Лабораторная работа 2.
Анализ данных климата с записными книжками Azure

Цель лабораторной работы. Получить практические знания по разработке регрессионных моделей машинного обучения с учителем

Загрузка данных для модели

1. В первой ячейке задайте тип ячейки **Markdown** и введите в нее "Анализ изменений климата в записной книжке Azure" (рис. 2.1).

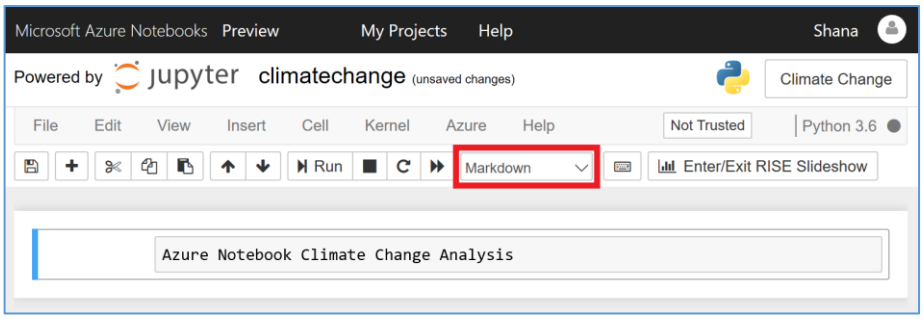


Рисунок 2.1 – Определение ячейки Markdown

2. Установите тип новой ячейки Code (Код). Затем необходимо импортировать библиотеки matplotlib, numpy, sklearn и seaborn (рис. 2.2).

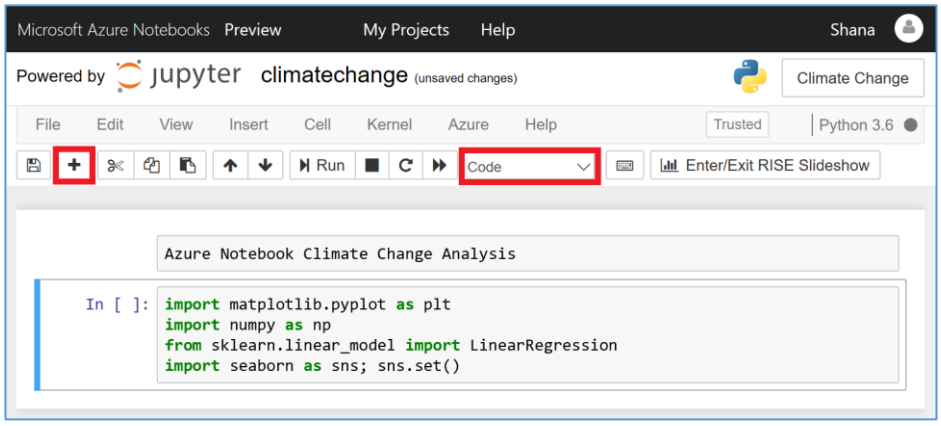


Рисунок 2.2 – Добавление ячейки кода

3. Скачайте архив файлов 5-year-mean-1951-1980.csv и 5-year-mean-1882-2012.csv, которые можно найти по этой ссылке. Разархивируйте файлы и переместите их в папку записной книжки.
4. Загрузите данные для анализа в проект. Для этого применим метод np.loadtxt .

```
yearsBase, meanBase = np.loadtxt('5-year-mean-1951-1980.csv',
                                 delimiter=',', usecols=(0, 1), unpack=True)
years, mean = np.loadtxt('5-year-mean-1882-2014.csv',
                         delimiter=',', usecols=(0, 1), unpack=True)
```

После выполнения функции loadtxt библиотеки NumPy данные будут находятся в памяти и могут использоваться приложением.

5. Для создания точечной диаграммы необходимо добавь следующий код, который использует библиотеку Matplotlib.

```
plt.scatter(yearsBase, meanBase)
plt.title('scatter plot of mean temp difference vs year')
plt.xlabel('years', fontsize=12)
plt.ylabel('mean temp difference', fontsize=12)
plt.show()
```

В результате выполнения данного кода будет сформирована точечная диаграмма (рис. 2.3).

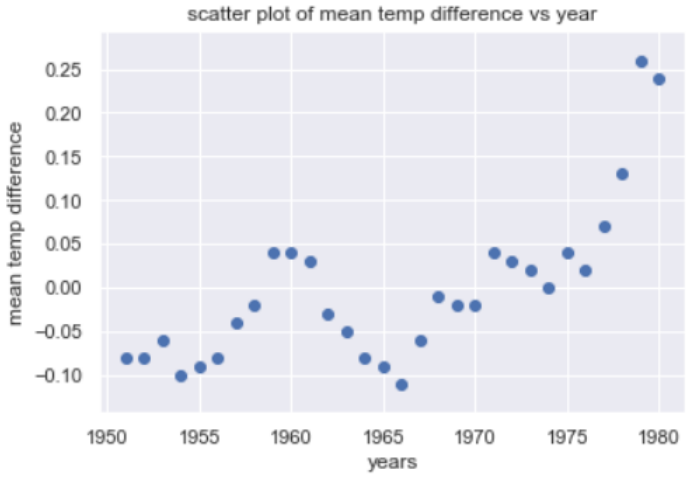


Рисунок 2.3 – Точечная диаграмма, созданная Matplotlib

Набор данных, который вы загрузили, использует средние значения за 30 лет между 1951 и 1980 годами для вычисления базовой температуры за этот период, а затем рассчитывает средние температуры за 5 лет, чтобы рассчитать разницу между средними значениями за 5 и за 30 лет для каждого года. Точечная диаграмма показывает различия годовой температуры.

Выполнение линейной регрессии с помощью Numpy

Точечные диаграммы — это удобное средство для визуализации данных, но предположим, что вы хотите наложить на точечную диаграмму линию тренда, чтобы показать тенденцию с течением времени. Один из способов вычисления таких линий трендов — линейная регрессия. Для выполнения линейной регрессии будем использовать NumPy и Matplotlib для рисования линии тренда на основе данных.

С помощью метода *np.polyfit* на основе данных для исследования создадим массивы коэффициентов линейной регрессии *m* и *b*. Определим функцию *f(x)* вычисления значений для массивов *m* и *b.* Построим график линейной регрессии.

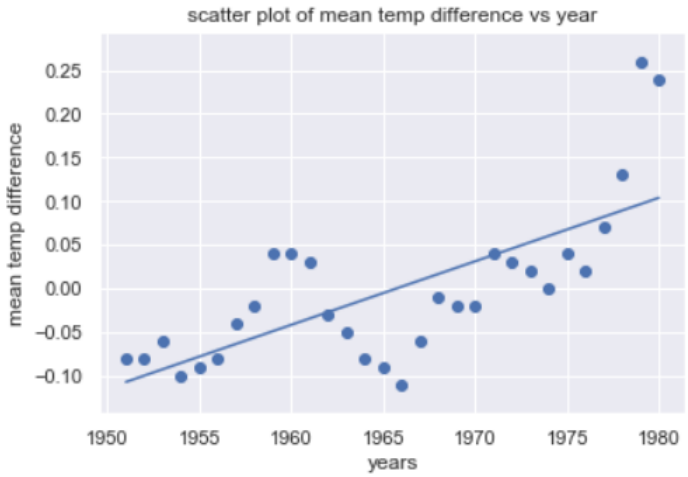
```
# Creates a linear regression from the data points
m,b = np.polyfit(yearsBase, meanBase, 1)

# This is a simple y = mx + b line function
def f(x):
    return m*x + b

# This generates the same scatter plot as before, but adds a line plot using the function above
plt.scatter(yearsBase, meanBase)
plt.plot(yearsBase, f(yearsBase))
plt.title('scatter plot of mean temp difference vs year')
plt.xlabel('years', fontsize=12)
plt.ylabel('mean temp difference', fontsize=12)
plt.show()

# Prints text to the screen showing the computed values of m and b
print(' y = {0} * x + {1}'.format(m, b))
plt.show()
```

Результат построения линейной регрессии приведен на (рис. 2.4)




```
y = 0.007279199110122374 * x + -14.309265850945529
```

Рисунок 2.4 – Точечная диаграмма с линией регрессии

По линии регрессии видно, что разница между средней температурой за 30 лет и из 5 лет увеличивается с течением времени. Большую часть вычислительных операций, необходимых для формирования линии регрессии, выполняет функция *polyfit* библиотеки NumPy, которая вычисляет значения *m* и *b* в уравнении *y = mx + b*.

Выполнение линейной регрессии с помощью Scikit-learn

Вычисление регрессии можно выполнить с помощью пакета scikit-learn. Это превосходное средство создания моделей машинного обучения для извлечения ценной информации из данных. В этом разделе пакет scikit-learn используется для вычисления линии тренда для данных NASA о климате.

Для построения модели регрессии используется метод *LinearRegression.* Обучение модели реализуется методом *fit,* а проверка точности модели реализуется методом *predict.* Визуализация результатов машинного обучения осуществляется средствами пакета Matplotlib.

```
In [5]: # Pick the Linear Regression model and instantiate it
        model = LinearRegression(fit_intercept=True)

        # Fit/build the model
        model.fit(yearsBase[:, np.newaxis], meanBase)
        mean_predicted = model.predict(yearsBase[:, np.newaxis])

        # Generate a plot like the one in the previous exercise
        plt.scatter(yearsBase, meanBase)
        plt.plot(yearsBase, mean_predicted)
        plt.title('scatter plot of mean temp difference vs year')
        plt.xlabel('years', fontsize=12)
        plt.ylabel('mean temp difference', fontsize=12)
        plt.show()

        print(' y = {0} * x + {1}'.format(model.coef_[0], model.intercept_))
```

Результат построения линейной регрессии приведен на (рис. 2.5)

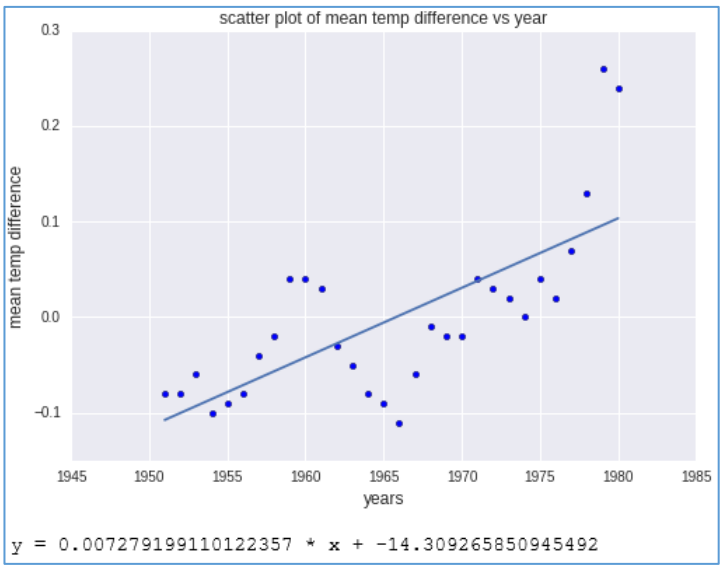


Рисунок 2.5 – Точечная диаграмма с линией регрессии, вычисленной scikit-learn

Выходные данные почти идентичны результатам, полученным при использовании пакета NumPy. Разница в том, scikit-learn проделал за вас больше работы. В частности, не нужно писать код линейной функции, как в случае с NumPy; функция scikit-learn *LinearRegression* сделала это автоматически. *scikit-learn* поддерживает многие типы регрессии, и это удобно при построении сложных моделей машинного обучения.

Анализ данных с использованием Seaborn

В данном разделе будем использовать пакет Seaborn, библиотеку для статистической визуализации, чтобы построить второй из двух наборов данных, который был загружен и охватывает период с 1882 по 2014 год. Пакет Seaborn может создать линию регрессии с проекцией, показывающей, где должны размещаться точки данных на основе регрессии, с одним простым вызовом функции.

Добавьте ячейку для кода и вставьте следующий код.

```
plt.scatter(years, mean)
plt.title('scatter plot of mean temp difference vs year')
plt.xlabel('years', fontsize=12)
plt.ylabel('mean temp difference', fontsize=12)
sns.regplot(yearsBase, meanBase)
plt.show()
```

Результат построения точечной диаграммы с линией регрессии и визуальным представлением диапазона, в который должны попадать точки данных приведен на рис. 2.6.

Рисунок 2.6 – Сравнение фактических и прогнозных значений, созданных с помощью Seaborn

Обратите внимание на то, каким образом точки данных для первых 100 лет соответствуют прогнозируемым значениям, а точки данных примерно с 1980 года — нет. Именно на таких моделях ученые строят предположения об ускорении изменения климата.

Задание на самостоятельную работу

Выберите датасет по вашему усмотрению и постойте регрессионную модель машинного обучения.

Задание по лабораторной работе

1. Изучить теоретический материал.
2. В среде Jupyter Notebook выполнить разработку моделей машинного обучения.
3. Подготовить отчет по лабораторной работе, включающий описание выполнения работы, код на языке Python и скриншоты результатов выполнения
4. Загрузить отчет по лабораторной работе и файл проекта на учебный портал.