

Fintech_Report7

毛阿妍3170102656

2019/8/28

Fintech_Report7

1:实验目的

2:实验步骤

2.1:数据预处理与特征工程

2.2:划分数据集、训练集重采样

2.3:模型的训练

2.3.1:KN

2.3.2:SVM

2.3.3:LR

2.4:在交叉检验集上对分类器的效果进行评估

3:实验结果

3.1:对比不同的分类算法对结果的影响

3.2:对比金融营销场景与其他金融科技场景算法的不同

1:实验目的

本实验旨在对金融用户数据进行数据挖掘与分析，从中找出目标客户，并对结果进行评估分析。

具体实验内容为根据葡萄牙某银行机构获得的用户数据，结合相应的标签(标签内容 y 为向该用户进行的推销是否成功)进行数据挖掘，判断用户是否为潜在客户，并评估预测结果。

2:实验步骤

2.1:数据预处理与特征工程

缺失值处理、分类变量数值化、数据特征预处理

data_preproess()函数

```
x = pd.get_dummies(data)
# 1: 缺失值处理
# 不处理 / 丢弃 / 填充
# age-average,sex/marriage/study etc-0
# 没有缺失值，所以跳过这一步

# 2: 字符串数据处理:建立字符串索引
# one-hot编码、建立字符串索引 (转换为出现频率)
```

```
# 3: 特征二值化:通过设置阈值,把数值sum_ckcs的特征转换为布尔值即客户是否会订购存款。
#           将sum_ckcs大于0的值设为1表示该客户会订购, sum_ckcs等于0的值设为0表示该客户不会订购存款。

# 4: 数据归一化: min-max将所有数据缩放到0-1之间

# 通过删除均值和缩放到单位方差来标准化特征
scaler = StandardScaler()
x = scaler.fit_transform(x)
```

2.2:划分数数据集、训练集重采样

(已实现)split_data()函数

2.3:模型的训练

利用常用的分类模型(包括感知机, SVM, 朴素贝叶斯, 决策树, logistic回归, 随机森林等等), 在训练集上进行训练。

2.3.1:KN

```
def predictKN(x_train, x_test, y_train):

    # your code here begin
    # train your model on 'x_train' and 'x_test'
    # predict on 'y_train' and get 'y_pred'

    var = VarianceThreshold(threshold=1)
    x_train = var.fit_transform(x_train)
    x_test = var.transform(x_test)

    model = KNeighborsClassifier()
    model.fit(x_train, y_train)
    y_pred = model.predict(x_test)

    # your code here end

    return y_pred
```

2.3.2:SVM

```
def predictSVM(x_train, x_test, y_train):

    # your code here begin
    # train your model on 'x_train' and 'x_test'
    # predict on 'y_train' and get 'y_pred'
```

```

clf = SVC()
clf.fit(x_train, y_train)

SVC(C=1.0, cache_size=200, class_weight=None, coef0=0.0,
    decision_function_shape=None, degree=3, gamma='auto', kernel='rbf',
    max_iter=-1, probability=False, random_state=None, shrinking=True,
    tol=0.001, verbose=False)

y_pred = clf.predict(x_test)

# your code here end

return y_pred

```

2.3.3:LR

```

def predictLR(x_train, x_test, y_train):

    # your code here begin
    # train your model on 'x_train' and 'x_test'
    # predict on 'y_train' and get 'y_pred'

    model = LogisticRegression()
    model.fit(x_train, y_train)
    y_pred = model.predict(x_test)

    # your code here end

    return y_pred

```

2.4:在交叉检验集上对分类器的效果进行评估

1. 可以利用实验 3 中实现的实现 LR、SVM 和感知机(Perceptron)三种分类算法的其中一种。
2. 可以利用 sklearn 中自带的分类器进行分类，详细的算法参考 FinMKT.py 中的#some usable model
3. 鼓励对比不同的分类算分对结果的影响。

我在precision、recall的基础上加了一个F1_score对分类器的效果进行评估。

```
F1_score = 2*precision*recall / (precision+recall)
```

3:实验结果

3.1:对比不同的分类算法分对结果的影响

在这三个分类算法中，LogisticRegression()的效果最好，SVM次之，KN最差。

```
FinMKT x
/usr/local/bin/python3.7 /Users/may/Desktop/实验7/FinMKT.py

===KN===
model precision:0.55 recall:0.34
F1_score:0.42

===SVM===
model precision:0.65 recall:0.37
F1_score:0.47

===LR===
model precision:0.66 recall:0.42
F1_score:0.51
```

3.2:对比金融营销场景与其他金融科技场景算法的不同

往往都需要数据预处理（标准化），所用具体模型有区别；例如智能投顾会有专门的MV、rnr等组合投资模型，而金融营销常常用各个分类算法。总体而言都会用到人工智能、机器学习等技术。