Data Mining Project Document

12/01/2020

Project Topic: An algorithm for classification using decision trees

Dataset: UCI Mushroom Data Set

This data set includes descriptions of hypothetical samples corresponding to 23 species of gilled mushrooms in the Agaricus and Lepiota Family. Each species is identified as definitely edible, definitely poisonous, or of unknown edibility and not recommended. This latter class was combined with the poisonous one. The Guide clearly states that there is no simple rule for determining the edibility of a mushroom; no rule like ``leaflets three, let it be'' for Poisonous Oak and Ivy.

1. Description of the considered problem

The Audubon Society Field Guide to North American Mushrooms states "there is no simple rule for determining the edibility of a mushroom". But machine learning can solve this issue.

2. Background

UCI dataset contains lots of features:

1. cap-shape: bell=b,conical=c,convex=x,flat=f, knobbed=k,sunken=s

2. cap-surface: fibrous=f,grooves=g,scaly=y,smooth=s

3. cap-color: brown=n,buff=b,cinnamon=c,gray=g,green=r, pink=p,purple=u,red=e,white=w,yellow=y

4. bruises?: bruises=t,no=f

5. odor: almond=a,anise=l,creosote=c,fishy=y,foul=f, musty=m,none=n,pungent=p,spicy=s

6. gill-attachment: attached=a,descending=d,free=f,notched=n

7. gill-spacing: close=c,crowded=w,distant=d

8. gill-size: broad=b,narrow=n

9. gill-color: black=k,brown=n,buff=b,chocolate=h,gray=g, green=r,orange=o,pink=p,purple=u,red=e, white=w,yellow=y

10. stalk-shape: enlarging=e,tapering=t

11. stalk-root: bulbous=b,club=c,cup=u,equal=e, rhizomorphs=z,rooted=r,missing=?

12. stalk-surface-above-ring: fibrous=f,scaly=y,silky=k,smooth=s

13. stalk-surface-below-ring: fibrous=f,scaly=y,silky=k,smooth=s

14. stalk-color-above-ring: brown=n,buff=b,cinnamon=c,gray=g,orange=o, pink=p,red=e,white=w,yellow=y

15. stalk-color-below-ring: brown=n,buff=b,cinnamon=c,gray=g,orange=o, pink=p,red=e,white=w,yellow=y

16. veil-type: partial=p,universal=u

17. veil-color: brown=n,orange=o,white=w,yellow=y

18. ring-number: none=n,one=o,two=t

19. ring-type: cobwebby=c,evanescent=e,flaring=f,large=l, none=n,pendant=p,sheathing=s,zone=z

20. spore-print-color: black=k,brown=n,buff=b,chocolate=h,green=r, orange=o,purple=u,white=w,yellow=y

21. population: abundant=a,clustered=c,numerous=n, scattered=s,several=v,solitary=y
22. habitat: grasses=g,leaves=l,meadows=m,paths=p, urban=u,waste=w,woods=d

Using data to create a decision tree seems as plausible way to deal with this problem.

3. Description of the own solution

We used XGBoost on python to create a gradient boosting model. Gradient boosting is a machine learning technique for regression and classification problems, which produces a prediction model in the form of an ensemble of weak prediction models, typically decision trees. It builds the model in a stage-wise fashion like other boosting methods do, and it generalizes them by allowing optimization of an arbitrary differentiable loss function.

XGBoost is an optimized distributed gradient boosting library designed to be highly efficient, flexible and portable.

4. Algorithmic complexity and correctness (quality) analysis

Boosting uses ensembles of models trained on resampled data and a vote to determine the final prediction. There are two key distinctions. First, the resampled datasets in boosting are constructed specifically to generate complementary learners. Second, rather than giving each learner an equal vote, boosting gives each learner's vote a weight based on its past performance.

$$G(x) = \operatorname{sign}\left[\sum_{m=1}^{M} \alpha_m G_m(x)\right]$$

Since the models in the ensemble are built to be complementary, it is possible to increase ensemble performance to an arbitrary threshold simply by adding additional classifiers to the group. Given the obvious utility of this finding, boosting is thought to be one of the most significant discoveries in machine learning.

Although boosting can create a model that meets an arbitrarily low error rate, this may not always be reasonable in practice. For one, the performance gains are incrementally smaller as additional learners are added, making some thresholds practically infeasible.

5. User Manual

Running train.py file uses to mushroom.csv to get data and then splits it into 3 parts: Train, validation and test. Then program trains on the train data while validating results using validation data. Finally program predicts results for test data and prints accuracy.

6. Technical documentation

```
 # Implementation of the scikit-learn API for XGBoost classification.

xg_model = XGBClassifier(n_estimators=1000, learning_rate=0.05)


# Fit gradient boosting model

xg_model.fit(X_train, Y_train, eval_set=[(X_val, Y_val)], verbose=False)


# Predict with data

test_preds = xg_model.predict(X_test)


#round predictions

predictions = [round(value) for value in test_preds]


# evaluate predictions

accuracy = accuracy_score(Y_test, predictions)


#print accuracy

print("Accuracy: %.2f%%" % (accuracy * 100.0))
```

7. Description of tests

After prototoype version we made 3 changes according to our teachers feedback.

1) We changed the hardcoded boosting part to XGBoost library methods.
2) We used one-hot encoding instead of decimal encoding.

3) We changed the hardcoded accuracy measurement to using sklearn accuracy_score function.

After this changes our accuracy hit to 100%. In prototype version we were getting numbers between 97% and 99%.

8. Conclusions, comments

Decision Tree is a machine learning method that contains its knowledge in the form of logical structures (rules) that can be understood with no statistical knowledge. And also its good at classifying poisonous mushrooms.

9. List of references to literature, web pages

https://en.wikipedia.org/wiki/Gradient_boosting

https://xgboost.readthedocs.io/en/latest/

https://www.kaggle.com/ankitkuls/xgboost-with-one-hot-encoding

https://datatuts.com/gradient-boosting-in-python-from-scratch/

https://machinelearningmastery.com/evaluate-gradient-boosting-models-xgboost-python/

Lecture Notes (Lab3 - Decision Trees.pdf)

Name of members

Burak Can Buyukbas

Anil Pinar Ozdemir

Ozan Aybars