МОСКОВСКИЙ ИНСТИТУТ ЭЛЕКТРОННОЙ ТЕХНИКИ
Институт системной и программной инженерии
и информационных технологий (Институт СПИНТех)

Лабораторный практикум по курсу
"Свёрточные нейронные сети в компьютерном зрении"
(02/20 – 06/20)

Лабораторная работа 2
«Анализ данных в Jupyter Notebook»

На этом лабораторном занятии вы научитесь использовать на практике Jupyter Notebook. Эта работа должна дать вам более четкое представление о том, почему Jupyter Notebooks так популярны.

Итак, в этом компьютерном практикуме вы произведете анализ данных в блокноте Jupyter. Допустим, вы аналитик данных, и вам было поручено выяснить, как исторически менялась прибыль крупнейших компаний в США. У вас для этого есть набор данных о компаниях из списка Fortune 500, охватывающих более 50 лет с момента первой публикации списка в 1955 году, собранных из открытого архива Fortune. CSV файл с этими данными для анализа включен в задание, вы можете найти его в директории с разработками лабораторных работ - *fortune500.csv*. Итак, ваша цель – узнать, как исторически менялась прибыль крупнейших компаний США.

Прежде чем приступить к выполнению лабораторных заданий, настоятельно рекомендуется ознакомиться с языком программирования Python, пакетом для анализа данных Pandas, а также основами работы с Jupyter Notebook, подробно рассмотренными в первом компьютерном практикуме данного курса.

1. Настройка

Обычно начинают с ячейки кода, предназначенной для импорта и настройки.

Упражнение
Запустите Jupyter Notebook и создайте новый блокнот, нажав кнопку «Создать» на панели инструментов в правом верхнем углу и выберите «Python 3». Дайте вашему блокноту соответствующее название из панели управления Jupyter, не забудьте, что для этого сперва необходимо закрыть ядро. Далее запустите заново ваш блокнот и напечатайте следующий код:

```
In [1]: import pandas as pd
        import matplotlib.pyplot as plt
        import seaborn as sns
        sns.set(style="darkgrid")

In [2]: df = pd.read_csv('fortune500.csv')
```

Здесь мы импортируем pandas для работы с нашими данными, Matplotlib для построения графиков и Seaborn для улучшения внешнего вида наших графиков. Обычно также импортируется NumPy, но в нашем случае, мы используем его через pandas. Последняя строка не является командой Python, она называется строковой магией, для инструктирования Jupyter захватывать графики Matplotlib и отображать их в выходных данных ячейки, это одна из расширенных функций, выходящих за рамки данной работы.

В следующей ячейке мы загружаем наш набор данных. Целесообразно делать это в отдельной ячейке на случай, если нам понадобится перезагрузить ее в любой момент.

2. Сохранение и контрольная точка

Теперь, когда вы начали создавать свой первый проект, полезно будет регулярно сохраняться. Нажатие **Ctrl + S** сохранит вашу записную книжку, вызвав команду «Save and Checkpoint».

Каждый раз, когда вы создаете новую записную книжку, создается не только файл вашей записной книжки, но и файл контрольной точки, он также является файлом **.ipynb** и будет расположен в скрытом подкаталоге вашего места сохранения с именем **.ipynb_checkpoints**. По умолчанию Jupyter каждые 120 секунд автоматически сохраняет ваш блокнот в этот файл контрольных точек, не изменяя основной файл блокнота. Когда вы сохраняете контрольную точку, файлы записной книжки и контрольной точки обновляются. Следовательно, контрольная точка позволяет вам восстановить несохраненную работу в случае непредвиденной проблемы. Вы можете вернуться к контрольной точке из меню через «*File> Revert to Checkpoint*».

Упражнение

Сохраните ваш Jupyter Notebook.

3. Изучение набора данных

Ваш блокнот благополучно сохранен, и вы загрузили наш набор данных переменную **df** – в наиболее часто используемую структуру данных **pandas**, которая называется **DataFrame** и в основном выглядит как таблица. Давайте посмотрим, как выглядят эти данные?

Упражнение

Создайте новую ячейку кода и введите команду df.head().

Результат должен выглядеть так:

```
In [3]: df.head()
```

Out[3]:

	Year	Rank	Company	Revenue (in millions)	Profit (in millions)
0	1955	1	General Motors	9823.5	806
1	1955	2	Exxon Mobil	5661.4	584.8
2	1955	3	U.S. Steel	3250.4	195.4
3	1955	4	General Electric	2959.1	212.6
4	1955	5	Esmark	2510.8	19.1

Далее создайте ещё одну ячейку и введите команду df.tail().
Вы должны получить следующий результат:

```
In [4]: df.tail()
```

Out[4]:

	Year	Rank	Company	Revenue (in millions)	Profit (in millions)
25495	2005	496	Wm. Wrigley Jr.	3648.6	493
25496	2005	497	Peabody Energy	3631.6	175.4
25497	2005	498	Wendy's International	3630.4	57.8
25498	2005	499	Kindred Healthcare	3616.6	70.6
25499	2005	500	Cincinnati Financial	3614.0	584

Итак, у нас есть необходимые нам столбцы, и каждая строка соответствует истории каждой компании за один год.

Давайте переименуем эти столбцы, чтобы мы могли обратиться к ним позже.

Упражнение

Создайте новую ячейку и выполните следующую команду:

```
In [ ]: df.columns = ['year', 'rank', 'company', 'revenue', 'profit']
```

Затем снова запустите ячейку с командой df.head(). Вы должны получить следующий результат:

```
In [6]: df.head()
```

Out[6]:

	year	rank	company	revenue	profit
0	1955	1	General Motors	9823.5	806
1	1955	2	Exxon Mobil	5661.4	584.8
2	1955	3	U.S. Steel	3250.4	195.4
3	1955	4	General Electric	2959.1	212.6
4	1955	5	Esmark	2510.8	19.1

4. Проверка набора данных

Далее вам нужно изучить набор данных. Являются ли они завершенными? Распознало ли pandas их, как ожидалось? Отсутствуют ли в них какие-либо значения?

Упражнение

Создайте новую ячейку и выполните команду *len(df)*:

Вы должны получить следующий результат:

```
In [7]: len(df)
Out[7]: 25500
```

Итак, у нас есть 500 строк за каждый год с 1955 по 2005 год включительно.

Давайте проверим, был ли наш набор данных импортирован правильно. Простая проверка заключается в том, чтобы увидеть, были ли типы данных (или dtypes) правильно интерпретированы.

Упражнение

Создайте новую ячейку и выполните команду *df.dtypes:*

Вы должны получить следующий результат:

```
In [8]: df.dtypes
Out[8]: year         int64
        rank         int64
        company     object
        revenue    float64
        profit      object
        dtype: object
```

Проанализируйте полученные данные.

Внимательно изучите полученный результат, вы должны были заметить, что с колонкой profit что-то не так – мы ожидаем, что она должна иметь тип float64, как и колонка revenue. Данный результат указывает на то, что колонка profit, вероятно, содержит нецелые значения, давайте это проверим.

Упражнение

Создайте новую ячейку и выполните следующую команду*:*

```
In [ ]: non_numberic_profits = df.profit.str.contains('[^0-9.-]')
        df.loc[non_numberic_profits].head()
```

Результат должен выглядеть так:

Out[9]:

	year	rank	company	revenue	profit
228	1955	229	Norton	135.0	N.A.
290	1955	291	Schlitz Brewing	100.0	N.A.
294	1955	295	Pacific Vegetable Oil	97.9	N.A.
296	1955	297	Liebmann Breweries	96.0	N.A.
352	1955	353	Minneapolis-Moline	77.4	N.A.

Посмотрите на колонку profit! Некоторые значения являются строками, которые использовались для указания отсутствующих данных - 'N.A.'. Что же нам с этим делать? Это зависит от того, сколько значений пропущено.

Упражнение

Создайте новую ячейку и выполните следующую команду *len(df.profit[non_numberic_profits]).*

В результате вы должны получить:

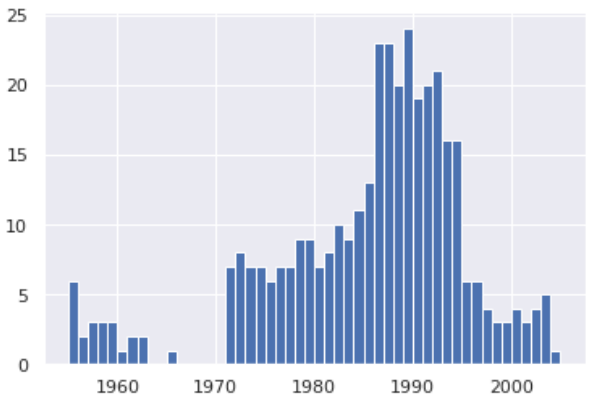
```
In [10]: len(df.profit[non_numberic_profits])
Out[10]: 369
```

Итак, мы видим, что это небольшая часть нашего набора данных, хотя и не совсем несущественная, поскольку составляет около 1,5%. Если строки, содержащие N.A., примерно одинаково распределены по годам, самым простым решением было бы просто удалить их. Давайте кратко рассмотрим их распределение.

Упражнение

Создайте новую ячейку и выполните следующую команду:

```
In [11]: bin_sizes, _, _ = plt.hist(df.year[non_numberic_profits],bins=range(1955, 2006))
```



На первый взгляд, мы видим, что недопустимые значения за один год составляют менее 25, а поскольку существует 500 точек данных в год, удаление этих значений будет

составлять менее 4% данных для худших лет. Действительно, кроме всплеска около 90-х годов, большинство лет имеют менее половины недостающих значений пика. Допустим, что для наших целей это приемлемо, и просто удалим эти строки.

Упражнение

Создайте новую ячейку и выполните следующую команду:

```
In [12]: df = df.loc[~non_numberic_profits]
         df.profit = df.profit.apply(pd.to_numeric)
```

Теперь давайте проверим, что у нас получилось.

Упражнение

Выполните заново следующие команды:

```
In [13]: len(df)
Out[13]: 25131
```

```
In [14]: df.dtypes
Out[14]: year         int64
         rank         int64
         company     object
         revenue    float64
         profit     float64
         dtype: object
```

Если вы получили результат такой же как на последнем изображении, значит вы успешно завершили настройку набора данных!

5. Графики с matplotlib

Теперь мы можем перейти к решению данного поставленной задачи, а именно, построить график средней прибыли за год и рассчитать доход. Но для начала определим некоторые необходимые переменные и метод.

Упражнение

Выполните следующие команды и поясните, что они делают:

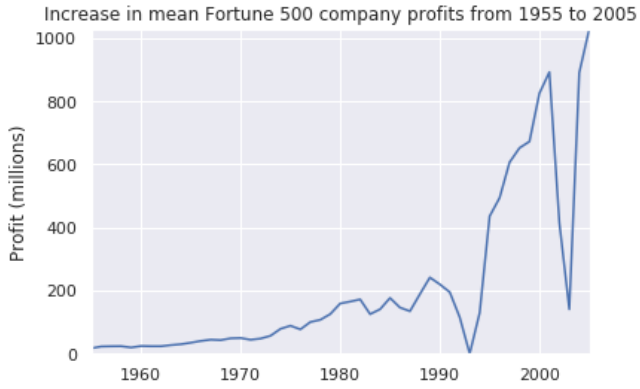
```
In [15]: group_by_year = df.loc[:, ['year', 'revenue', 'profit']].groupby('year')
         avgs = group_by_year.mean()
         x = avgs.index
         y1 = avgs.profit
         def plot(x, y, ax, title, y_label):
             ax.set_title(title)
             ax.set_ylabel(y_label)
             ax.plot(x, y)
             ax.margins(x=0, y=0)
```

Теперь давайте наконец-то построим график прибыли!

Упражнение

Постройте график с помощью следующих команд:

```
In [16]: fig, ax = plt.subplots()
plot(x,y1,ax,'Increase in mean Fortune 500 company profits from 1955 to 2005','Profit (millions)')
```



Проанализируйте полученный график.

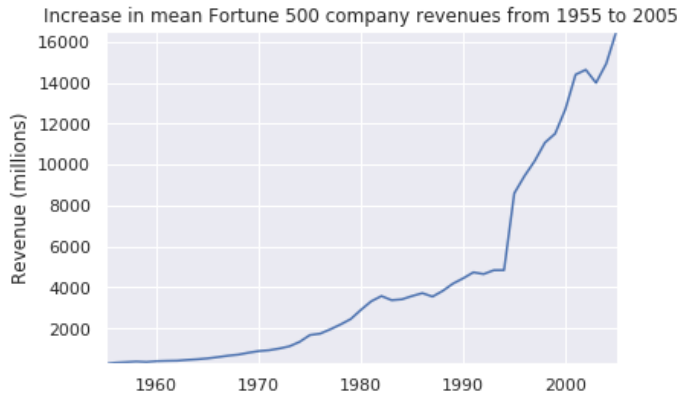
Кажется, график нашей функции немного похож на экспоненту, но с огромными провалами. Они должны соответствовать рецессии начала 1990-х. Довольно интересно увидеть это в данных. Но почему прибыль возвращается к еще более высоким уровням после каждой рецессии?

Может быть, доходы могут рассказать нам больше?

Упражнение

Постройте график с помощью следующих команд:

```
y2 = avgs.revenue
fig, ax = plt.subplots()
plot(x, y2, ax, 'Increase in mean Fortune 500 company revenues from 1955 to 2005', 'Revenue (millions)')
```



Проанализируйте полученный график.

Здесь мы видим другую сторону истории. Доходы отнюдь не так сильно пострадали!

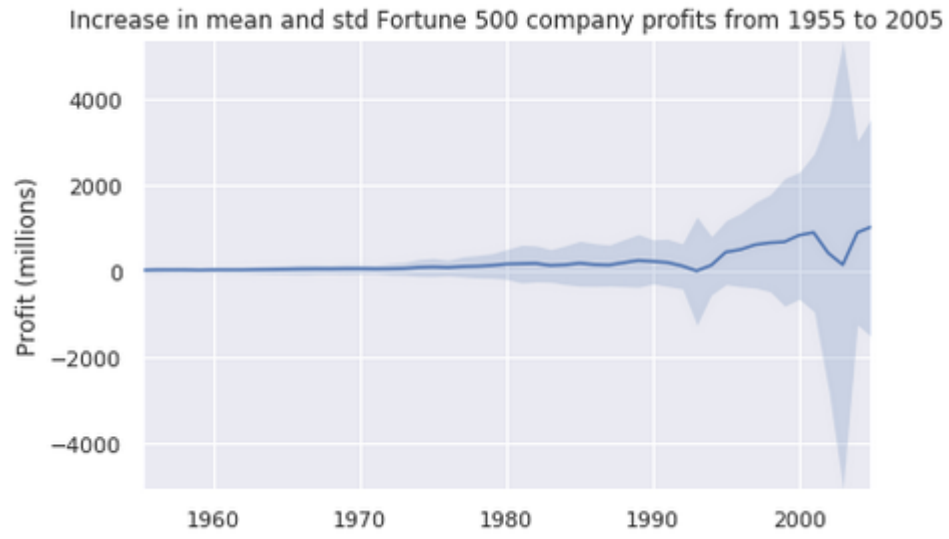
Давайте попробуем наложить эти графики с +/- их стандартными отклонениями.

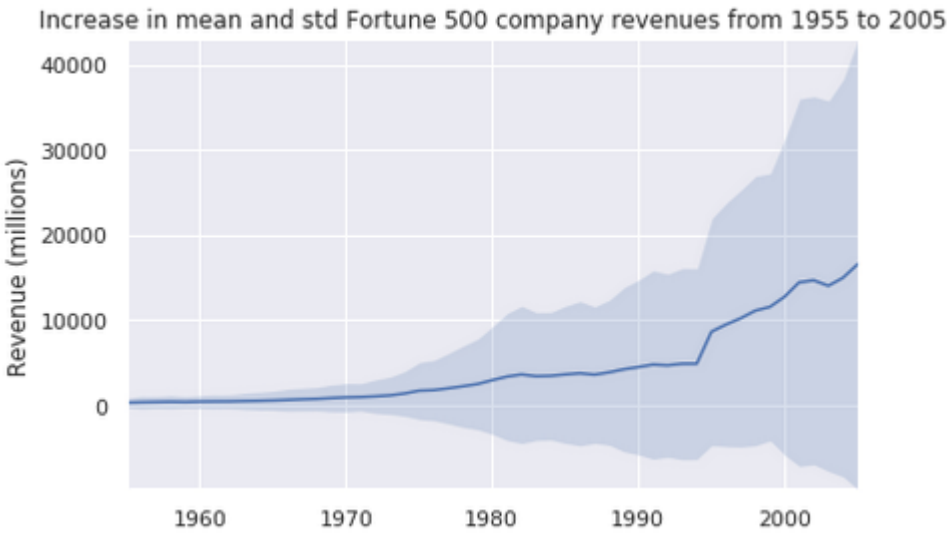
Упражнение

Создайте новую ячейку и выполните следующие команды:

```
In [ ]:
    def plot_with_std(x, y, stds, ax, title, y_label):
        ax.fill_between(x, y - stds, y + stds, alpha=0.2)
        plot(x, y, ax, title, y_label)
    fig, (ax1, ax2) = plt.subplots(ncols=2)
    title = 'Increase in mean and std Fortune 500 company %s from 1955 to 2005'
    stds1 = group_by_year.std().profit.values
    stds2 = group_by_year.std().revenue.values
    plot_with_std(x, y1.values, stds1, ax1, title % 'profits', 'Profit (millions)')
    plot_with_std(x, y2.values, stds2, ax2, title % 'revenues', 'Revenue (millions)')
    fig.set_size_inches(14, 4)
    fig.tight_layout()
```

Вы должны получить следующие графики:





Проанализируйте полученные результаты!

Как видите, стандартные отклонения огромны. Некоторые компании из списка Fortune 500 зарабатывают миллиарды, в то время как другие теряют миллиарды, и риск

увеличивается вместе с ростом прибыли за последние годы. Возможно, некоторые компании работают лучше, чем другие.

Итак, данный блокнот помог нам легко исследовать наш набор данных в одном месте без переключения контекста между приложениями, и наша работа является доступной и воспроизводимой. Если бы мы хотели создать более краткий отчет для конкретной аудитории, мы могли бы быстро реорганизовать нашу работу, объединив ячейки и удалив промежуточный код.

Задание к лабораторной работе

1) Проведите анализ исторического изменения прибыли крупнейших компаний США с помощью Jupyter Notebook, последовательно выполнив все упражнения из данного практикума.

2) Представьте отчёт в виде файла **.ipynb** с результатами вашей работы.

3) Проанализируйте полученные результаты, что вы можете сказать об изменении прибыли рассмотренных крупнейших компаний? Проанализируйте последние, полученные графики, являются ли прибыли первых 10% более или менее волатильными, чем нижних 10%?