Лабораторная работа 3. Байесовский классификатор

1. Цели

Приобрести навыки построения байесовского классификатора

2. Задачи

1. Построить модель гауссовского наивного байесовского классификатора для бинарной классификации
2. Оценить точность модели

3. Теоретические сведения

Методические указания для решения поставленного задания

3.1. Бинарная классификация

Наивный байесовский алгоритм – алгоритм классификации, основанный на теореме Байеса с допущением о независимости признаков. Другими словами, алгоритм предполагает, что наличие какого-либо признака в классе не связано с наличием какого-либо другого признака. Например, с помощью такой модели можно определить письмо, содержащее спам. В данной работе модель будет определять, поступит ли студент в магистратуру, на основе его дохода и размера апартаментов

В данной работе вы будете работать с гауссовским наивным байесовским классификатором. Модификация заключается в том, что классификатор предполагает гауссовское распределение объектов

Для того, чтобы рассчитать модель классификатора используются следующие формулы, где m – число объектов класса y

$$\mu_y = \frac{1}{m}\sum_{i=1}^{m} x_i$$

$$\Sigma_y = \frac{1}{m}\sum_{i=1}\left(x_i - \mu_y\right)\cdot\left(x_i - \mu_y\right)^T$$

$$p(x|y) = N\left(\mu_y, \Sigma_y\right) = \frac{1}{\sqrt{(2\pi)^D\left|\det\left(\Sigma_y\right)\right|}}\exp\left(-\frac{1}{2}\left(x-\mu_y\right)^T\Sigma_y^{-1}\left(x-\mu_y\right)\right)$$

Мультиколлинеарность – тесная корреляционная взаимосвязь между факторами. Мультиколлинеарность может стать причиной переобучения модели, что приведет к неверным результатам её работы. Кроме того, избыточные параметры увеличивают сложность модели, а значит и время ее тренировки. Также мультиколлинеарность факторов плоха тем, что модель будет содержать избыточные переменные, а это значит, что:

1. Осложняется интерпретация параметров как величин действия факторов, параметры теряют смысл и следует рассматривать другие переменные;
2. Оценки параметров ненадежны – получаются большие стандартные ошибки, которые меняются с изменением объема наблюдений, что делает модель непригодной для прогнозирования.

Для оценки мультиколлинеарности используется матрица парных коэффициентов корреляции

3.2. Построение модели с помощью `scikit-learn`

Пример построения модели будет приведён ниже, демонстрация будет проводиться на данных из набора данных Go to college dataset (рус. Набор данных о поступлении в колледж)

3.2.1. Подключение библиотек

Подключение библиотек

```
import pandas as pd
import seaborn as sns

from sklearn import metrics
from sklearn.model_selection import train_test_split
from sklearn.naive_bayes import GaussianNB
```

3.2.2. Работа с набором данных

Подключение набора данных

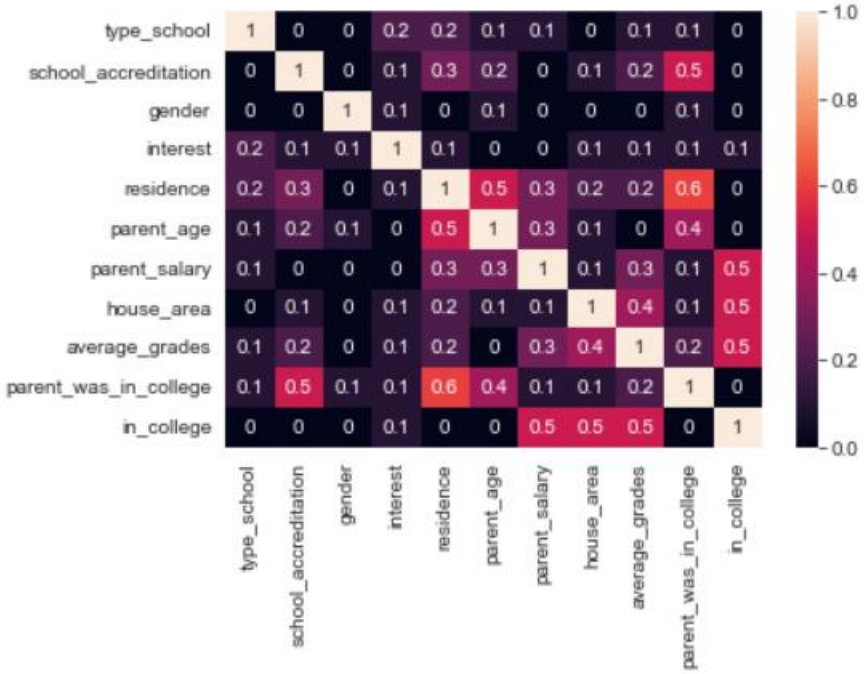
```
PATH = "college.csv"
dataset = pd.read_csv(PATH)
```

Факторизация значений в столбцах, содержащих строковые значения

```
for column in dataset:
    if type(dataset[column][0]) is str:
        dataset[column] = pd.factorize(dataset[column])[0]
```

Быстрая, визуальная проверка данных на мультиколлинеарность – не выявила особо сильных связей

```
sns.heatmap(
    round(
        abs(dataset.corr()),
        1,
    ),
    annot=True,
)
```



3.2.3. Построение модели

Разделение выборки на тренировочную и тестовую часть, а также на параметры и эталоны

```
train_input, test_input, train_output, test_output = train_test_split(
    dataset.drop('in_college', axis=1),
    dataset["in_college"],
    test_size=0.2
)
```

Обучение, тестирование и проверка модели

```
model = GaussianNB()
model.fit(train_input, train_output)

predictions = model.predict(test_input)
accuracy = metrics.accuracy_score(predictions, test_output)

print(f"Точность модели на тестовом участке = {accuracy}")
```

Точность модели на тестовой части набора данных составила `0.735`

4. Задание

Выбрать с сайта kaggle.com набор данных в формате `.csv`, пригодный для построения бинарной классификации, загрузить и подготовить его к дальнейшей обработке. Наборы данных не должны повторяться внутри группы. Задание индивидуальное. Требования:

1. Построить модель гауссовского наивного байесовского классификатора
2. Оценить точность модели
3. Указать какие знания можно получить из набора
4. Сохранить `IPython Notebook`

4.1. Продвинутое задание

Построить вторую модель, без использования средств `scikit-learn`