RunWalk Classification

27.11.2022

Vivek Viradia
Rubixe
1st Cross Road, Kudlu Gate, 560068
Bangalore, Karnataka

About Data

This data is collected from 36 different users as they performed some day-to-day human activities such as—walking, sitting, standing, jogging, ascending and descending stairs for a specific period of time. In all cases, data is collected at a frequency of 20 samples per second, that is one record every 50 milliseconds.The dataset has 6 columns – 'user', 'activity', 'timestamp', 'x-axis', 'y-axis', and 'z-axis'.'user' denotes the user ID, 'timestamp' is the Unix timestamp in nanoseconds, and the rest are the accelerometer readings along the x, y, and z axes/dimensions at a given instance of time. Our target variable(class-label) is 'activity' which we intend to predict.

Performance Analysis of Classification Methods

Logistic Regression

With logistic regression, a discrimination model is created according to the number of groups in the structure of the data. With this model, the new data taken into the dataset is classified. The purpose of using logistic regression is to create a model that will establish the relationship between the least variable and the most suitable dependent and independent variables.

```
from sklearn.metrics import accuracy_score,classification_report,confusion_matrix
print(confusion_matrix(Y_test,Y_pred))
print(classification_report(Y_test,Y_pred))
lracc = accuracy_score(Y_test,Y_pred)
print(lracc)
```

```
[[10722   318]
 [  673 10434]]
              precision    recall  f1-score   support

           0       0.94      0.97      0.96     11040
           1       0.97      0.94      0.95     11107

    accuracy                           0.96     22147
   macro avg       0.96      0.96      0.96     22147
weighted avg       0.96      0.96      0.96     22147

0.9552535332099156
```

Decision Tree

Decision Tree is the most powerful and popular tool for classification and prediction. A Decision tree is a flowchart-like tree structure, where each internal node denotes a test on an attribute, each branch represents an outcome of the test, and each leaf node (terminal node) holds a class label.

```
from sklearn.tree import DecisionTreeClassifier
dt = DecisionTreeClassifier(splitter='best',criterion='entropy',min_samples_split=3,min_samples_leaf=3)
dt.fit(X_train,Y_train)
Y_pred = dt.predict(X_test)
print(confusion_matrix(Y_test,Y_pred))
print(classification_report(Y_test,Y_pred))
dtacc = accuracy_score(Y_test,Y_pred)
print(dtacc)
```

```
[[10864   176]
 [  182 10925]]
              precision    recall  f1-score   support

           0       0.98      0.98      0.98     11040
           1       0.98      0.98      0.98     11107

    accuracy                           0.98     22147
   macro avg       0.98      0.98      0.98     22147
weighted avg       0.98      0.98      0.98     22147

0.983835282431029
```

XG Boost

The XGBoost (eXtreme Gradient Boosting) is a popular and efficient open-source implementation of the gradient boosted trees algorithm. Gradient boosting is a supervised learning algorithm that attempts to accurately predict a target variable by combining an ensemble of estimates from a set of simpler and weaker models.

```
from xgboost import XGBClassifier
gb = XGBClassifier()
gb.fit(X_train,Y_train)
Y_pred = gb.predict(X_test)
print(confusion_matrix(Y_test,Y_pred))
print(classification_report(Y_test,Y_pred))
gbacc = accuracy_score(Y_test,Y_pred)
print(gbacc)
```

```
[[10953    87]
 [   98 11009]]
              precision    recall  f1-score   support

           0       0.99      0.99      0.99     11040
           1       0.99      0.99      0.99     11107

    accuracy                           0.99     22147
   macro avg       0.99      0.99      0.99     22147
weighted avg       0.99      0.99      0.99     22147

0.991646724161286
```

Random Forest Classifier

The purpose of the random forest classifier classification method is to bring together the decisions made by many trees trained in different training sets instead of a single decision tree.

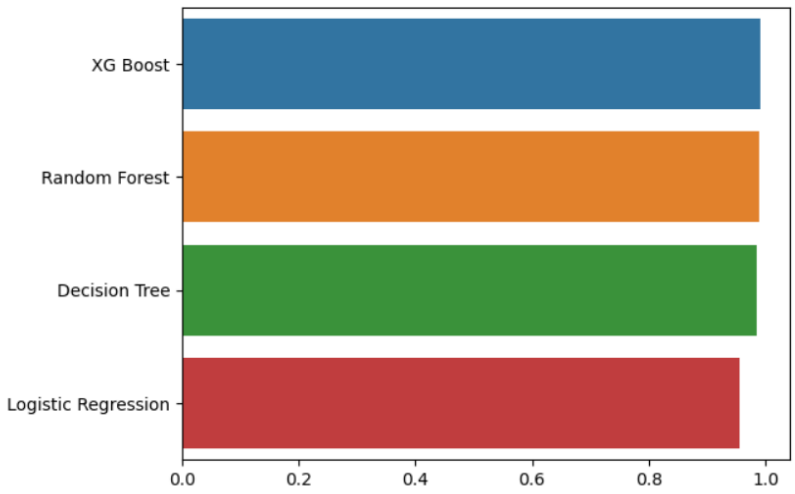
```
from sklearn.ensemble import RandomForestClassifier
rf = RandomForestClassifier(n_estimators=200,min_samples_split=2,min_samples_leaf=1,max_features='sqrt',criterion='entropy')
rf.fit(X_train,Y_train)
Y_pred = rf.predict(X_test)
print(confusion_matrix(Y_test,Y_pred))
print(classification_report(Y_test,Y_pred))
rfacc = accuracy_score(Y_test,Y_pred)
print(rfacc)
```

```
[[10929   111]
 [  109 10998]]
              precision    recall  f1-score   support

           0       0.99      0.99      0.99     11040
           1       0.99      0.99      0.99     11107

    accuracy                           0.99     22147
   macro avg       0.99      0.99      0.99     22147
weighted avg       0.99      0.99      0.99     22147

0.990066374678286
```

Conclusion And Findings

```
sns.barplot(x=models.Accuracy.sort_values(ascending=False).values,y = models.Accuracy.sort_values(ascending=False).index)
```

```
<AxesSubplot:>
```



1. **Accuracy of Logistic Regression 0.955**
2. **Accuracy of Random Forest 0.990**
3. **Accuracy of XG Boost 0.991**
4. **Accuracy of Decision Tree 0.983**

Based on the observations above, we can conclude that XG Boost is the best-fitting model with 99.1% accuracy for the given problem.

GitHub Link for Source Code:-
https://github.com/VivekViradia/Walk_run_Classification