Neubaitics

HealthCure Medical Project Presentation

Detection of Health Conditions Using Data Analysis

Presented by Rethanya M

Neubaitics

Introduction

Purpose of the Project:

To develop advanced methods for detecting various health conditions early using data analysis.

Importance of Early Detection in Healthcare

- Early detection can save lives by identifying health issues before they become severe.
- It allows for timely treatment, improving patient outcomes and reducing healthcare costs.

Health Condition Analyzed

01 Covid Detection

02 Alzheimer Detection

03 Brain Tumor Detection

04 Breast Cancer Detection

05 Pneumonia Detection

06 Heart Disease Detection

07 Diabetes Detection

Project Methodology

01

Data Collection:

Sources:

- Public health databases

Types of Data:

- Lab test results
- Imaging data

02

Data Preprocessing:

Cleaning:

- Removing duplicates Handling missing values.

Preparation:

- Normalizing numerical features-ie-*min-max*
- Encoding categorical variables-ie, *one-hot*

03

Feature Selection:

- **Identifying key features for each condition**
- **Example: Age, blood pressure, cholesterol levels for such disease**

Project Methodology

04

Data Visualization:

- To illustrate the patterns and insights from the data

1. Histogram
2. Bar Charts
3. etc

05

Model Selection:

Algorithms Used:

- Logistic Regression
- Random Forest
- Support vector machine
- Decision Tree
- Convolutional Neural Networks

06

Evaluation Metrics:

- **Accuracy:** Measure of correct predictions.
- **Precision:** Measure of true positive predictions.
- **Recall:** Measure of identified actual positives.

WORKFLOW MAP

DATA COLLECTION

```
dataset = pd.read_csv("/content/drive/MyDrive/Csv  files/heart.csv")
```

```
dataset.head(5)
```

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target
0	63	1	3	145	233	1	0	150	0	2.3	0	0	1	1
1	37	1	2	130	250	0	1	187	0	3.5	0	0	2	1
2	41	0	1	130	204	0	0	172	0	1.4	2	0	2	1
3	56	1	1	120	236	0	1	178	0	0.8	2	0	2	1
4	57	0	0	120	354	0	1	163	1	0.6	2	0	2	1

Gathering data from various sources.
Sources:

- Public health databases

WORKFLOW MAP

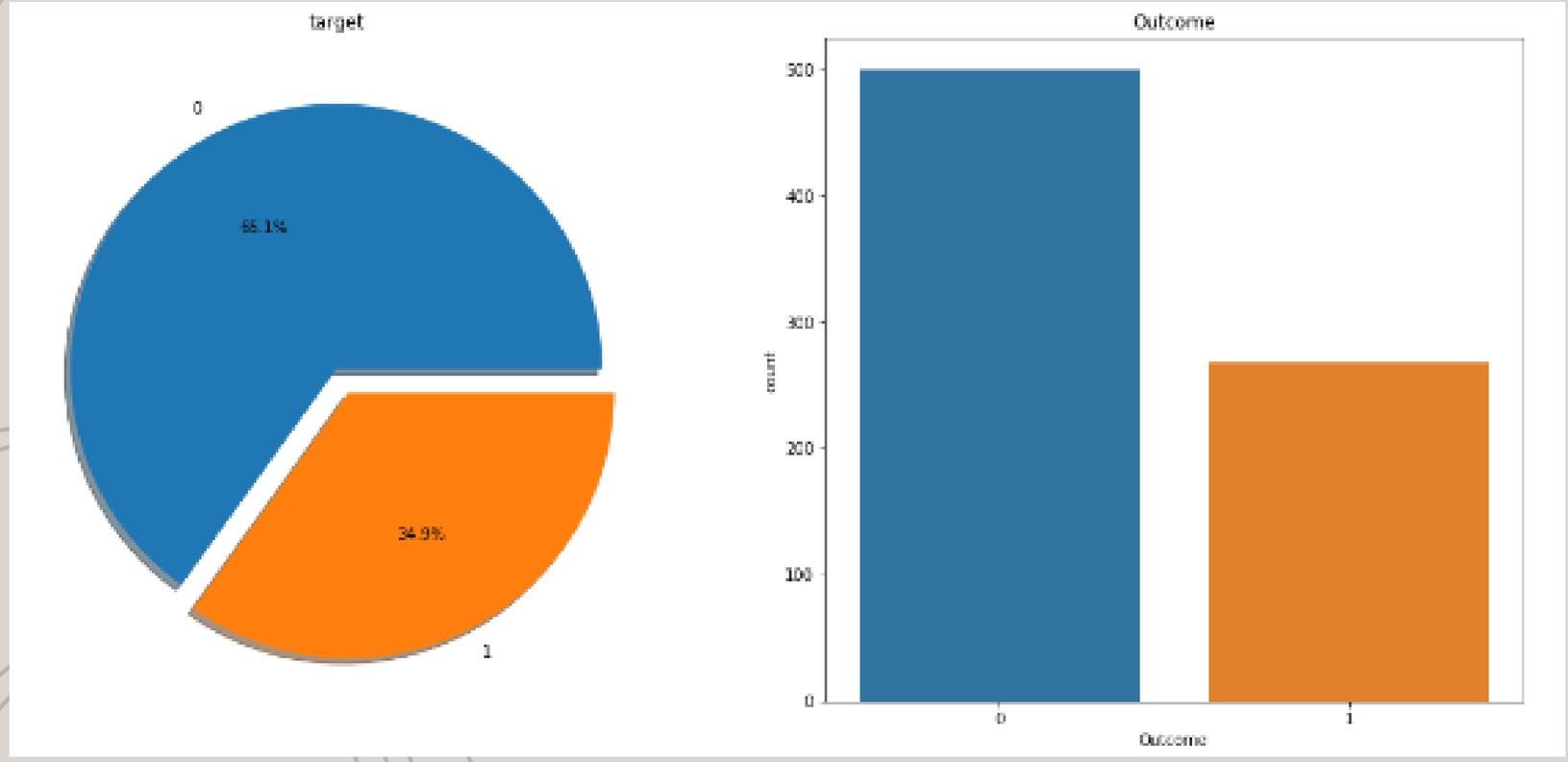
DATA PREPROCESSING

```
# Replace '?' with NaN
df.replace('?', np.nan, inplace=True)

# Option 1: Drop rows with missing values
df.dropna(inplace=True)

# Option 2: Fill missing values (e.g., with the mean)
 df.fillna(df.mean(), inplace=True)
```

It is required tasks for cleaning the data and making it suitable for a machine learning model which also increases the accuracy

WORKFLOW MAP

Data Visualization



By using data visualization techniques, analysts can identify trends, patterns, in the data and communicate these insights to stakeholders with easily understandable.

WORKFLOW MAP

Feature Selection

```
X_train = sc.fit_transform(X_train)
X_test = sc.transform(X_test)

print(X_train)
print(X_test)
```

```
[[ 0.20742038 -1.21557539 -0.70263971 ... -0.15118275 -0.5998845
  -0.3456841 ]
 [-0.07437241 -0.85989514 -0.70263971 ... -0.58327281 -0.5998845
  -0.3456841 ]
 [ 0.28648974  1.98554685  2.26183069 ...  2.87344764  2.49421575
   0.99176619]
 ...
 [-0.91775601  1.98554685  0.61490269 ... -0.15118275  0.08769334
   4.33539192]
 [ 0.18449704 -0.14853464 -0.70263971 ... -0.58327281 -0.5998845
  -0.3456841 ]
 [ 0.1341195   1.98554685  2.26183069 ...  2.00926753  1.80663792
   4.33539192]]
[[-0.64883356  1.27418635 -0.04386851 ... -0.15118275  0.08769334
  -0.3456841 ]
 [ 0.16311236  1.27418635  1.60305949 ...  1.57717747  1.80663792
   3.66666677]
```

Normalization is used to scale numerical features to a common range, typically [0, 1] to ensure that all features contribute equally to the analysis and used to improve the performance of many machine learning algorithms.

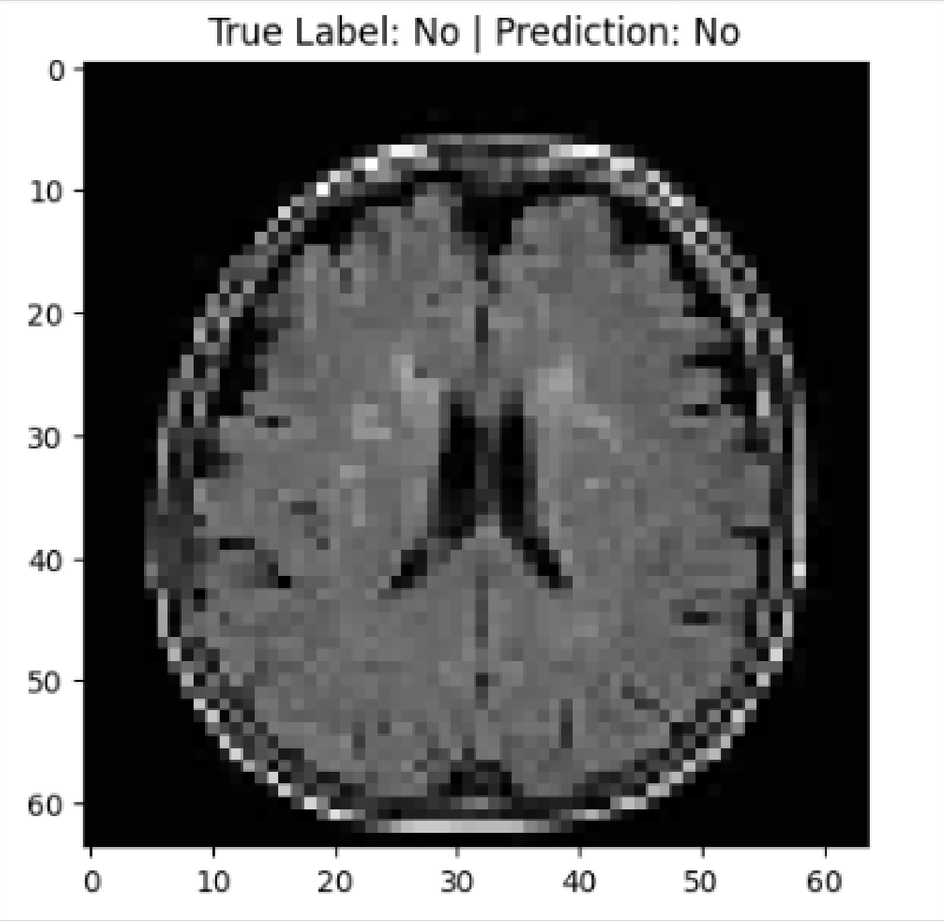
WORKFLOW MAP

Model Training

```
from sklearn.model_selection import train_test_split

predictors = dataset.drop("target",axis=1)
target = dataset["target"]

X_train,X_test,Y_train,Y_test = train_test_split(predictors,target,
                                                 test_size=0.20,random_state=0)
```

- Divide the data into training and testing sets.
- Choosing appropriate machine learning algorithms (e.g., Logistic Regression, Decision Trees,CNN,..)
- Training Process: Feed the training data into the algorithm to learn the relationships between the input features and the target outcomes.

Model Training-Deep Learnng



- Training a CNN model to predict the presence of a disease (e.g., brain tumor) using labeled image data.
- Load and preprocess the image to match the input size required by the CNN model (e.g., 64x64 pixels).
- Predict the label for each selected image.
- Display the image with its true label and predicted label for visual comparison.

WORKFLOW MAP

Model Evaluation

lr	Logistic Regression	0.7038	0.8514	0.7038	0.7163	0.7017
rf	Random Forest Classifier	0.6470	0.7841	0.6470	0.6555	0.6445
dt	Decision Tree Classifier	0.6239	0.6908	0.6239	0.6373	0.6218
svm	SVM - Linear Kernel	0.5102	0.0000	0.5102	0.4506	0.4173

- Evaluate how well the trained model performs on unseen data.

ie., Accuracy, Precision, Recall, F1 Score

Result--Getting Input from User for Prediction(Heart Detection)

```
user_data = {
    'age': 35,
    'sex': 1,
    'cp': 0,
    'trestbps': 120,
    'chol': 230,
    'fbs': 0,
    'restecg': 1,
    'thalach': 170,
    'exang': 0,
    'oldpeak': 1.0,
```

```
rf = RandomForestClassifier(random_state=best_x)

rf.fit(X_train, Y_train)

# Make prediction for user input
user_prediction = rf.predict(user_df)

# Output the prediction
if user_prediction[0] == 1:
    print("Based on the input data, it is predicted that you have the disease.")
else:
    print("Based on the input data, it is predicted that you do not have the disease.")
```

```
Based on the input data, it is predicted that you have the disease.
```

This provides a clear explanation of how user input is processed, how the model is trained and used for prediction,

Result--Giving User Input for Prediction(Diabetes Detection)

```
input_data = (5,166,72,19,175,25.8,0.587,51)

# changing the input_data to numpy array
input_data_as_numpy_array = np.asarray(input_data)
```

```
if (prediction[0] == 0):
  print('The person is not diabetic')
else:
  print('The person is diabetic')
```

```
[[ 0.3429808   1.41167241  0.14964075 -0.09637905  0.82661621 -0.78595734
   0.34768723  1.51108316]]
[1]
The person is diabetic
```

Predicted Based on the given user input and converted into Normalisation ,then evaluated model predicted whether is person is having diabetes

Recommendation

1.Covid Detection:

Early Detection and Testing:
Implement widespread testing and contact tracing to quickly identify and isolate Covid-19 cases. Let say In South Korea, extensive testing and contact tracing helped control the spread of Covid-19 early in the pandemic.

2.Alzheimer Detection

Cognitive Screening Programs:

Establish regular cognitive screening for individuals over 65 ages to detect early signs of Alzheimer's.

These tests are typically administered by healthcare professionals and can be done in various settings such as clinics, hospitals, or during routine health check-ups.

Recommendation

3.BrainTumor Detection

Using MRI and CT scans combined with AI to detect and classify brain tumors accurately. letsay In some Hospitals uses advanced MRI technology and AI algorithms to distinguish between different types of brain tumors.

4.Breast Cancer Detection

AI-Enhanced Image Analysis: Use AI to analyze mammogram images and identify early signs of cancer. LetsayIn american cancer society recommends annual mammograms for women aged 45-54 and biennial screenings for those 55 and older.

Recommendation

5.Pneumonia Detection

Recommend techniques for improving image clarity, emphasize early detection features.

6.Heart Disease Detection

Wearable Health Monitors-

smart watches:
Promote the use of wearable devices like smart watches to monitor heart health and detect abnormalities. Let say The Apple Watch includes an ECG feature that can notify users of potential heart issues.

Recommendations

- **Early Intervention**: Prioritize high-risk patients.
- **Lifestyle Changes**: Advise on diet and exercise.
- Follow-Up: Regular monitoring for high-risk profiles.
- Telehealth for Management:

Conclusion

- **User Input Processing:** Structuring user data enables seamless integration into predictive models, facilitating real-time decision-making.
- **Model Effectiveness:** Trained on labeled data, the Random Forest Classifier offers reliable predictions validated by performance metrics.
- **Clinical Benefits:** Identifying high-risk individuals enhances patient care through targeted interventions and resource optimization.
- **Future Directions:** Overall Builds trust and promotes ethical use of predictive analytics in healthcare.

Neubaitics

Thank You

Presented by Rethanya