Improvi goodas as pol import warnings Warrings. Litteswarnings ('ignore') from Albarn. Metrics i Proport accuracy-score, classification-report from Aleann. Orsemble Propost Gradient Boosting Classifier, Adviscost Classifier John Splean, linear-model Emport Light He Regression (V. from . Okteurn, model - Selection import train test - eplet, Gridsauch Joan goleann tree Import Decision traclassifier Allean. proprocessing Proport label Encoder. Step: 1 Original = pd. read_cov("forman_Activily_Data. (SV") organal head) Original - columne Original. shape Ori opinal. Alypes. Original, into() Original. value-counts() label_encoder: Label Encoder() Orreginal ["Aabd'-Activity"]= label-encodes. fet-transform (original ("Activity")) Stepia Oxiginal. Activity- Value_counts() original. label_ Activity. value - counts()

0

Lab10. Patients Physical Activities Prediction using Boosting

Objectives

In this lab, you will recognize physical activities such as laying, 'sitting' or 'walking' using Gradient Boosting, AdaBoost and Voting Classifiers.

Learning Outcomes

After completing this lab, you will be able to

- Create a small dataset with selected rows based on fewer target labels
- Build GradientBoostingClassifier, fit and predict on test data
- Print accuracy and classification report
- Find the best no. of decision trees and learning rate using GridSearch and Cross Validation
- Build AdaBoost classifier model with GridSearchEV, fit and predict
- Select best parameter values for n_estimators and learning_rate
- Build LogisticRegressionCV model, fit, predict and print scores
- Build VotingClassifier using other models, fit, predict and print scores
- Interpret results and parameter values
- Change parameter values and play around with models

Business Use Case

As you a data scientist, one popular hospital in your city has asked you to design a model based on the mobile phone data of their patients. Your model will help the hospital to understand the physical activity level of their patients. The physical activities are classified into 3 levels – Laying, Sitting and Walking, flased on the prediction, Doctors will recommend health diets, exercise and physical activities to the new patients.

Dataset

You will use a dataset named, Human_Activity_Data.csv.

Step1. [Understand Data]

Using Pandas, import "Human_Activity_Data.csv" file and print properties such as head, shape, columns, dtype, info and value_counts.

Step2. [Build a small dataset]

- As it is a big dataset, execution time will be long for training and testing. So build a small dataset
 with only 3 classes, laying, sitting and walking, where each class with 500 samples. So, shape of
 this new dataframe will be (1500, 563). That is 1500 rows and 563 features.
- Store this new dataframe as a CSV file.

Step3. [Build GradientBoostingClassifier]

- · Import your new reduced CSV file
- Print basic properties of the new CSV file
- Split it into training set and test set (30% samples for testing)
- Create GradientBoostingClassifier, fit and predict
- Print accuracy and classification report

Step4. [Find Best no. of trees and Best Learning Rate using Grid Search and Cross Validation]

- Create GridSearchCV model with GradientBoostingClassifier
- Parameters: param_grid = {'n_estimators': [50, 100, 200, 400], 'learning_rate': [0.1, 0.01]}
- Perform fit and predict
- · Print accuracy, classification report
- Print best parameters such as best no. of trees and learning rate. Use the attribute best_estimator_

DEPT OF DATA SCIENCE, BISHOP HEBER COLLEGE IT RICHY

DR.K.RAJKUMAR

Take Liket 3000 Samples! 'tem't = Original [original[' ACHVILY] = = 1 [ANIMIG] [: BOO] tem & = original [original [original [Activity] == 817Thucy] [500] fem3: Original [brigha] ['ACHRITY'] == |WALKENG'] [:500] {emy = original [original ['Activity] == ! STANDING [:500] temps = Original [original [Activity] = = WALKING . LUPSTATES tempt = Organal Conginal [IActivity] ==!WALKINGnow-d= pd. concat ([tem-1, tem 2, tem 3, temy, tem 5 tem 6] revi-afito - Cer ("human activity-disped 3000. (sv") new-df, head () new- of shape new-df. Columns new- db. dtypes new-db. value -counts() X = new-dg.drope Activity, 'lakel - Activity', axisa) y= new- of, label- ANTRHY X-train, X-test, y-train, y-test = train-test-split (X-184) Ees-Size sond, Billd . Gradlen & Broging? 0 = Gradent Boosting Classifier (Subample 50, 10, 0= estimators=100) 0 made of (x-train 14-train) y- pred = model: Prepet (x test) Print (acumacy-score (y-test 14-pred))
- Frint (classification-report (y-test 14-pred))

Step5. [Build AdaBoostClassifier]

- Create AdaBoostClassifier with DecisionTreeClassifier
- Create GridSearchCV with AdaBoostClassifier model that you created as before
- Parameters: param_grid = {'n_estimators': [100, 150, 200], 'learning_rate': [0.01, 0.001]}
- Perform fit, predict
- Print accuracy, classification report
- Print best parameters such as best no. of trees and learning rate. Use the attribute best estimator

Step6. [Build LogisticRegressionCV classifier]

- Create a LogisticRegressionCV model with the parameters Cs=5, cv=4, penalty='l2'.
- · Perform fit and predict
- Print classification report

Step7. [Build VotingClassifier]

- Build VotingClassifier model with GradientBoostingClassifier and LogisticRegressionCV that you created in the previous steps
- · Perform fit and predict operations
- Print classification report

Step8. [Interpret your results]

- Analyze your results
- · Change parameters and play with your code

base 2 = Ada Boest Classifier (base astinatox = base, random - state=0)

param = grid = g' n - estimatox d': [roc, 150, 200], 'baming xate': [0:01, 10:00]

mode | a = Croid Seasehor (base a, faram - grid, cv=10, n - gobs = -)

mode | a - fit (x - fain, y - trovin)

y-pred 2 = mode | 2 - predict (x test)

Print (accordy - score (y - test), y - pred 2))

print (classification - report (y - test, y - pred 2))

print (classification - report (y - test, y - pred 2))

print (classification - report (y - test, y - pred 2))

mode | 3 - best - estimator

Build Lagistic Regression 5

mode | 3 - fit (x - tean - y - town)

y - pred 3 = mode | 3. * predict (x - test)

operations as mode | 3. * predict (x - test)

(dicheage Store (y.fest, y pred 3)) (classification_report (y-test, y-text)) FPrd Best noing trees? towam-grid = ('n-estinators': [50,109200400], 'learning-rate' =[0,1,0,0)]} all - scores = Cross-val - Score Cestimator= model -, x=x train, y = y foatn; CV=5) Print(all - Scores) pride 14 - = Girld Saurch CV (astinator = mudel =), paran-grid = param -gra, custin= 30bs=-1) Model 4 stit (X train, g train) y- pred 4= model 4- product (x-test) Pant (accusacy geore (y-test, y-pred i)) Dant (Claserfication-resport (y-test, y-predy)) model 4- best estimator. Build Noting Classifier for 3000 Lamplest modelite = Voting Classifier (estimators = [(//, model 3_), (goc ; four a), voting = soft! made & . fit (x-tsain, y-train) 4- bred 5= mode 15-poedict (x,dest) Prent (accuracy_swee(y-test, y-pood to)) Print (classification-report (y-test, y-pred 5)) From this 2000 samples bu temp 1 = new -of thouse of CACHVity 2= LAYING (500 temp & = new db (new-df [Activity] = String[500)

Scanned with CamScanner

tem p3: now_df[new_db['Activity']==[MALKING][: 500] ds = td. concat ([temp], temps, temps]) dt to-csv ("human activity - clipped 1500. Csv") of = pol. read. cov. (". human -activity. clippedition, asv") of heads) of shape of. Columns of dtypes df. Popo() d. Value - counts) Step:3, [Build Gradient Boosting Classifier) X = do drop (TAcHrify', 'Cabo) - Activity], axes=1) y = of. label - Activity. x_toin,x_test,y_train,y_test = train, test_ split (X) y, test_stre = 002,) model = Gradient Boosting Cassifles Coubsample = 0.5, n. estimotoriston, baoring_ rate zive) model for (x-train, y-town) y - pred = model. Predict (x-lest) of brook DEPT OF DATA SCIENCE BISHOP HEBER COLLEGE! TRICHY DR.K.RAJKUMAR

areusaey - ecore (g-testry-pred) ARN + Classification - report (g. test 14- Pred) Stepty Poorain_grid = {\n_extinators'; [50,100,200,400], Learning_sate of [, 0.0] all scores-cross and -score (estimator = model, x =x-train, y=g) froun, and print (all_scores) model à s Grédiseascher (estimates = model, param-gréd sparam gréd, CV=5+n-pbs=-1) prnf(all-scres) mudo) a = Gradsearch CV (estimator = mode), param-grid-param-grid, CV=5, n-gob=-1) model 3. flt (x-train, y-train) y-pseda = modely psediet (x-fast) 4- preda accuracy-score (of-fest, y-preda) knode 1 2. best_ estimatos_ step! 5 bases Decision tree Classifier (max features-4) base 2 = Ada Boostelass Ples (base-estimator = base , random state 0) parsam-good-of in essimators fire, 150,200], learning-sate [00,00] madel 3 = Good Season CV (base a, pasam-grad, cv=10,n-jobs=-1) model 35 for (x train, y train) y-prods = models, predict (y. fest) 4 ~ pred 3. accuracy. Sure (y-test, y-pred 3) Print (classification= report (y-lest, n_towd3)

Scanned with CamScanner

```
NOTE
     model 3- best_estinator
Stap 16
     model 4 = Logrestic Regression (CV=A, C3=5, penalty='12')
     model 4. Lif (x train, y train)
     U_ predict (X-test)
      y_ pred4.
    accusing - since [ y text, y-pred 4)
    Dent (classification=report (y-test, y-possed 4))
  step) 7
    model to = Voling lawiffer (28 Hmators = [( lr', model 4))
                                 ('q be', baser), voting=ledt)
    model5, LA (x-train)
    4 psed 5=model 5. predict (x_test)
      4- psed 5
      accuracy: score (y- test 14-pred 5)
     Print (classification-report y-test, y-predto))
    Stopis
Gradient Boosting Classifier
      model 6 = Gradient Boosting Classifier (n_estimators = 50,
                                               learning tate = 1.0,
       model to tet (x train, y - train)
        y-predb=model6. predict (x-test)
                 PRISHOPHEBERGONEGE TRICHY
```

Occursacy-Score (y-test, y-prod 6)

Print (Classification-report (y-test, y-prod 6)

Ada Boost Classification-report (y-test, y-prod 6)

base 3 = Ada Boost classification - Report (bouse 3, param-grid, cv=5, n-folos=1)

model 7 = Bridieanchev (bouse 3, param-grid, cv=5, n-folos=1)

model 7 - fet (x-tran, y-train)

U-prod 7 = model 7 - product (x-test)

cy-prod 7
accuracy some (y-test,y-prod 7)

Print (classification-report (y-test,y-prod 7))