Московский авиационный институт
(национальный исследовательский университет)

Факультет: «Компьютерные науки и прикладная математика»
Кафедра: 806 «Вычислительная математика и программирование»

Лабораторная работа №0

по курсу «Машинное обучение»
Тема: «Анализ данных»

Студент: Мариничев И. А.
Группа: М8О-308Б-19
Преподаватель: Ахмед С. Х.
Оценка:

Москва
2022

1. Постановка задачи.

В данной лабораторной работе вы выступаете в роли предприимчивого начинающего стартапера в области машинного обучения. Вы заинтересовались этим направлением и хотите предложить миру что-то новое и при этом неплохо заработать. От вас требуется определить задачу, которую вы хотите решить и найти под нее соответствующие данные. Так как вы не очень богаты, вам предстоит руками проанализировать данные, визуализировать зависимости, построить новые признаки и сказать хватит ли вам этих данных, и если не хватит найти еще. Вы готовитесь представить отчет ваши партнерам и спонсорам, от которых зависит дальнейшая ваша судьба. Поэтому тщательно работайте:) И главное, день промедления и вас опередит ваш конкурент, да и сплагиаченная работа отразится на репутации.

По сути, в данной лабораторной работе вы выполняете часть работы BI системы. Если вы заинтересовались этим направлением, то можно будет в дальнейшем что-то придумать)

2. Описание данных.

Будем работать с данными **covtype.data** о типе лесного покрытия из репозитория UCI. Доступно 7 различных классов:

Каждый объект описывается 54 признаками. Более подробно они описаны в файле **covtype.info**.

Соответственно, поставим задачу классификации - определить тип лесного покрова.

3. Порядок действий

1. Скачать данные и привести их к типу .csv (comma-separated values)

2. Добавить названия столбцов в соответствии с `covtype.info`

3. Разделить набор данных на тестовый, тренировочный и валидационный

4. Провести анализ на чистоту данных и отсутствие бесполезных значений

5. Разделить признаки на числовые и категориальные, чтобы работать с ними отдельно

6. Произвести анализ зависимостей для числовых признаков:

   - графики распределений,
   - двойные графики
   - матрица корреляций

7. Произвести графический анализ зависимостей для категориальных признаков

8. Добавить при необходимости или удалить признаки, исходя из проведенного анализа

9. Провести тестовое обучение модели и посмотреть на точность

3. Анализ данных.

Для начала импортируем все нужные библиотеки, которые необходимы для работы с данными.

```
import pandas as pd
import numpy as np
from matplotlib import pyplot as plt
import seaborn as sns

from sklearn.preprocessing import StandardScaler
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score
```

Начнем с самой важной части - посмотрим на данные. Приведем названия колонок в презентабельный вид в соответсвии с информационным файлом, загрузим их и посмотрим на небольшую часть.И заодно разделим признаки на числовые и категориальные, а также выделим целевой признак.

Разделение на train/validation/test

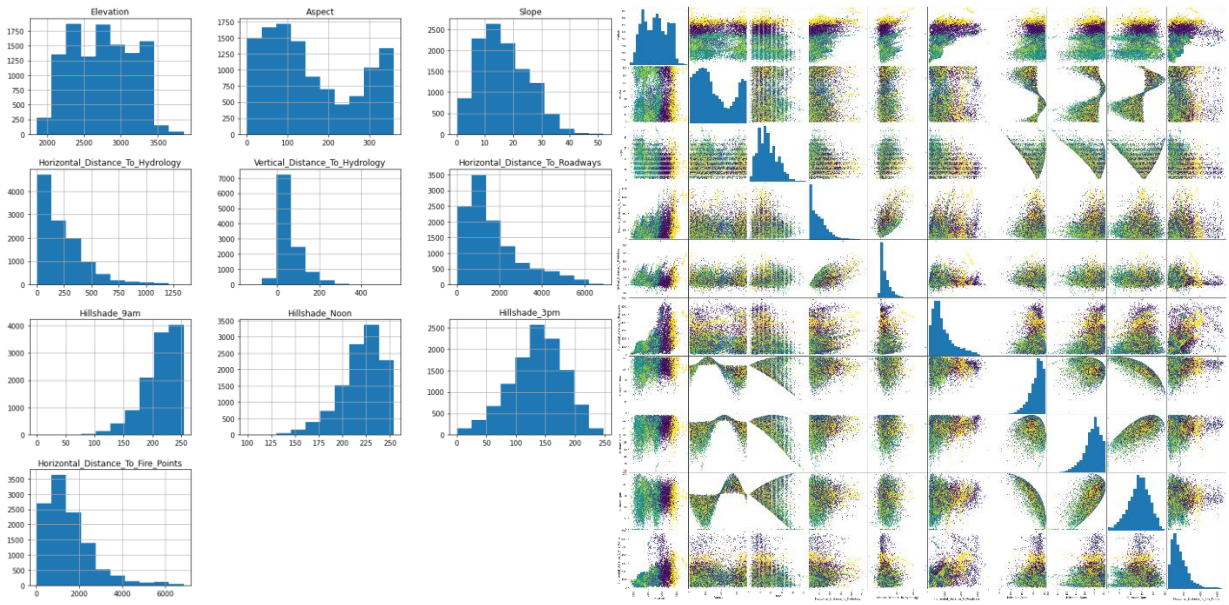
Заранее разделим наши данные на тестовую, валидационную и тренировочную выборки, чтобы не допустить утечек. В информационном файле сказано следующее:

```
-- first 11,340 records used for training data subset
-- next 3,780 records used for validation data subset
-- last 565,892 records used for testing data subset
```

```
X_train = covtype_data[feature_cols][0:11340]
y_train = covtype_data[target_col][0:11340]

X_valid = covtype_data[feature_cols][11340:15120]
y_valid = covtype_data[target_col][11340:15120]

X_test = covtype_data[feature_cols][15120:581012]
y_test = covtype_data[target_col][15120:581012]

print(X_train.shape, y_train.shape)
print(X_valid.shape, y_valid.shape)
print(X_test.shape, y_test.shape)
```

```
(11340, 54) (11340,)
(3780, 54) (3780,)
(565892, 54) (565892,)
```

Посмотрим на распределение числовых признаков и на двойные графики



Распределения:

- Hillshade_9am и Hillshade_Noon имеют бимодальное и левостороннее распределение.
- Horizontal_Distance_To_Firepoints, Horizontal_Distance_To_Roadways, Hori zontal_Distance_To_Hydrology имеют бимодальное и правостороннее распределение.
- Elevation похоже на равномерное распределение..
- Slope, Vertical_Distance_To_Hydrology, Hillshade_3pm показывают симметричное и бимодальное распределение.

Зависимости между признаками:

- Elevation имеет положительную динамику с:
  - Vertical_Distance_To_Hydrology
  - Horizontal_Distance_To_Roadways
  - Horizontal_Distance_To_Firepoints
  - Horizontal_Distance_To_Hydrology
- При увеличени Aspect Hillshade_Noon и Hillshade_3pm возрастают.
- Slope имеет отрицательную динамику с:
  - Elevation
  - Horizontal_Distance_To_Roadways

    - Hillshade_9am, Hillshade_Noon и Hillshade_3pm
    - Horizontal_Distance_To_Firepoints
- Horizontal_Distance_To_Hydrology имеет положительную динамику с:
    - Horizontal_Distance_To_Firepoints
    - Horizontal_Distance_To_Roadways
    - Vertical_Distance_To_Hydrology
- Vertical_Distance_To_Hydrology - Slope and Vertical_Distance_To_Hydrology - Horizontal_Distance_To_Hydrology коллинеарны.
- As Horizontal_Distance_To_Roadways increases, Horizontal_Distance_To_Fi repoints возрастает Slope падает.
- Hillshade_9am shows имеет отрицательную динамику с Hillshade_3pm и Aspect, при увеличении Hillshade_9am возрастает Elevation.
- Hillshade_Noon имеет положительную динамику с:
    - Elevation
    - Aspect
    - Horizontal_Distance_To_Roadways
    - Hillshade_3pm
    - Horizontal_Distance_To_Firepoints
- Hillshade_3pm имеет отрицательную динамику с Hillshade_9am и имеет положительную динамику с Hillshade_Noon.

Коллинеарные признаки:

- hillshade noon - hillshade 3 pm
- hillsahde 3 pm - hillshade 9 am
- vertical distance to hydrology - horizontal distance to hydrology
- elevation - slope

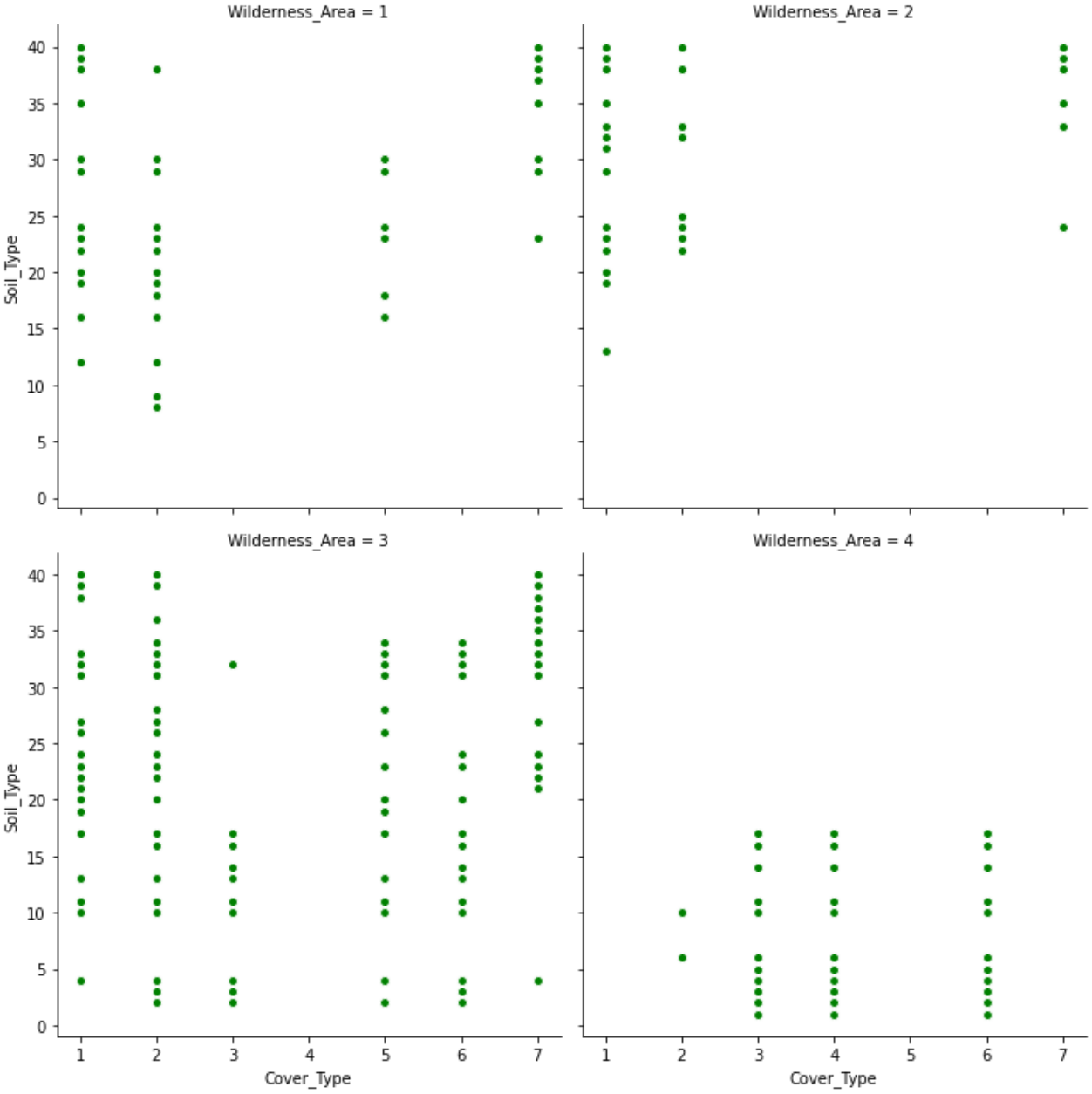
Коэффициент Пирсона

	Elevation	Aspect	Slope	Horizontal_Distance_To_Hydrology	Vertical_Distance_To_Hydrology	Horizontal_Distance_To_Roadways	Hillshade_9am
Elevation	1.00	-0.01	-0.31	0.41	0.12	0.58	0.10
Aspect	-0.01	1.00	0.03	0.04	0.06	0.06	-0.59
Slope	-0.31	0.03	1.00	-0.06	0.27	-0.28	-0.20
Horizontal_Distance_To_Hydrology	0.41	0.04	-0.06	1.00	0.65	0.20	-0.03
Vertical_Distance_To_Hydrology	0.12	0.06	0.27	0.65	1.00	0.01	-0.10
Horizontal_Distance_To_Roadways	0.58	0.06	-0.28	0.20	0.01	1.00	-0.00
Hillshade_9am	0.10	-0.59	-0.20	-0.03	-0.10	-0.00	1.00
Hillshade_Noon	0.22	0.33	-0.61	0.08	-0.13	0.24	-0.01
Hillshade_3pm	0.09	0.64	-0.33	0.08	-0.04	0.18	-0.78
Horizontal_Distance_To_Fire_Points	0.44	-0.05	-0.24	0.16	-0.02	0.49	0.08
Wilderness_Area1	0.33	-0.13	-0.15	-0.01	-0.12	0.37	0.17
Wilderness_Area2	0.26	0.03	-0.07	0.09	0.01	-0.08	-0.01
Wilderness_Area3	0.36	0.03	-0.11	0.20	0.07	0.12	-0.01
Wilderness_Area4	-0.78	0.07	0.29	-0.24	0.03	-0.44	-0.14
Cover_Type	0.01	-0.00	0.09	-0.02	0.07	-0.10	-0.01
Soil_Type	0.83	-0.01	-0.24	0.28	0.05	0.46	0.03

Hillshade_Noon	Hillshade_3pm	Horizontal_Distance_To_Fire_Points	Wilderness_Area1	Wilderness_Area2	Wilderness_Area3	Wilderness_Area4	Cover_Type	Soil_Type
0.22	0.09	0.44	0.33	0.26	0.36	-0.78	0.01	0.83
0.33	0.64	-0.05	-0.13	0.03	0.03	0.07	-0.00	-0.01
-0.61	-0.33	-0.24	-0.15	-0.07	-0.11	0.29	0.09	-0.24
0.08	0.08	0.16	-0.01	0.09	0.20	-0.24	-0.02	0.28
-0.13	-0.04	-0.02	-0.12	0.01	0.07	0.03	0.07	0.05
0.24	0.18	0.49	0.37	-0.08	0.12	-0.44	-0.10	0.46
-0.01	-0.78	0.08	0.17	-0.01	-0.01	-0.14	-0.01	0.03
1.00	0.61	0.12	-0.02	0.04	0.19	-0.20	-0.10	0.06
0.61	1.00	0.04	-0.12	0.04	0.13	-0.05	-0.06	0.06
0.12	0.04	1.00	0.44	0.04	0.00	-0.42	-0.09	0.36
-0.02	-0.12	0.44	1.00	-0.10	-0.47	-0.37	-0.23	0.38
0.04	0.04	0.04	-0.10	1.00	-0.16	-0.13	0.01	0.20
0.19	0.13	0.00	-0.47	-0.16	1.00	-0.57	0.12	0.19
-0.20	-0.05	-0.42	-0.37	-0.13	-0.57	1.00	0.08	-0.63
-0.10	-0.06	-0.09	-0.23	0.01	0.12	0.08	1.00	0.08
0.06	0.06	0.36	0.38	0.20	0.19	-0.63	0.08	1.00

Можно заметить, что между всеми признаками Hillshade есть коллинеарность. Кроме того, Hillshade_9am имеет самый низкий коэффициент Пирсона, равный *-0.01*, поэтому этот признак мы в дальнейшем исключим.

К сожалению, вообще все признаки имеют довольно низкий коэффициент Пирсона для целевого столбца. Посмотрим теперь на то, как связаны категориа льные признаки



- Зона 3 содержит разнообразные типы почвы и покрытий
- Зона 4 содержит только типы почвы с 1 по 20
- Покров 7 растет на типе почвы от 25 до 40
- Покровы 5 и 6 может расти на большинстве типов почвы
- Покров 3 часто растет на типе почвы от 0 до 15
- Покровы 1 и 2 может расти на любом типе почвы

Исходя из этого мы добавим несколько признаков, связывающих Soil_Type и Wilderness_Area

Создание новых признаков

Для данного датасета можно применить некоторую обработку данных о растояниях и добавить в виде новых признаков их вариации, такие как евклидово расстояние для Horizontal_Distance_To_Hydrology и Vertical_Distance_To_Hydrology и среднее для всех остальных растояний.

Для того, чтобы повысить корреляцию вычислим квадратный корень для всех признаков, помотрим на коэффициент Пирсона и там где он выше исходного, проведем замену признаков

Теперь соберем все наши наработки в одну единственную функцию преобразования и применим ее к тренирочному, валидационному и тестовому наборам признаков

```
def transform(data):
    # добавим числовой столбец типа почвы вместо one-hot-encoded
    make_soil_type_num(data)

    # добавим линейные комбинации столбцов расстояний
    data['Euclidian_Distance_To_Hydrology'] =
(data['Horizontal_Distance_To_Hydrology']**2 +
data['Vertical_Distance_To_Hydrology']**2)**0.5
    data['Mean_Elevation_Vertical_Distance_Hydrology'] = (data['Elevation'] +
data['Vertical_Distance_To_Hydrology'])/2
    data['Mean_Distance_Hydrology_Firepoints'] =
(data['Horizontal_Distance_To_Hydrology'] +
data['Horizontal_Distance_To_Fire_Points'])/2
    data['Mean_Distance_Hydrology_Roadways'] =
(data['Horizontal_Distance_To_Hydrology'] +
data['Horizontal_Distance_To_Roadways'])/2
    data['Mean_Distance_Firepoints_Roadways'] =
(data['Horizontal_Distance_To_Fire_Points'] +
data['Horizontal_Distance_To_Roadways'])/2

    # добавим произведения категориальных признаков
    data['WA1_To_ST'] = (data['Wilderness_Area1'] * data['Soil_Type'])
    data['WA2_To_ST'] = (data['Wilderness_Area2'] * data['Soil_Type'])
    data['WA3_To_ST'] = (data['Wilderness_Area3'] * data['Soil_Type'])
    data['WA4_To_ST'] = (data['Wilderness_Area4'] * data['Soil_Type'])

    # заменим столбцы для которых корень показал лучшее значение
    data['sqrt' + 'Horizontal_Distance_To_Hydrology'] =
np.sqrt(data['Horizontal_Distance_To_Hydrology'])
    data['sqrt' + 'Mean_Distance_Hydrology_Roadways'] =
np.sqrt(data['Mean_Distance_Hydrology_Roadways'])
    data['sqrt' + 'Euclidian_Distance_To_Hydrology'] =
np.sqrt(data['Euclidian_Distance_To_Hydrology'])

    # окончательный список признаков
    wilderness_areas = ['Wilderness_Area1', 'Wilderness_Area2',
'Wilderness_Area3', 'Wilderness_Area4']
    transformed_features = ['WA1_To_ST', 'WA2_To_ST', 'WA3_To_ST', 'WA4_To_ST',
'sqrtHorizontal_Distance_To_Hydrology', 'sqrtMean_Distance_Hydrology_Roadways',
'sqrtEuclidian_Distance_To_Hydrology',
                           'Mean_Elevation_Vertical_Distance_Hydrology',
'Mean_Distance_Firepoints_Roadways', 'Mean_Distance_Hydrology_Firepoints']

    all_features = (['Elevation', 'Aspect', 'Slope',
'Vertical_Distance_To_Hydrology', 'Horizontal_Distance_To_Roadways',
```

```
                    'Hillshade_Noon', 'Hillshade_3pm',
'Horizontal_Distance_To_Fire_Points' ] + wilderness_areas +
                    ['Soil_Type'] + transformed_features)
    data = data[all_features]

transform(X_train)
transform(X_valid)
transform(X_test)
```

Масштабирование признаков

Необходимо привести все признаки к одному масштабу. Для этого в sklearn есть StandardScaler.

StandardScaler для каждого признака x_i считает среднее μ_i и стандартное отклонение σ_i на обучающем датасете и обновляет признаки следующим образом:

$$x_i^{new} = \frac{x_i - \mu_i}{\sigma_i}$$

```
scaler = StandardScaler()
X_train = scaler.fit_transform(X_train, y_train)
X_valid = scaler.fit_transform(X_valid, y_valid)
X_test = scaler.transform(X_test)
```

Анализ завершен, наши данные готовы быть переданы на обучение.

Обучение

Используем модель RandomForestClassifier на наших подготовленных данных и посмортим на результаты

```
model_rf = RandomForestClassifier(n_estimators = 50, oob_score = True, n_jobs =
-1)
model_rf.fit(X_train, y_train)

y_train_pred = model_rf.predict(X_train)
y_valid_pred = model_rf.predict(X_valid)
y_test_pred = model_rf.predict(X_test)

oob_score = model_rf.oob_score_

train_accuracy = accuracy_score(y_train, y_train_pred)
valid_accuracy = accuracy_score(y_valid, y_valid_pred)
test_accuracy = accuracy_score(y_test, y_test_pred)

print(f'Точность на тренировочном наборе данных: {train_accuracy:.3f}')
print(f'Точность на валидационном наборе данных: {valid_accuracy:.3f}')
print(f'Точность на тестовом наборе данных: {test_accuracy:.3f}')
print(f'Out-of-Bag оценка: {oob_score:.3f}')

Точность на тренировочном наборе данных: 1.000
Точность на валидационном наборе данных: 0.865
Точность на тестовом наборе данных: 0.751
Out-of-Bag оценка: 0.861
```

4. Выводы.

В ходе данной лабораторной работы я научился анализировать наборы данных, создавать признаки на основе найденных закономерностей и зависимостей.

Набор данных о типе лесных покровов предоставляет довольно интересные возможности для анализа. К сожалению, почти все признаки в отдельности имеют довольно низкую корреляцию с типом лесного покрова, что не позволяет достичь максимальной точности. Кроме того, некоторые признаки оказались коллинеарными и их пришлось исключить. После пробного обучения модели *RandomForestClassifier* была достигнута точность 0.75 на тестовых данных.