Московский государственный технический университет им. Н.Э. Баумана
Кафедра «Системы обработки информации и управления»

Лабораторная работа
по дисциплине
«Методы машинного обучения»
на тему
«Технологии разведочного анализа и обработки данных.»

Выполнил:
студент группы ИУ5-64Б
Береговая Д.

_____

Москва — 2019 г.

1. Задание

1.1. Задача №1.

Для заданного набора данных проведите корреляционный анализ. В случае наличия пропусков в данных удалите строки или колонки, содержащие пропуски. Сделайте выводы о возможности построения моделей машинного обучения и о возможном вкладе признаков в модель.

1.1.1. Ход работы

К сожалению я не разобрался с датасетом 3 варианта, потому что не нашел заголовки атрибутов, а там все на английском, и это не такая простая задача. Поэтому в данном задании будет использован датасет 6 варианта Admission_Predict.csv

1.1.2. Импортируем библиотеки

[10]:
```
import os
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.preprocessing import MinMaxScaler
from sklearn.linear_model import LinearRegression, LogisticRegression
from sklearn.model_selection import train_test_split
from sklearn.neighbors import KNeighborsRegressor, KNeighborsClassifier
from sklearn.metrics import accuracy_score, balanced_accuracy_score
from sklearn.metrics import precision_score, recall_score, f1_score,␣
 ↪classification_report
from sklearn.metrics import confusion_matrix
from sklearn.metrics import plot_confusion_matrix
from sklearn.model_selection import GridSearchCV
from sklearn.metrics import mean_absolute_error, mean_squared_error,␣
 ↪mean_squared_log_error, median_absolute_error, r2_score
from sklearn.metrics import roc_curve, roc_auc_score
from sklearn.svm import SVC, NuSVC, LinearSVC, OneClassSVM, SVR, NuSVR,␣
 ↪LinearSVR
from sklearn.tree import DecisionTreeClassifier, DecisionTreeRegressor,␣
 ↪export_graphviz
from sklearn.ensemble import RandomForestClassifier, RandomForestRegressor
from sklearn.ensemble import ExtraTreesClassifier, ExtraTreesRegressor
from sklearn.ensemble import GradientBoostingClassifier,␣
 ↪GradientBoostingRegressor
from sklearn.datasets import load_boston
from gmdhpy import gmdh
%matplotlib inline
sns.set(style="ticks")
```

1.1.3. Зададим выборку

[11]:
```
from sklearn.datasets import load_boston
X, y = load_boston(return_X_y=True)
print(X.shape)
```

```
(506, 13)
```

1.1.4. Запишем выборку в файл

[42]:
```
data = load_boston()

data = pd.DataFrame(data=data['data'], columns = data['feature_names'])
```

1.1.5. Проверим правильность создания выборки

[43]:
```
data.head()
```

[43]:

	CRIM	ZN	INDUS	CHAS	NOX	RM	AGE	DIS	RAD	TAX	PTRATIO	B	LSTAT
0	0.00632	18.0	2.31	0.0	0.538	6.575	65.2	4.0900	1.0	296.0	15.3	396.90	4.98
1	0.02731	0.0	7.07	0.0	0.469	6.421	78.9	4.9671	2.0	242.0	17.8	396.90	9.14
2	0.02729	0.0	7.07	0.0	0.469	7.185	61.1	4.9671	2.0	242.0	17.8	392.83	4.03
3	0.03237	0.0	2.18	0.0	0.458	6.998	45.8	6.0622	3.0	222.0	18.7	394.63	2.94
4	0.06905	0.0	2.18	0.0	0.458	7.147	54.2	6.0622	3.0	222.0	18.7	396.90	5.33

1.1.6. Проверим типы данных

[44]:
```
data.dtypes
```

[44]:
```
CRIM       float64
ZN         float64
INDUS      float64
CHAS       float64
NOX        float64
RM         float64
AGE        float64
DIS        float64
RAD        float64
TAX        float64
PTRATIO    float64
B          float64
LSTAT      float64
```

```
dtype: object
```

1.1.7. Проверяем датасет на наличие пустых значений

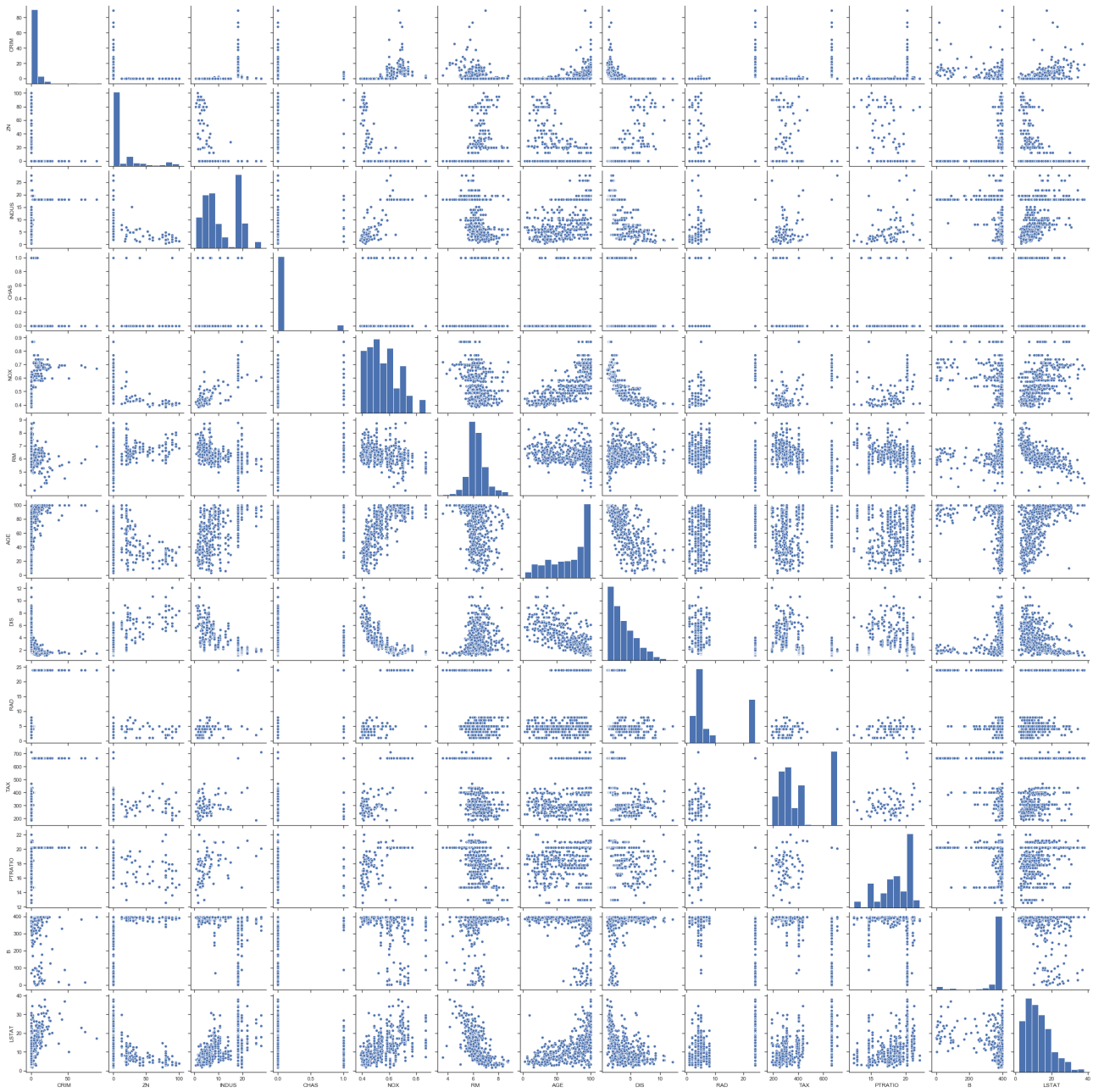
[45]:
```
data.isnull().sum()
```

[45]:
```
CRIM       0
ZN         0
INDUS      0
CHAS       0
NOX        0
RM         0
AGE        0
DIS        0
RAD        0
TAX        0
PTRATIO    0
B          0
LSTAT      0
dtype: int64
```

1.1.8. Построим парную диаграмму для наглядности структуры наших данных

[46]:
```
sns.pairplot(data)
```

[46]:
```
<seaborn.axisgrid.PairGrid at 0x1a2a413590>
```



1.1.9. Построим корреляционную матрицу

[47]:
```
fig, ax = plt.subplots(figsize=(15,10))
sns.heatmap(data.corr(), annot=True, fmt='.2f', cmap='GnBu')
```

[47]: <matplotlib.axes._subplots.AxesSubplot at 0x1a2eedae10>

2. Вывод

1. В матрице признаки хорошо коррелируют между собой. Это значит, что на их основании можно будет построить обучающую модель.
2. Если за целевой признак взять LSTAT, то модель будем строить по признакам AGE, NOX, TAX, CRIM.