Classification of Cardiac Arrhythmias Patients

G. Yashwanth (160116733119)

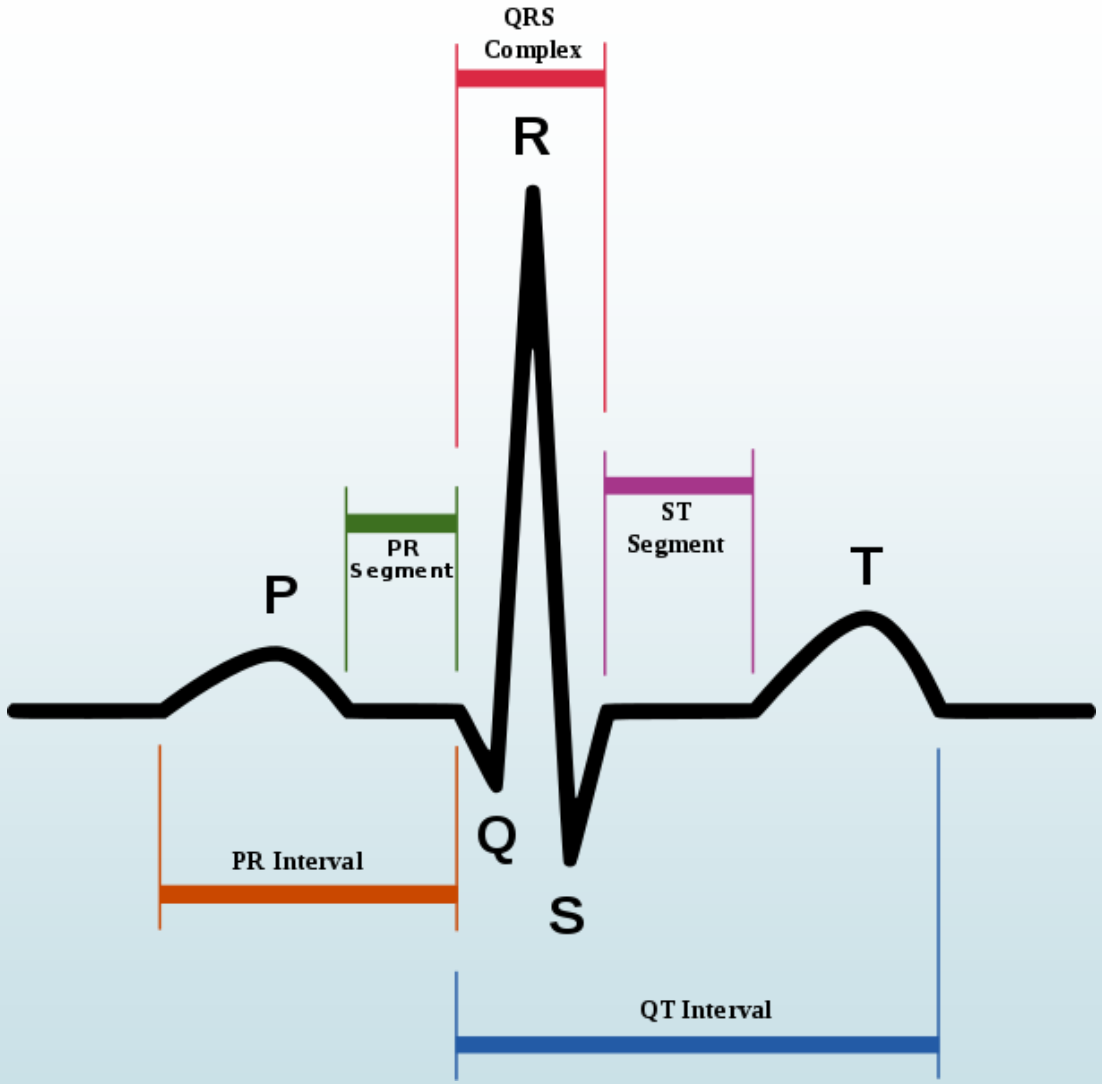
P. Dharanish Reddy (160116733182)

Scope:

- This project is used to classify the Arrhythmias patients into 16 categories
- This work has great potential to serve the medicine industry.
- It is used to detect a patient having a serious cardiac problem and can be treated in time.

ECG Graph

- P- is the atrial systole contraction pulse
- Q- is a downward deflection immediately preceding the ventricular contraction
- R- is the peak of the ventricular contraction
- S- is the downward deflection immediately after the ventricular contraction
- T- is the recovery of the ventricles
- U- is the successor of the T wave but it is small and not always observed



Data:

75,0,190,80,91,193,371,174,121,-16,13,64,-
2,?,63,0,52,44,0,0,32,0,0,0,0,0,0,0,44,20,36,0,28,0,0,0
,0,0,0,52,40,0,0,0,60,0,0,0,0,0,0,52,0,0,0,0,0,0,0,0,0,
0,0,56,36,0,0,32,0,0,0,0,0,0,48,32,0,0,0,56,0,0,0,0,0,0,
80,0,0,0,0,0,0,0,0,0,0,0,0,0,40,52,0,0,28,0,0,0,0,0,0,0,48
,48,0,0,32,0,0,0,0,0,0,0,52,52,0,0,36,0,0,0,0,0,0,0,52,4
8,0,0,32,0,0,0,0,0,0,0,56,44,0,0,32,0,0,0,0,0,0,-
0.2,0.0,6.1,-1.0,0.0,0.0,0.0,0.6,2.1,13.6,30.8,0.0,0.0,1.7,-
1.0,0.6,0.0,1.3,1.5,3.7,14.5,0.1,-
5.2,1.4,0.0,0.0,0.0,0.0,0.8,-0.6,-10.7,-15.6,0.4,-
3.9,0.0,0.0,0.0,0.0,0.0,-0.8,-1.7,-10.1,-22.0,0.0,0.0,5.7,-
1.0,0.0,0.0,-0.1,1.2,14.1,22.5,0.0,-
2.5,0.8,0.0,0.0,0.0,1.0,0.4,-4.8,-2.7,0.1,-
6.0,0.0,0.0,0.0,0.0,0.0,-0.8,-0.6,-24.0,-29.7,0.0,0.0,2.0,-
6.4,0.0,0.0,0.2,2.9,-12.6,15.2,-0.1,0.0,8.4,-
10.0,0.0,0.0,0.6,5.9,-3.9,52.7,-0.3,0.0,15.2,-
8.4,0.0,0.0,0.9,5.1,17.7,70.7,-0.4,0.0,13.5,-
4.0,0.0,0.0,0.9,3.9,25.5,62.9,-0.3,0.0,9.0,-
0.9,0.0,0.0,0.9,2.9,23.3,49.4,8

data preprocessing

- data cleaning

1. Removing duplicate values.
2. Removing null values.

- Feature selection
- Removing overfitting

Feature selection

```
# Split-out validation dataset
array = df.values
Y = array[:,279]
Y=Y.astype('int')
X = array[:,0:279]
#print(X[0])
from sklearn.feature_selection import RFE
model = LogisticRegression()
rfe = RFE(model, 30)
fit = rfe.fit(X, Y)
print("Num Features: %d" % fit.n_features_)
print("Selected Features: %s" % fit.support_)
print("Feature Ranking: %s" % fit.ranking_)
```

```
Num Features: 30
Selected Features: [False False False False False False False False False False False False False
 False False False False False False False False False False False False False
 False False False False False False False False False False  True False
 False False False False False False False False False False False False
 False False False False False False False False False False False False
 False False False False False False False False False False False False
 False False False False False False False False False False False False
 False False False False False False False False False False False False
 False False False False False False False False False False False False
 False False False False False False False False False False False False
 False False False False False False False False False False False False
 False False False False False False False False False False False False
 False False False False False False False False False False False False
 False False False False  True  True False False False  True  True False
 False  True  True False False False False  True False False False  True
  True False False False False  True False False  True  True  True
  True False False  True False False False False False False False False
 False False False False False  True False False False False False False
 False False False  True False  True False  True False False False False
 False False False False False  True False  True False False False  True
  True False False False False False False False False False False False
 False False False  True False False False False False False False False
 False False  True False False  True False  True False False False False
  True False False]
Feature Ranking: [124  84 179 155  55 159 137 176 141 168 191 158 120 181  29  56  69 162
  61 243  53 197 149 205  14 233 226  54 106 100  20 188  73 199   1 154
 207 225  67 132 163 126  40 108 129 223 146 184   4  37  27  89  74  88
  71 130  99 139 214 200  11 190 153  26 133 101 105 248  80 249 152 186
 220 121 213 145  76  75  82  70  43 211  92 235 127 202 231  60  57  58
  79 119 135 222 107 193 156  22 174  42  96  44  93 196  94 192 111  48
  17 206 177  18  47  86 114 217 147 166 230  81  21 189 112  98  32  83
  62 102 173 229 169 242 247 198 209  50  36 164 115 250  35 246 224 240
 208 237 221  49  24 103 140 244  52 165 215 150 241 238  85 167   1   1
   6 182 234   1   1  66  65   1   1  77 151 178 210   1 138  78 113   1
   1  10  15 203 216   5   1  68  34   1   1   1   1 183 228   1 161 131
  90 185 157   2   8 195 239 104 143  39  87   1   7   9 142 148 204  19
 125  33  38   1  16   1  23   1 194  91  59  45  97 116  30  13  28   1
 227   1 118  46  95   1   1 117  72 201 232 171  12  51 123   3 160  41
 110 122 218   1 134 109 172   1 136 180 144 212 245 175   1  31  64   1
 187   1 128 219 236 170   1  25  63]
```

Training

```
for name, model in models:
    kfold = model_selection.KFold(n_splits=10, random_state = seed)
    cv_results = model_selection.cross_val_score(model, X_train, Y_train, cv=kfold, scoring=scoring)
    results.append(cv_results)
    names.append(name)
    msg = "%s: %f (%f)" % (name, cv_results.mean(), cv_results.std())
    print(msg)
```

```
LR: 0.629129 (0.089640)
KNN: 0.601051 (0.075987)
SVM: 0.540015 (0.052591)
```

Testing

```
# Make predictions on validation dataset
out=["Normal","Ischemic changes","Old Anterior Myocardial Infarction","Old Inferior Myocardial Infarction","Sinus tachycardy","Sinus bradycardy","Ventricular Premature Contraction","Supraven
for name, model in models:
    model.fit(X_train, Y_train)
    predictions = model.predict(X_validation)
    print(name)
    print(accuracy_score(Y_validation, predictions))
    #for i in predictions:
    #  print(out[i-1]+" ")
    print(predictions)
    #print(Y_validation)
    #print(classification_report(Y_validation, predictions))
```

```
LR
0.6373626373626373
[10  1  2  1  1  1  1  6  1  2  6  1 10  1  1 10  1 16  2  1  5  1  1 16
  6  1  1  1  1  1  1  6 10  1  5  2  2  2  1  1  1  1  1  1  1  1 10
  1  2  1  1  1 10  1  1  1  1 10  6  1 16  1 10  1  1 16  1  1  1  1  1
 10  1  1  1  1  1  1  1  1  1  1 16  2 16  1  5  1  1  1]
KNN
0.5934065934065934
[ 1  1  1  1  1  1  1  1  1  2  1  1  1  1  1 10  1  3  1  1  1  1  1  1
  1  1  1  1  1  1  1  1  1  1  1  2  2  2  1  1  1  1  1  1  1  1  1  1
  1  1  1  1  1 10  1  1  1  1 10  1  1  1  1  1  1  1  1  1  1  1  1  1
  1  1  1  1  1  1  1  1  1  1  1  2  1  1  1 10  1  1  1]
SVM
0.5494505494505495
[1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1]
```

Output

```
LR
0.6373626373626373
['Right bundle branch block ', 'Normal ', 'Ischemic changes ', 'Normal ', 'Normal ', 'Normal ', 'Normal ', 'Sinus bradycardy ', 'Normal ', 'Ischemic changes ', 'Sinus bradycardy ',
[10  1  2  1  1  1  1  6  1  2  6  1 10  1  1 10  1 16  2  1  5  1  1 16
  6  1  1  1  1  1  1  6 10  1  5  2  2  2  1  1  1  1  1  1  1  1  1 10
  1  2  1  1  1 10  1  1  1  1 10  6  1 16  1 10  1  1 16  1  1  1  1  1
 10  1  1  1  1  1  1  1  1  1  1 16  2 16  1  5  1  1  1]
```

References:

https://www.verywellhealth.com/overview-of-cardiac-arrhythmias-1746267
https://www.medicalnewstoday.com/articles/8887.php
http://cs229.stanford.edu/proj2014/AlGharbi%20Fatema,%20Fazel%20Azar,%20Haider%20Batool,%20Cardiac%20Arrhythmias%20Patients.pdf
https://pandas.pydata.org/pandas-docs/stable/
https://keras.io/
https://scikit-learn.org/stable/documentation.html
https://arxiv.org/abs/1801.10033

THANK YOU