
plt.figure(figsize=(10, 6))
plt.bar(models, [lr_score, mlp_score], color=["blue", "orange"])
plt.title("Model Accuracy Comparison")
plt.xlabel("Models")
plt.ylabel("Accuracy Score")
plt.xticks(rotation=45)
plt.show()
```



Start coding or [generate](#) with AI.