روش‌های انتخاب ویژگی

در بخش قبلی، چگونگی پیش پردازش و آماده سازی داده برای یادگیری ماشین را با جزییات بررسی کردیم. در این بخش، جزییات مربوط به گزینش ویژگی های داده (data feature selection) و جنبه های مختلف مربوط به آن را بررسی می کنیم.

اهمیت روش‌های انتخاب ویژگی

کارایی مدل یادگیری ماشین، ارتباط مستقیمی با ویژگی های داده مورد استفاده برای آموزش آن دارد. اگر ویژگی های داده فراهم شده برای مدل ML نا مربوط باشد، روی کارایی مدل تاثیر منفی خواهد داشت. از طرف دیگر، اگر ویژگی های داده استفاده شده مناسب باشد، دقت مدل یادگیری ماشین شما، به خصوص رگرسیون خطی و لوجستیک (linear and logistic regression)، می تواند افزایش یابد.

حال، سوالی که مطرح می شود این است که گزینش خودکار ویژگی (automatic feature selection) چیست؟ می توان آن را به عنوان روندی تعریف کرد که به کمک آن، ویژگی هایی از داده که بیشترین ارتباط را با خروجی دارند یا متغیر های پیش بینی که به آنها علاقه داریم را انتخاب می کنیم. همچنین به آن انتخاب خصیصه (attribute selection) نیز می گویند.

موارد زیر برخی از مزایای گزینش خودکار ویژگی قبل از مدلسازی داده است.

- اعمال انتخاب ویژگی قبل از مدلسازی داده، پوشش بیش از حد (overfitting) را کاهش می دهد.
- اعمال انتخاب ویژگی قبل از مدلسازی داده، دقت مدل یادگیری ماشین را افزایش می دهد.
- اعمال انتخاب ویژگی قبل از مدلسازی داده، زمان یادگیری و آموزش را کاهش می دهد.

روش های گزینش ویژگی:

موارد زیر روش های گزینش خودکار ویژگی است که می توانیم برای مدلسازی داده یادگیری ماشین در پایتون از آن استفاده کنیم.

انتخاب تک متغیره (univariate):

این روش گزینش ویژگی در انتخاب آن ویژگی هایی که با کمک تست آماری، قوی ترین رابطه را با متغیرهای پیش بینی دارند، بسیار مفید است. می توان با کمک کلاس SelectKBest0class از کتابخانه پایتون scikit-learn ، روش گزینش ویژگی تک متغیره را پیاده سازی کرد.

مثال: در این مثال، با استفاده از مجموعه داده Pima Indians Diabetes ، و با کمک تست آماری chi-square ، به انتخاب ۴ خصیصه که بهترین ویژگی ها را دارند، خواهیم پرداخت.

```
from pandas import read_csv
from numpy import set_printoptions
from sklearn.feature_selection import SelectKBest
from sklearn.feature_selection import chi2
path = r'C:\pima-indians-diabetes.csv'
names = ['preg', 'plas', 'pres', 'skin', 'test', 'mass', 'pedi', 'age',
'class']
dataframe = read_csv(path, names=names)
array = dataframe.values
```

سپس، آرایه را به دو مولفه ورودی و خروجی تفکیک می کنیم.

```
X = array[:,0:8]
Y = array[:,8]
```

خطوط زیر از کد، بهترین ویژگی های مجموعه داده را انتخاب می کند.

```
test = SelectKBest(score_func=chi2, k=4)
fit = test.fit(X,Y)
```

همچنین می توانیم مطابق خواسته خود، داده را برای خروجی خلاصه کنیم. در اینجا، دقت را روی ۲ تنظیم می کنیم و ۴ خصیصه داده با بهترین ویژگی ها و بهترین امتیاز هر ویژگی را نشان می دهیم.

```
set_printoptions(precision=2)
print(fit.scores_)
featured_data = fit.transform(X)
print ("\nFeatured data:\n", featured_data[0:4])
```

خروجی:

```
[ 111.52 1411.89 17.61 53.11 2175.57 127.67 5.39 181.3 ]
Featured data:
[[148.  0. 33.6 50. ]
[  85.  0. 26.6 31. ]
[ 183.  0. 23.3 32. ]
[  89. 94. 28.1 21. ]]
```

حذف بازگشتی ویژگی: همانطور که از نام آن بر می آید، روش انتخاب ویژگی RFE (حذف بازگشتی ویژگی - Recursive feature elimination) ، به صورت بازگشتی خصیصه ها را حذف می کند و مدل را با خصیصه های باقی مانده می سازد. با کمک کلاس *RFE* از کتابخانه پایتون *scikit-learn* ، می توان روش انتخاب ویژگی RFE را پیاده سازی کرد.

مثال: در این مثال، با استفاده از RFE و الگوریتم رگرسیون لجستیک (logistic regression algorithm)، به انتخاب ۳ خصیصه برتر که دارای بهترین ویژگی ها هستند، از مجموعه داده Pima Indians Diabetes می پردازیم.

```
from pandas import read_csv
from sklearn.feature_selection import RFE
from sklearn.linear_model import LogisticRegression
path = r'C:\pima-indians-diabetes.csv'
names = ['preg', 'plas', 'pres', 'skin', 'test', 'mass', 'pedi', 'age',
'class']
dataframe = read_csv(path, names=names)
array = dataframe.values
```

سپس، آرایه را به مولفه های ورودی و خروجی آن تفکیک می کنیم.

```
X = array[:,0:8]
Y = array[:,8]
```

کد زیر بهترین ویژگی ها را از یک مجموعه داده انتخاب می کند.

```
model = LogisticRegression()
rfe = RFE(model, 3)
fit = rfe.fit(X, Y)
print("Number of Features: %d")
print("Selected Features: %s")
print("Feature Ranking: %s")
```

خروجی:

```
Number of Features: 3
Selected Features: [ True False False False False True True False]
Feature Ranking: [1 2 3 5 6 1 1 4]
```

با توجه به خروجی بالا، می بینیم که RFE ، preg ، mass و pedi را به عنوان ۳ ویژگی برتر انتخاب کرده است. آنها در خروجی با ۱ مشخص شده اند.

تجزیه و تحلیل مولفه اصلی(PCA): PCA که عموما روش کاهش داده نیز نامیده می شود، از آنجایی که از جبر خطی برای تبدیل مجموعه داده به یک قالب فشرده استفاده می کند، یک روش انتخاب ویژگی بسیار مفید است. می توان با استفاده تر کلاس PCA از کتابخانه پایتون scikit-learn ، روش انتخاب ویژگی PCA را پیاده سازی کرد. می توانیم تعداد مولفه های اصلی در خروجی را انتخاب کنیم.

مثال: در این مثال، برای انتخاب ۳ عدد از بهترین مولفه های اصلی مجموعه داده Pima Indians Diabetes ، از PCA استفاده می کنیم.

```
from pandas import read_csv
from sklearn.decomposition import PCA
path = r'C:\pima-indians-diabetes.csv'
names = ['preg', 'plas', 'pres', 'skin', 'test', 'mass', 'pedi', 'age',
'class']
dataframe = read_csv(path, names=names)
array = dataframe.values
```

سپس، ارایه را به مولفه های ورودی و خروجی تفکیک می کنیم.

```
X = array[:,0:8]
Y = array[:,8]
```

کد زیر ویژگی ها را از مجموعه داده استخراج می کند.

```
pca = PCA(n_components = 3)
fit = pca.fit(X)
print("Explained Variance: %s") % fit.explained_variance_ratio_
print(fit.components_)
```

خروجی:

```
Explained Variance: [ 0.88854663 0.06159078 0.02579012]
[[ -2.02176587e-03 9.78115765e-02 1.60930503e-02 6.07566861e-02
9.93110844e-01 1.40108085e-02 5.37167919e-04 -3.56474430e-03]
[ 2.26488861e-02 9.72210040e-01 1.41909330e-01 -5.78614699e-02
-9.46266913e-02 4.69729766e-02 8.16804621e-04 1.40168181e-01]
[ -2.24649003e-02 1.43428710e-01 -9.22467192e-01 -3.07013055e-01
2.09773019e-02 -1.32444542e-01 -6.39983017e-04 -1.25454310e-01]]
```

با توجه به خروجی بالا می بینیم که ۳ مولفه اصلی شباهت کمی با داده source دارد.

اهمیت ویژگی:

همانطور که از نام آن مشخص است، از روش اهمیت ویژگی برای انتخاب ویژگی های اهمیت(importance) استفاده می شود. اساسا این روش برای انتخاب ویژگی ها، از یک طبقه بندی کننده(classifier) تحت نظارت آموزش دیده استفاده می کند. می توان با کمک کلاس ExtraTreeClassifier از کتابخانه پایتون scikit-learn ، این روش انتخاب ویژگی را پیاده سازی کرد.

مثال:در این مثال، برای انتخاب ویژگی از مجموعه داده Pima Indians Diabetes، از ExtraTreeClassifier استفاده خواهیم کرد.

```
from pandas import read_csv
from sklearn.ensemble import ExtraTreesClassifier
path = r'C:\Desktop\pima-indians-diabetes.csv'
names = ['preg', 'plas', 'pres', 'skin', 'test', 'mass', 'pedi', 'age',
'class']
dataframe = read_csv(data, names=names)
array = dataframe.values
```

سپس، آرایه را به مولفه های ورودی و خروجی تفکیک می کنیم.

```
X = array[:,0:8]
Y = array[:,8]
```

کد زیر ویژگی ها را از مجموعه داده استخراج می کند.

```
model = ExtraTreesClassifier()
model.fit(X, Y)
print(model.feature_importances_)
```

خروجی:

```
[ 0.11070069 0.2213717 0.08824115 0.08068703 0.07281761 0.14548537
0.12654214 0.15415431]
```

با توجه به خروجی، می بینیم که برای هر خصیصه امتیازی وجود دارد. هر چه امتیاز بالاتر باشد، اهمیت آن خصیصه بالاتر خواهد بود.