Data Science With Brain Tumor Survival Month Prediction

Zhaopu Teng

Aug 13, 2020.

1 Introduction

A research about Survival month with the Brain Tumor patient. I get the dataset from https://seer.cancer.gov/data.
I am using part of the data which is from year 2010 to 2016. It have 35406 tuples and 41 attributes.
The Class attribute is survival month. I choose 14 of the attributes to do the prediction. Which are:

1. Age at diagnosis (numeric)
2. Tumor size (was numeric, but I converge to nominal)
3. Year of diagnosis (nominal)
4. Primary Site - labeled (nominal)
5. Laterality (nominal)
6. Radiation recode (nominal)
7. Chemotherapy recode (yes, no/unk) (nominal)
8. Sequence number (nominal)
9. First malignant primary indicator (nominal)
10. SEER Combined Mets at DX-bone (2010+) (nominal)
11. SEER Combined Mets at DX-brain (2010+) (nominal)

12. SEER Combined Mets at DX-liver (2010+) (nominal)
13. SEER Combined Mets at DX-lung (2010+) (nominal)
14. Sex (nominal)

2 Preprocess

2.1 Preprocess for Class

The Class, 'survival month', was numeric data. For prediction, I converged it to two group. One is from 0 month to 18 month. The other one is over 18 month.
For further study on this topic, I may continue to converge it to three group, which will be more practical for advisement to doctor.

```
survival_month_normalize = []
if discretize_num == 2:
    for m in survival_month:
        temp = ''
        if m <= 18:
            temp = 0
        else:
            temp = 18
        survival_month_normalize.append(temp)
if discretize_num == 3:
    for m in survival_month:
        temp = ''
        if m <= 6:
            temp = 0
        elif 6 < m <= 18:
            temp = 6
        else:
            temp = 18
        survival_month_normalize.append(temp)
```

2.2 Preprocess for attributes

Tumor size was numric data type. I was converged it to nominal data type approximate twenty different category.
I also unified converged all attributes with OrdinalEncoder() and StandardScaler(), in order to fit all the algorithm I am going to test.

```
enc = preprocessing.OrdinalEncoder()
enc.fit(x)
x = enc.transform(x)
scaler = preprocessing.StandardScaler()
scaler.fit(x)
x = scaler.transform(x)

x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=.5, random_state=0)
return x_train, x_test, y_train, y_test
```

3 Algorithm Selected

I selected totally 13 different machine learning algorithm, which are:

1. Logistic
2. KNN
3. Ada+Logistic
4. GaussianNB
5. Lda
6. Qda
7. Decision Tree
8. Random Forest
9. SVM with Linear
10. SVM with RBF
11. SVM with Poly
12. Ada + SVM Linear
13. Ada + SVM with RBF

4 Summarized Result

The summarized table is shown as follow:

Algorithm	Accuracy	ROC Aera	Confusion Matrix		
Logistic	0.718	0.683	9035	1662	
			3272	3554	
KNN	0.729	0.692	9221	1476	n_neighbors=59
			3263	3563	
Ada+Log	0.717	0.678	9147	1550	n_estimators=20
			3412	3414	learning_rate=1
GaussianNB	0.567	0.619	4126	6571	
			1013	5813	
Lda	0.72	0.684	9090	1607	
			3295	3531	
Qda	0.646	0.675	5806	4891	
			1320	5506	
DecisionTre	0.667	0.65	7769	2928	
			2904	3922	
RandomFor	0.732	0.709	8686	2011	
			2687	4139	
SVM_linear	0.719	0.682	9105	1592	
			3322	3504	
SVM_RBF	0.745	0.709	9354	1343	
			3120	3706	
SVM_poly	0.734	0.7	9140	1557	degree=3
			3108	3718	
Ada+SVM_L	0.722	0.695	8727	1970	n_estimators=1
			2902	3924	learning_rate=.5
Ada+SVM_R	0.731	0.701	9131	1566	n_estimators=3
			3141	3685	learning_rate=.5

I also included ROC Area which is a very useful index to decide a good model or not. We can see that the SVM with RBF has the highest accuracy and also a good ROC Aera.

5 Working in Progress

I still try to work on this project after finish the class to see if I can find more interesting result. Here are some directions I can continue working on:

1. Try different attribute selected method.
2. Discretize the Class ('Survival month') to 3 or more interval.
3. Search better algorithms.
4. Balance the number of samples in the different class if necessary.