



МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РФ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО
ОБРАЗОВАНИЯ «ЛИПЕЦКИЙ ГОСУДАРСТВЕННЫЙ
ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

Институт
Кафедра

компьютерных наук
автоматизированных систем управления

Лабораторная работа №1
по дисциплине «Машинное обучение»

Студент М-РИТ-25-1

(подпись, дата)

Станиславчук С. М.

Руководитель
Доцент, д.т.н.

(подпись, дата)

Сараев П. В.

Липецк 2025

1. Задание кафедры

- 1) Загрузите файл с данными airquality.
- 2) Выполните задания 2 — 6 из методич. указаний № 571 для л.р.1

2. Цель работы

Изучение основ работы с языком R / Python и базовых методов работы с данными.

3. Ход работы

В качестве языка программирования для выполнения лабораторной работы, я выбрал Python. Он обладает удобной библиотекой pandas для работы с табличными данными. Чтение csv файла осуществляется с помощью метода `read_csv()`; чтение строк без пропусков `dropna()`; найти минимальный / максимальный / рассчитать средний элементы: `min()`, `max()`, `mean()`.

Пример выполнения программы 1:

```
Число строк в таблице: 153
Число столбцов в таблице: 7
Число строк без пропусков вообще: 111
Число строк с пропусками одновременно по Ozone и Solar.R: 2
```

Статистика по столбцам:

Столбец: Ozone
Минимальное значение: 1.00
Максимальное значение: 168.00
Диапазон варьирования: 167.00
Среднее значение: 42.13
Количество пропусков: 37

Столбец: Solar.R
Минимальное значение: 7.00
Максимальное значение: 334.00
Диапазон варьирования: 327.00
Среднее значение: 185.93
Количество пропусков: 7

Столбец: Wind
Минимальное значение: 1.70
Максимальное значение: 20.70
Диапазон варьирования: 19.00
Среднее значение: 9.96
Количество пропусков: 0

Столбец: Temp
Минимальное значение: 56.00
Максимальное значение: 97.00

Диапазон варьирования: 41.00

Среднее значение: 77.88

Количество пропусков: 0

Среднее значение по столбцу Solar.R для 5-го месяца: 181.30

Дополнительная информация:

Всего строк с пропусками в любом столбце: 42

Процент строк без пропусков: 72.55%

3. Вывод

В результате выполненной лабораторной работы, изучил основы работы с табличными данными в языке программирования Python.

Приложение 1

```
# TASK (1=R-oriented) 2
# Посчитайте:
#   * Число строк в таблице
#   * Число столбцов в таблице
#   * Число строк, без NA
#   * Число строк, имеющих пропуски одновременно по столбцам Ozone и Solar.R
#   * Диапазоны варьирования (мин. и макс. значения), а также средние значения по столбцам
#     Ozone, Solar.R, Wind и Temp (без NA)
#   * Среднее значение по столбцу Solar.R для 5-го месяца (без NA)
import pandas as pd

df = pd.read_csv('dataset.csv')

ozone_name = 'Ozone'
solar_name = 'Solar.R'
wind_name = 'Wind'
temp_name = 'Temp'
month_name = 'Month'
day_name = 'Day'

print(f"Число строк в таблице: {df.shape[0]}")
print(f"Число столбцов в таблице: {df.shape[1]}")

no_missing_anywhere = df.dropna().shape[0]
print(f"Число строк без пропусков вообще: {no_missing_anywhere}")

o3_rh_missing = df[df[ozone_name].isnull() & df[solar_name].isnull()].shape[0]
print(f"Число строк с пропусками одновременно по {ozone_name} и {solar_name}: {o3_rh_missing}")

columns_to_analyze = [ozone_name, solar_name, wind_name, temp_name]

print("\nСтатистика по столбцам:")
print("-" * 50)

for col in columns_to_analyze:
    if col in df.columns:
        print(f"\nСтолбец: {col}")
        print(f"  Минимальное значение: {df[col].min():.2f}")
        print(f"  Максимальное значение: {df[col].max():.2f}")
        print(f"  Диапазон варьирования: {df[col].max() - df[col].min():.2f}")
        print(f"  Среднее значение: {df[col].mean():.2f}")
        print(f"  Количество пропусков: {df[col].isnull().sum()}")
    else:
        print(f"\nСтолбец {col} не найден в таблице")

solar_mean_5th_month = df.loc[df[month_name] == 5, solar_name].mean()

print(f"\nСреднее значение по столбцу {solar_name} для 5-го месяца: {solar_mean_5th_month:.2f}")

print("\nДополнительная информация:")
print(f"Всего строк с пропусками в любом столбце: {df.isnull().any(axis=1).sum()}")
print(f"Процент строк без пропусков: {no_missing_anywhere/df.shape[0]*100:.2f}%")
```

Приложение 2

```
# TASK 4
# Найдите и выведите на экран средние значения по столбцу Solar.R для каждого месяца

import pandas as pd

df = pd.read_csv('dataset.csv')

ozone_name = 'Ozone'
solar_name = 'Solar.R'
wind_name = 'Wind'
temp_name = 'Temp'
month_name = 'Month'
day_name = 'Day'

monthly_solar_mean = df.groupby(month_name)[solar_name].mean()

print("Средние значения по столбцу {solar_name} для каждого месяца:")
print("-" * 45)

for month, mean_value in monthly_solar_mean.items():
    print(f"Месяц {month}: {mean_value:.2f}")

print(f"\nОбщая статистика:")
print(f"Среднее {solar_name} за все месяцы: {df[solar_name].mean():.2f}")
print(f"Месяц с максимальным {solar_name}: {monthly_solar_mean.idxmax()}"
      f"\n{{monthly_solar_mean.max():.2f}}")
print(f"Месяц с минимальным {solar_name}: {monthly_solar_mean.idxmin()}"
      f"\n{{monthly_solar_mean.min():.2f}}")
```

Приложение 3

```
# TASK 5
# maxTemp.py
# Реализуйте функцию maxTemp(days = 1), которая возвращает требуемое количество пар месяц/день
# с максимальной температурой. Например, если параметр days == 3, то должны быть выведены ровно
# 3 пары значений "месяц/день". Параметр days - кол-во дней - некоторое натуральное число.

import pandas as pd

ozone_name = 'Ozone'
solar_name = 'Solar.R'
wind_name = 'Wind'
temp_name = 'Temp'
month_name = 'Month'
day_name = 'Day'

def maxTemp(n=5):
    """
    Возвращает n пар (месяц, день) с максимальной температурой

    Parameters:
    n (int): количество возвращаемых пар (по умолчанию 5)

    Returns:
    list: список кортежей в формате [(месяц, день, температура), ...]
    """
    try:
        df = pd.read_csv('dataset.csv')

        if month_name not in df.columns or temp_name not in df.columns:
            raise ValueError("В dataset.csv отсутствуют необходимые столбцы {month_name} или {temp_name}")

        daily_max_temp = df.groupby([month_name, day_name])[temp_name].max().reset_index()

    except Exception as e:
        print(f"Ошибка при чтении файла: {e}")
```

```

    daily_max_temp_sorted = daily_max_temp.sort_values(temp_name, ascending=False)

    top_n_days = daily_max_temp_sorted.head(n)

    result = []
    for _, row in top_n_days.iterrows():
        result.append((int(row[month_name]), int(row[day_name]), float(row[temp_name])))

    return result

except FileNotFoundError:
    raise FileNotFoundError("Файл dataset.csv не найден. Убедитесь, что он находится в
текущей директории.")
except Exception as e:
    raise Exception(f"Ошибка при выполнении функции: {str(e)}")

def print_max_temp_results(results):
"""
    Красиво выводит результаты функции maxTemp
"""

    month_names = {
        1: 'Январь', 2: 'Февраль', 3: 'Март', 4: 'Апрель',
        5: 'Май', 6: 'Июнь', 7: 'Июль', 8: 'Август',
        9: 'Сентябрь', 10: 'Октябрь', 11: 'Ноябрь', 12: 'Декабрь'
    }

    print("Дни с максимальной температурой:")
    print("-" * 40)
    for i, (month, day, temp) in enumerate(results, 1):
        month_name = month_names.get(month, f'Месяц {month}')
        print(f'{i:2}. {month_name:10} {day:2} число: {temp:.1f}°C")

if __name__ == "__main__":
    try:
        results = maxTemp(5)
        print_max_temp_results(results)

    except Exception as e:
        print(f"Ошибка: {e}")

```

Приложение 4

```

# TASK 3
# meanAir.py
# Напишите функцию, которая рассчитывает среднее значение по одному из столбцов factor (Ozone,
# Solar.R, Wind), когда Temp принимает значения от tMin до tMax включительно. Значения по-
умолчанию
# tMin = 60, tMax = 80

import pandas as pd

ozone_name = 'Ozone'
solar_name = 'Solar.R'
wind_name = 'Wind'
temp_name = 'Temp'
month_name = 'Month'
day_name = 'Day'

tMin = 60
tMax = 80

```

```

def calculate_mean(factor, tmin=tMin, tmax=tMax):
    """
    Рассчитывает среднее значение по столбцу factor для строк,
    где температура T находится в диапазоне [tmin, tmax]

    Parameters:
    factor (str): название столбца
    tmin (float): минимальное значение температуры (по умолчанию 60)
    tmax (float): максимальное значение температуры (по умолчанию 80)

    Returns:
    float: среднее значение указанного фактора
    """
    try:
        df = pd.read_csv('dataset.csv')

        filtered_data = df[(df[temp_name] >= tmin) & (df[temp_name] <= tmax)]

        if len(filtered_data) == 0:
            raise ValueError(f"Нет данных с температурой в диапазоне [{tmin}, {tmax}]")

        mean_value = filtered_data[factor].mean()

        count = len(filtered_data)
        print(f"Найдено {count} строк с температурой от {tmin} до {tmax}°C")
        print(f"Среднее значение {factor}: {mean_value:.2f}")

        return mean_value

    except FileNotFoundError:
        raise FileNotFoundError("Файл dataset.csv не найден. Убедитесь, что он находится в
текущей директории.")
    except Exception as e:
        raise Exception(f"Ошибка при расчете: {str(e)}")

if __name__ == "__main__":
    try:
        result = calculate_mean(ozone_name)
        print(f"Результат: {result:.2f}")
    except Exception as e:
        print(f"Ошибка: {e}")

```

Приложение 5

```

# TASK 6
# testSet.py
# Реализуйте функцию testSet(perc = 20), которая возвращает тестовое множество, состоящее из
# заданного процента строк. Строки должны выбираться случайным образом. Параметр perc -
# вещественное число от 0 до 100. Если переданное значение параметра выходит за пределы [0;100]
# необходимо выдать сообщение об ошибке.

import pandas as pd
import random

ozone_name = 'Ozone'
solar_name = 'Solar.R'
wind_name = 'Wind'
temp_name = 'Temp'

```

```

month_name = 'Month'
day_name = 'Day'

def testSet(perc=20):
    """
    Возвращает множество, состоящее из заданного процента строк, выбранных случайно

    Parameters:
    perc (int): процент строк для выборки (от 0 до 100, по умолчанию 20)

    Returns:
    set: множество индексов выбранных строк
    """
    try:
        # Проверка корректности параметра
        if not 0 <= perc <= 100:
            raise ValueError("Параметр perc должен быть в диапазоне от 0 до 100")

        # Загрузка данных
        df = pd.read_csv('dataset.csv')

        # Получаем общее количество строк
        total_rows = len(df)

        # Вычисляем количество строк для выборки
        sample_size = int(total_rows * perc / 100)

        # Проверка, что выборка не пустая (для perc > 0)
        if perc > 0 and sample_size == 0:
            sample_size = 1

        # Генерируем случайные индексы без повторений
        random_indices = set(random.sample(range(total_rows), sample_size))

        # Выводим информацию о выборке
        print(f"Всего строк в датасете: {total_rows}")
        print(f"Выбрано {len(random_indices)} строк ({perc}%)")
        print(f"Индексы выбранных строк: {sorted(random_indices)}")

        return random_indices

    except FileNotFoundError:
        raise FileNotFoundError("Файл dataset.csv не найден. Убедитесь, что он находится в
текущей директории.")
    except ValueError as ve:
        raise ve
    except Exception as e:
        raise Exception(f"Ошибка при выполнении функции: {str(e)}")

# Дополнительная функция для получения DataFrame с выбранными строками
def get_test_set_data(perc=20):
    """
    Возвращает DataFrame с выбранными случайными строками

    Parameters:
    perc (int): процент строк для выборки (от 0 до 100)

    Returns:
    tuple: (множество индексов, DataFrame с данными)
    """
    try:
        indices = testSet(perc)
        df = pd.read_csv('dataset.csv')
        test_df = df.iloc[list(indices)].copy()
        return indices, test_df

    except Exception as e:
        raise Exception(f"Ошибка при получении данных: {str(e)}")

if __name__ == "__main__":

```

```
try:  
    print("Пример 1: 20% строк (по умолчанию)")  
    result1 = testSet()  
    print(f"Выбрано {len(result1)} индексов\n")  
  
except ValueError as ve:  
    print(f"Ошибка в параметрах: {ve}")  
except Exception as e:  
    print(f"Ошибка: {e}")
```