SRP
Parkinson's disease Detection

Using Machine Learning

Mohamed Muzzamil S
2018115057

Problem Statement

- To detect whether the patient is prone to Parkinson's disease as an early diagnosis step.
- To extract and manipulate the gait analysis data of patients suffering from Parkinson's disease.
- The data analysis highlights the acceleration of knee and thigh which helps to find the freeze condition.

Existing Solutions

- Voice detection system to find the patients affected with Parkinson's disease.
- Tremor Sensoring system to detect shaking movements of the patients body.

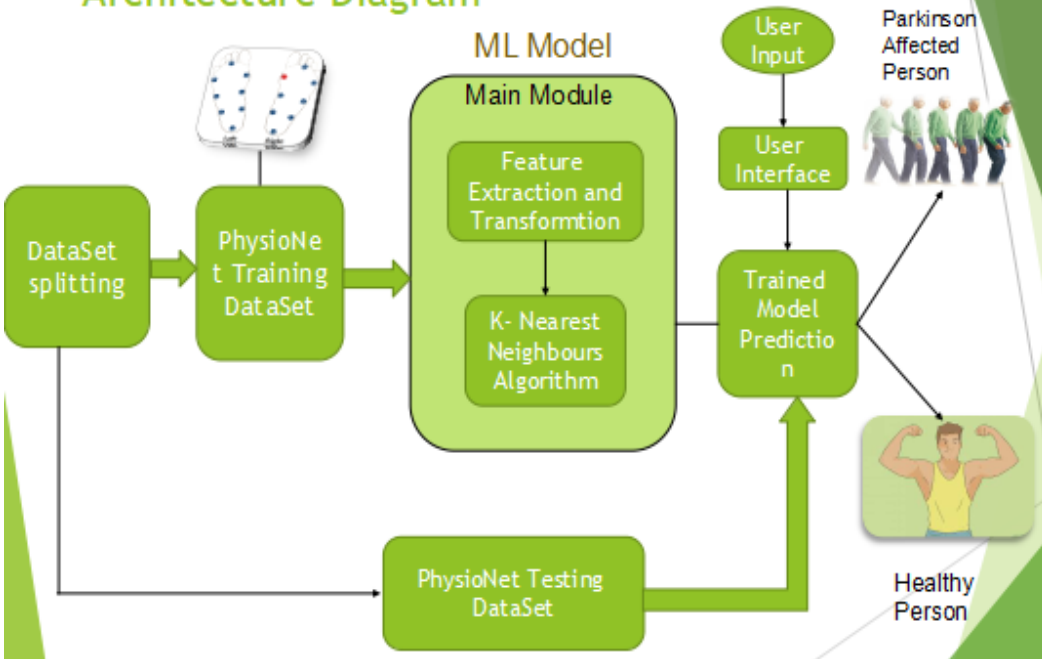
Proposed Solution

- Gait analysis (for eg: knee and thigh accelerations,vertical ground force reaction).
- Using machine learning algorithm to extract and manipulate gait analysis data.
- To find how gait of a patient affected with Parkinson's disease different from a normal patient.

Methodology

- Dataset is extracted from PhysioNet Parkinson Database.
- Algorithm Used: K-Nearest Neighbours
- By using this algorithm ,data is manipulated to find whether the patient is affected with Parkinson disease .

Architecture Diagram



Machine Learning Model (K Nearest Neighbours)

```
import pickle
```

```
import numpy as nm
import matplotlib.pyplot as plt
import pandas as pd


data_set= pd.read_csv('par.csv')
print(data_set.head())

 #independent and dependent
x= data_set.iloc[ :,0:8].values
y= data_set.iloc[:, 8].values


# Split the dataset into training and test set
from sklearn.model_selection import train_test_split
x_train, x_test, y_train, y_test= train_test_split(x, y, test_size= 0.25,
random_state=0)
cl=0.43
#feature Scaling
from sklearn.preprocessing import StandardScaler
st_x= StandardScaler()
x_train= st_x.fit_transform(x_train)
x_test= st_x.transform(x_test)

#print(x_test.shape)

#fit knn classifier to the training set
from sklearn.neighbors import KNeighborsClassifier
classifier= KNeighborsClassifier(n_neighbors=18, p=2 )
classifier.fit(x_train, y_train)

pickle.dump(classifier,open('model.pkl','wb'))
pickle.dump(st_x,open('model1.pkl','wb'))

error_rate = []


for i in range(1, 40):

  knn = KNeighborsClassifier(n_neighbors = i)
```

```
  knn.fit(x_train, y_train)
  pred_i = knn.predict(x_test)
  error_rate.append(nm.mean(pred_i != y_test))

plt.figure(figsize =(10, 6))
plt.plot(range(1, 40), error_rate, color ='blue',
        linestyle ='dashed', marker ='o',
    markerfacecolor ='red', markersize = 10)

plt.title('Error Rate vs. K Value')
plt.xlabel('K')
plt.ylabel('Error Rate')
```

Flask Code in Python(to integrate front end with back end)

```
import numpy as np
from flask import Flask, request, jsonify, render_template
import pickle

app = Flask(__name__)
model = pickle.load(open('model1.pkl', 'rb'))

@app.route('/')
def home():
    return render_template('index.html')

@app.route('/predict',methods=['POST'])
def predict():
    '''
    For rendering results on HTML GUI
    '''
    int_features = [float(x) for x in request.form.values()]
    final_features = [np.array(int_features)]
    prediction = model.predict(final_features)

    output = round(prediction[0], 2)

    return render_template('index.html', prediction_text='Probabilty of being affected with
Parkinson Disease: {}'.format(output))

@app.route('/predict_api',methods=['POST'])
def predict_api():
    '''
```

```
    For direct API calls trought request
    '''
    data = request.get_json(force=True)
    prediction = model.predict([np.array(list(data.values()))])

    output = prediction[0]
    return jsonify(output)

if __name__ == "__main__":
    app.run(debug=True)
```

Index.html

```
<!DOCTYPE html>
<html >
<!--From https://codepen.io/frytyler/pen/EGdtg-->
<head>
  <meta charset="UTF-8">
  <title>ML API</title>
  <link href='https://fonts.googleapis.com/css?family=Pacifico' rel='stylesheet' type='text/css'>
<link href='https://fonts.googleapis.com/css?family=Arimo' rel='stylesheet' type='text/css'>
<link href='https://fonts.googleapis.com/css?family=Hind:300' rel='stylesheet' type='text/css'>
<link href='https://fonts.googleapis.com/css?family=Open+Sans+Condensed:300'
rel='stylesheet' type='text/css'>
<link rel="stylesheet" href="{{ url_for('static', filename='style.css') }}">

</head>

<body>
 <div class="login">
          <h1>Parkinson Disease Detection</h1>

     <!-- Main Input For Receiving Query to our ML -->
    <form action="{{ url_for('predict')}}"method="post">
          <input type="text" name="LeftForce1" placeholder="LeftForce1" required="required" />
        <input type="text" name="LeftForce2" placeholder="LeftForce2" required="required" />
          <input type="text" name="LeftForce3" placeholder="LeftForce3" required="required" />
       <input type="text" name="LeftForce4" placeholder="LeftForce4" required="required" />
       <input type="text" name="RightForce1" placeholder="RightForce1" required="required" />
       <input type="text" name="RightForce2" placeholder="RightForce2" required="required" />
       <input type="text" name="RightForce3" placeholder="RightForce3" required="required" />
       <input type="text" name="RightForce4" placeholder="RightForce4" required="required" />
<p>   {{ prediction_text }}</p>
```

```
        <button type="submit" class="btn btn-primary btn-block btn-large">Predict</button>
    </form>

   <br>
   <br>


 </div>


</body>
</html>
```

Style.css

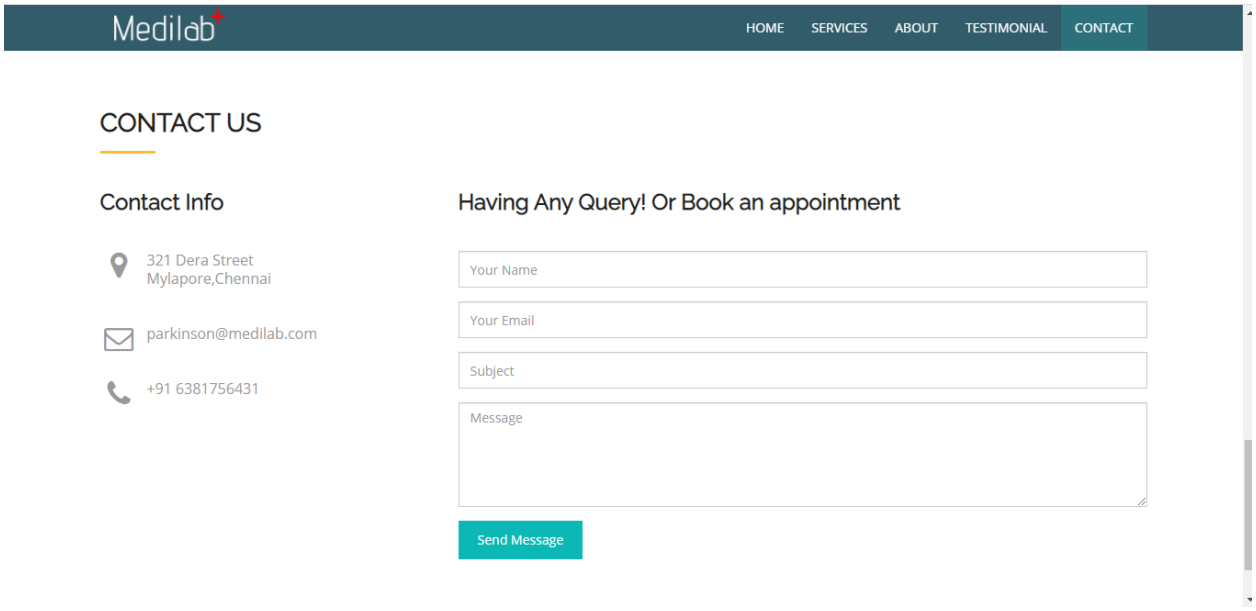
```
html {  overflow-y:scroll; }
body{
    margin: 0;
    padding: 0;
    background: url(image.jpg);
    background-size: 100% 100%;
    background-position: center;
    background-repeat: no-repeat;
    background-attachment: fixed;
    font-family: sans-serif;
}
.login-box{
    width: 320px;
    height: 760px;
    background: rgba(0, 0, 0, 0.5);
    color: #fff;
    top: 50%;
    left: 50%;
    position: absolute;
    transform: translate(-50%,-50%);
    box-sizing: border-box;
    padding: 70px 30px;
}
.avatar{
    width: 100px;
    height: 100px;
    border-radius: 50%;
    position: absolute;
    top: -50px;
```

```
    left: calc(50% - 50px);
}
h1{
    margin: 0;
    padding: 0 0 20px;
    text-align: center;
    font-size: 22px;
}
.login-box p{
    margin: 0;
    padding: 0;
    font-weight: bold;
}
.login-box input{
    width: 100%;
    margin-bottom: 20px;
}



.login-box input[type="text"], input[type="password"]
{
    border: none;
    border-bottom: 1px solid #fff;
    background: transparent;
    outline: none;
    height: 40px;
    color: #fff;
    font-size: 16px;
}
.login-box input[type="submit"]
{
    border: none;
    outline: none;
    height: 40px;
    background: #1c8adb;
    color: #fff;
    font-size: 18px;
    border-radius: 20px;
}
.login-box input[type="submit"]:hover
{
    cursor: pointer;
    background: #39dc79;
```

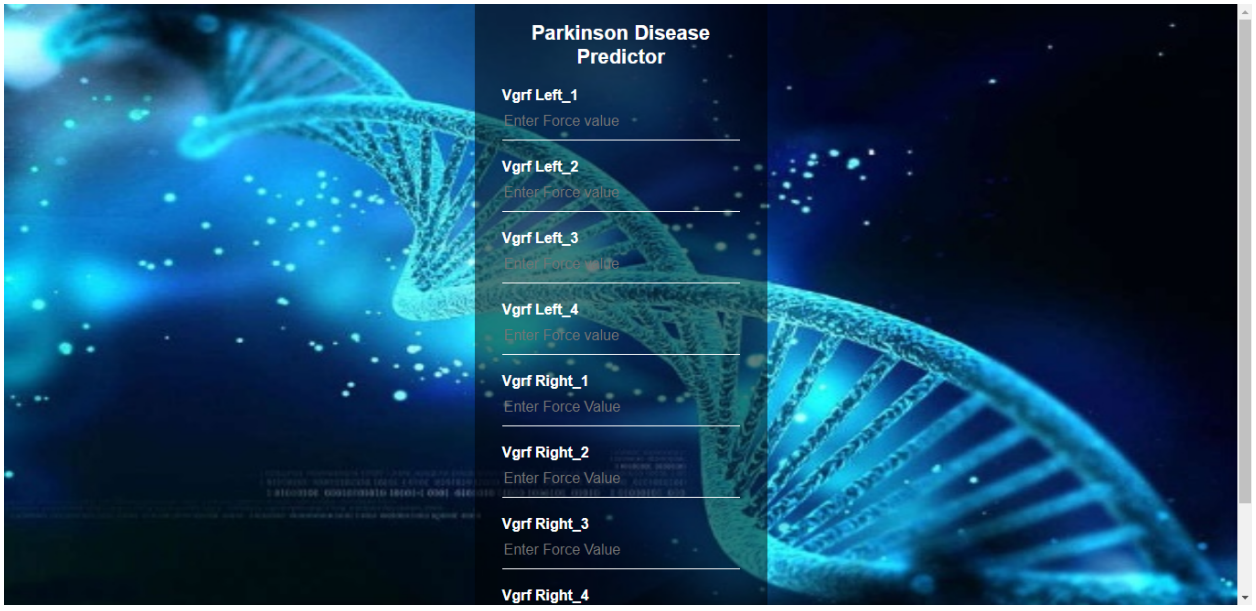
```
    color: #000;
}

.login-box a{
    text-decoration: none;
    font-size: 14px;
    color: #fff;
}
.login-box a:hover
{
    color: #39dc79;
}
```

Home Page

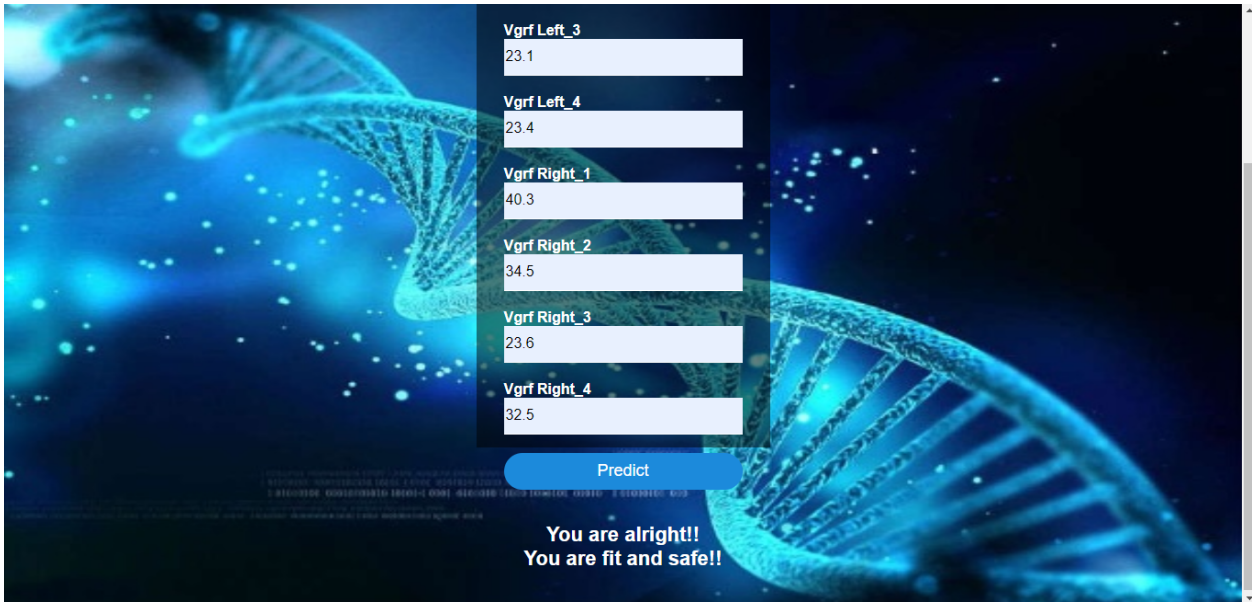
Contact Us



Prediction Page



Output


Vgrf Left_3
23.1
Vgrf Left_4
23.4
Vgrf Right_1
40.3
Vgrf Right_2
34.5
Vgrf Right_3
23.6
Vgrf Right_4
32.5
Predict
You are alright!!
You are fit and safe!!