МГТУ им. Н. Э. Баумана, кафедра ИУ5

курс “Технология машинного обучения”

Лабораторная работа №1

«Разведочный анализ данных. Исследование и визуализация данных.»

ВЫПОЛНИЛ:

Матюнин да Вейга Р.А.

Группа: ИУ5-61Б

ПРОВЕРИЛ:

Гапанюк Ю.Е.

Москва 2020

Цель лабораторной работы: изучение различных методов визуализации данных.

Задание:

- Выбрать набор данных (датасет).

Для лабораторных работ не рекомендуется выбирать датасеты большого размера.

- Создать ноутбук, который содержит следующие разделы:

1. Текстовое описание выбранного Вами набора данных.
2. Основные характеристики датасета.
3. Визуальное исследование датасета.
4. Информация о корреляции признаков.

- Сформировать отчет и разместить его в своем репозитории на github.

Выполненная работа:

Некоторые считают, что изменения в климате являются главной проблемой нашего времени, тогда как другие считают, что это миф, основанный на недостоверной науке. Давайте посмотрим, как же на самом деле обстоят дела.

В качестве датасета для данной работы, я выбрал данные о температуре поверхности Земли, собранные и предоставленные на сайте:

- https://www.kaggle.com/berkeleyearth/climate-change-earth-surface-temperature-data/data

Датасет представляет из себя набор данных мировых температур от 1750 до 2015 гг. и содержит следующие поля:

- **Date**: поле даты, которая содержит месяц и год. Поля 1750-1850 гг. содержат информацию только о средних температурах (**LandAverageTemperature** и **LandAverageTemperatureUncertainty)**, 1850-2015 гг. еще и о максимальной и минимальной температуры земли и мировой температуры океана и земли.
- **LandAverageTemperature**: средняя температура Земли (в Цельсиях).
- **LandAverageTemperatureUncertainty**: 95% доверительный интервал от среднего значения температуры
- **LandMaxTemperature**: максимальная температура Земли (в Цельсиях)
- **LandMaxTemperatureUncertainty**: 95% доверительный интервал от максимального значения температуры
- **LandMinTemperature**: минимальная температура Земли (в Цельсиях)
- **LandMinTemperatureUncertainty**: 95% доверительный интервал от минимального значения температуры
- **LandAndOceanAverageTemperature**: средняя температура суши и океана (в Цельсиях)
- **LandAndOceanAverageTemperatureUncertainty**: 95% доверительный интервал от среднего значения температуры суши и океана

Текст программы и экранные формы с примерами выполнения программы:

In [1]:
```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
%matplotlib inline
sns.set(style="ticks")
```

In [2]:
```
data = pd.read_csv('data/GlobalTemperatures.txt', sep=",")
```

In [3]:
```
# Размер датасета - 3192 строк, 9 колонок
data.shape
```

Out[3]:
```
(3192, 9)
```

In [4]:
```
total_count = data.shape[0]
print('Всего строк: {}'.format(total_count))
```

```
Всего строк: 3192
```

In [5]:
```
# Список колонок
data.columns
```

Out[5]:
```
Index(['dt', 'LandAverageTemperature', 'LandAverageTemperatureUncertainty',
       'LandMaxTemperature', 'LandMaxTemperatureUncertainty',
       'LandMinTemperature', 'LandMinTemperatureUncertainty',
       'LandAndOceanAverageTemperature',
       'LandAndOceanAverageTemperatureUncertainty'],
      dtype='object')
```

In [6]:
```
# Список колонок с типами данных
data.dtypes
```

Out[6]:
```
dt                                            object
LandAverageTemperature                       float64
LandAverageTemperatureUncertainty            float64
LandMaxTemperature                           float64
LandMaxTemperatureUncertainty                float64
LandMinTemperature                           float64
LandMinTemperatureUncertainty                float64
LandAndOceanAverageTemperature               float64
LandAndOceanAverageTemperatureUncertainty    float64
dtype: object
```

Некоторые данные в датасете отсутствуют, такие, как информация о максимальных и минимальных температурах Земли и мировой температуры океана и суши в период 1750-1850 гг.

In [7]:
```
# Проверим наличие пустых значений
# Цикл по колонкам датасета
for col in data.columns:
    # Количество пустых значений
    temp_null_count = data[data[col].isnull()].shape[0]
    print('{} - {}'.format(col, temp_null_count))
```

```
dt - 0
LandAverageTemperature - 12
LandAverageTemperatureUncertainty - 12
LandMaxTemperature - 1200
LandMaxTemperatureUncertainty - 1200
LandMinTemperature - 1200
LandMinTemperatureUncertainty - 1200
LandAndOceanAverageTemperature - 1200
LandAndOceanAverageTemperatureUncertainty - 1200
```

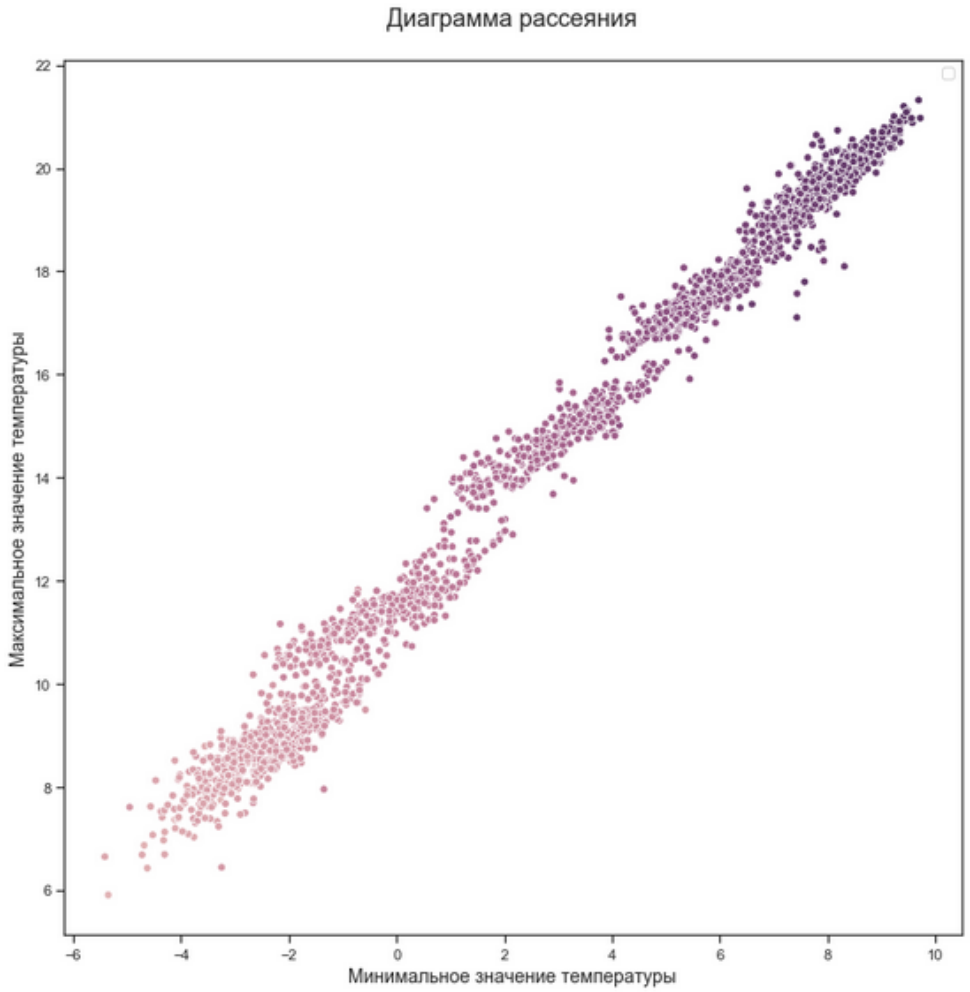
Визуальное исследование датасета.

Визуально исследовать наш датасет мы будем при помощи диаграмм рассеивания и гистограмм. С помощью диаграммы рассеивания мы сможем оценить существуют ли отношения или корреляция между этими двумя переменными, например, для максимальных и минимальных значений.

```
In [9]: fig, ax = plt.subplots(figsize=(12,12))
        sns.scatterplot(ax=ax, x='LandMinTemperature', y='LandMaxTemperature',
        plt.legend('')
        plt.title(r'Диаграмма рассеяния', fontsize=18, y=1.03);

        plt.xlabel('Минимальное значение температуры', fontsize=14)
        plt.ylabel('Максимальное значение температуры', fontsize=14)
```

```
Out[9]: Text(0, 0.5, 'Максимальное значение температуры')
```



Можно видеть, что между полями присутствует положительная (оба значения увеличиваются), линейная зависимость.

С помощью гистограммы мы можем оценить плотность вероятности распределения данных для минимальных, средних и максимальных температур.

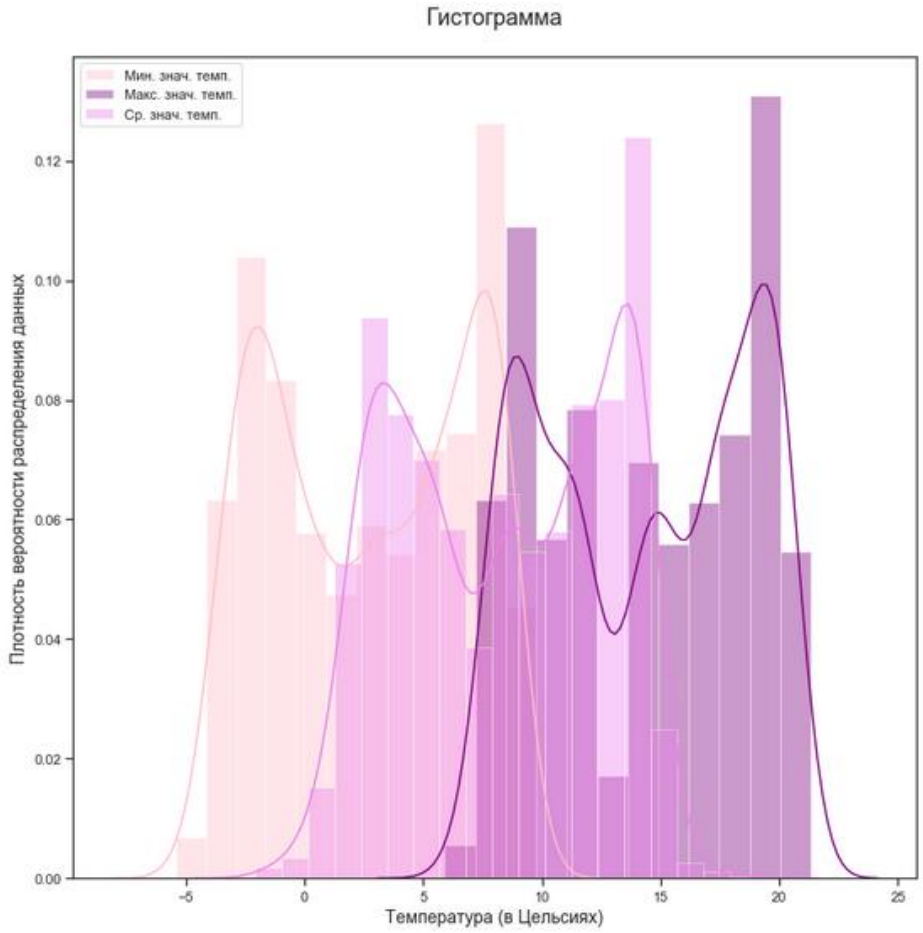
```
In [31]: bplots(figsize=(12,12))
         a['LandMinTemperature'], label = u'Мин. знач. темп.', color='pink')
         a['LandMaxTemperature'], label = u'Макс. знач. темп.', color='purple')
         a['LandAverageTemperature'], label = u'Ср. знач. темп.', color='violet'



         ограмма', fontsize=18, y=1.03);

         ература (в Цельсиях)', fontsize=14)
         ность вероятности распределения данных', fontsize=14)
```

```
Out[31]: Text(0, 0.5, 'Плотность вероятности распределения данных')
```



Комбинация гистограмм и диаграмм рассеивания для всего набора данных: выводится матрица графиков с группировкой по значениям признака средней температуры (**LandAverageTemperature**). На пересечении строки и столбца, которые соответствуют двум показателям, строится диаграмма рассеивания. В главной диагонали матрицы строятся гистограммы распределения соответствующих показателей.

```
In [*]: sns.pairplot(data, hue='LandAverageTemperature')
```

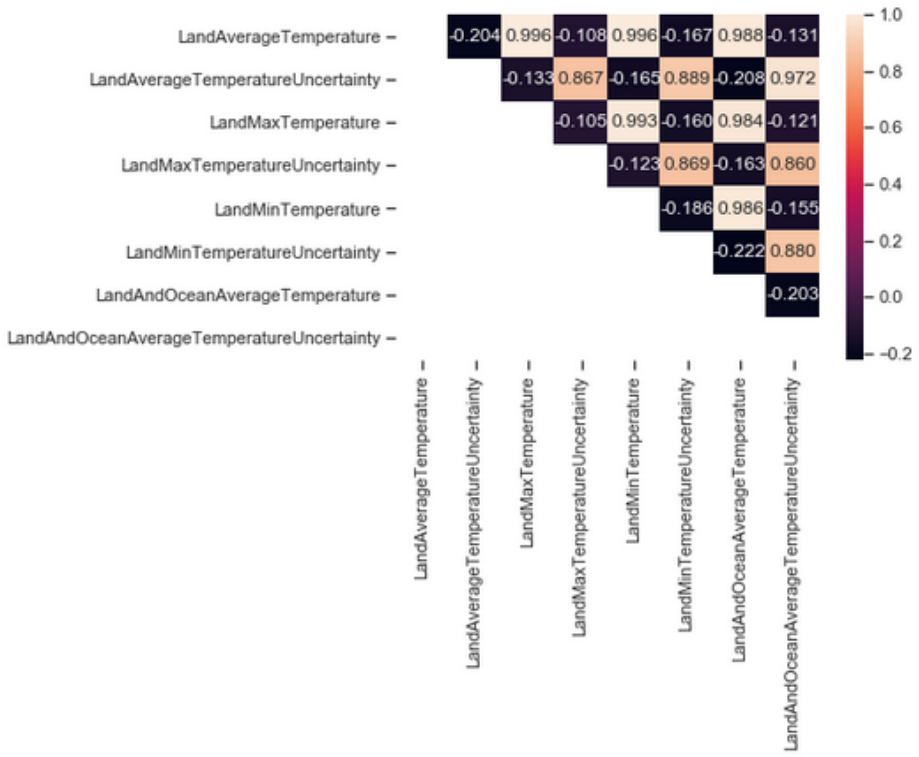
```
c:\users\r a t i\appdata\local\programs\python\python38\lib\site-p
ackages\seaborn\distributions.py:288: UserWarning: Data must have
variance to compute a kernel density estimate.
  warnings.warn(msg, UserWarning)
```

Информация о корреляции признаков

Проверка корреляции признаков позволяет решить две задачи:

1. Понять какие признаки (колонки датасета) наиболее сильно коррелируют с целевым признаком (например с средней температурой **LandAverageTemperature)**. Именно эти признаки будут наиболее информативными для моделей машинного обучения. Признаки, которые слабо коррелируют с целевым признаком, можно попробовать исключить из построения модели, иногда это повышает качество модели. Нужно отметить, что некоторые алгоритмы машинного обучения автоматически определяют ценность того или иного признака для построения модели.
2. Понять какие нецелевые признаки линейно зависимы между собой. Линейно зависимые признаки, как правило, очень плохо влияют на качество моделей. Поэтому если несколько признаков линейно зависимы, то для построения модели из них выбирают какой-то один признак.

```
In [16]: # Треугольный вариант матрицы
         mask = np.zeros_like(data.corr(), dtype=np.bool)
         # чтобы оставить нижнюю часть матрицы
         # mask[np.triu_indices_from(mask)] = True
         # чтобы оставить верхнюю часть матрицы
         mask[np.tril_indices_from(mask)] = True
         sns.heatmap(data.corr(), mask=mask, annot=True, fmt='.3f')
Out[16]: <matplotlib.axes._subplots.AxesSubplot at 0x25138712a30>
```



Корреляционная матрица содержит коэффициенты корреляции между всеми парами признаков, она симметрична относительно главной диагонали. На главной диагонали расположены единицы (корреляция признака самого с собой). На основе корреляционной матрицы можно сделать выводы, которые помогут с решениями задач или для определения ненужных в выборке значений.

- Ноутбук с выполненной работой и отчет размещены в репозитории на github: https://github.com/Yorati/TMO