

Министерство Образования Российской Федерации
Университет Иннополис

ИТОГОВАЯ АТТЕСТАЦИЯ
по специальности «Архитектор в области искусственного интеллекта»
на тему:

**№18. Разработка и тестирование приложения для анализа и прогнозирования
метеорологических данных, влияющих на промышленные объекты, с
использованием машинного обучения**

Выполнил: Сорокин Максим Евгеньевич
Руководитель: Корнеева Елена Игоревна

Иннополис, 2023

СОДЕРЖАНИЕ

| | |
|---|----|
| Этап 1. Исследование и поиск применения машинного обучения | 3 |
| Этап 2. Подготовка данных | 11 |
| Этап 3. Обработка данных перед загрузкой в модель | 25 |
| Этап 4. Разделение данных на тренировочную, тестовую и валидационную. Обучение и тестирование моделей..... | 30 |
| Этап 5. Сохранение и загрузка моделей. | 42 |
| Итог..... | 43 |

Этап 1. Исследование и поиск применения машинного обучения

Найден датасет нефтяной скважины №807 на Kaggle (2013-2021). Этую скважину будем исследовать вместе с погодными условиями:

<https://www.kaggle.com/datasets/ruslanzalevskikh/oil-well>

The screenshot shows the Kaggle interface for the 'Oil well' dataset. On the left, there's a sidebar with navigation links like 'Create', 'Home', 'Competitions', 'Datasets' (which is selected), 'Models', 'Code', 'Discussions', 'Learn', and 'More'. Below that is a 'Your Work' section with a list of recent datasets: 'Oil well', 'Power consumption in ...', 'Titanic Dataset', 'Electrical power qualit...', and 'Laptop Price Predictio...'. The main content area has a title 'Oil well' and a subtitle 'Oil well operation parameters (2013-2021)'. It includes a search bar, a file download button ('Download (123 kB)'), and a thumbnail image of an oil rig. Below the title are tabs for 'Data Card', 'Code (1)', and 'Discussion (1)'. A large section titled 'About Dataset' contains an introduction, a detailed description of the data collection process, and a note about the well being drilled in 2013. To the right, there are sections for 'Usability' (rating 6.76), 'License' (Original Authors), 'Expected update frequency' (Not specified), and 'Tags' (Oil and Gas). At the bottom, there's a note about cookie usage and a Windows activation notice.

The screenshot shows the 'Oil Well' dataset table view. The table has 2941 rows and 9 columns. The columns are: 'Oil well operation...', 'Column1', 'Column2', 'Column3', 'Column4', 'Column5', 'Column6', 'Column7', and 'Column8'. The first row contains descriptive text for each column. Subsequent rows provide data points for each day from July 2013 to June 2014. The table is displayed in a grid format with horizontal and vertical scroll bars. To the right of the table, there's a 'Data Explorer' panel showing 'Version 1 (144.98 kB)' and a file named 'Oil well.xlsx'. The bottom of the screen shows the Windows taskbar with various pinned icons and the system tray.

Есть архивный документ нефтяной скважины №807 на официальном сайте службы государственной охраны объектов культурного наследия Ямало-Ненецкого автономного округа:

<https://nasledie89.yanao.ru/documents/active/47135/>

ГИКЭ по проекту «Разведочная скважина №807 Уренгойского НГКМ»
(площадь 21,05155 га) в Пурском районе ЯНАО Тюменской области в
2019 году.

9 сентября 2019 | Тип: Акт государственной историко-культурной экспертизы | Уровень: Локальный
Деятельность: Заключения ГИКЭ документации, за исключением научных отчетов о выполненных археологических полевых работах, содержащей результаты исследований, в соответствии с которыми определяется наличие или отсутствие объектов, обладающих признаками объекта культурного наследия, на земельных участках, подлежащих воздействию земельных, строительных, непроизводственных, хозяйственных работ, указанных в настоящей статье работ по использованию лесов и иных работ

Скачать 3.07 MB (загружено 8 раз) Основной документ

Скачать 48.12 KB (загружено 0 раз) Сводка 21.05155

Срок обсуждения по 27 сентября 2019 года.
Написать комментарий

Использование лесов и иных работ по проекту «Разведочная скважина №807 Уренгойского НГКМ» (площадь 21,05155 га) в Пурском районе ЯНАО Тюменской области в 2019 году. Историко-культурные изыскания по проекту «Разведочная скважина №807 Уренгойского НГКМ» (площадь 21,05155 га) в Пурском районе ЯНАО Тюменской области в 2019 году. Историко-культурные изыскания по проекту «Разведочная скважина №807 Уренгойского НГКМ» (площадь 21,05155 га) в Пурском районе ЯНАО Тюменской области в 2019 году.

Административное расположение объекта:
Тюменская область, ЯНАО, Пурский район. Объект находится в 114 км по azimuthу 341,21° от аэропорта г. Тарко-Сале, и в 27,1 км по azimuthу 135,49° от аэропорта г. Новый Уренгой.

Объект входит в земельный участок:
Объект расположен в кадастровом квартале 89:05:020501, кадастровые номера участков не присвоены.

Данные о лесопосадке:
По данным Департамента природно-ресурсного регулирования, лесных отношений и развития нефтегазового комплекса ЯНАО, участки расположены на землях, не входящих в состав земель лесного фонда.

К тексту настоящего Акта приложен Научно-технический отчет № 13-2019 Историко-культурные изыскания по проекту «Разведочная скважина №807 Уренгойского НГКМ» (площадь 21,05155 га) в Пурском районе ЯНАО Тюменской области в 2019 году. Омск, 2019.

Письмо о проведении государственной историко-культурной экспертизы по проекту «Разведочная скважина №807 Уренгойского НГКМ» (площадь 21,05155 га) в Пурском районе ЯНАО Тюменской области в 2019 году № 02.09.2019-03 от 02.09.2019 на имя эксперта А.В. Соколова;

Перечень документов, представленных в акте:
Научно-технический отчет № 13-2019 Историко-культурные изыскания по проекту «Разведочная скважина №807 Уренгойского НГКМ» (площадь 21,05155 га) в Пурском районе ЯНАО Тюменской области в 2019 году. Омск, 2019.

Изменено: 26 декабря 2019 17:13 Количество просмотров: 7 Поделиться

Краткие сведения об испрашиваемых под освоение землях:

- «Разведочная скважина №807 Уренгойского НГКМ» (площадь 21,05155 га).

Административное расположение объекта:

- Тюменская область, ЯНАО, Пуровский район. Объект находится в 114 км по азимуту 341,21° от аэропорта г. Тарко-Сале, и в 27,1 км по азимуту 135,49° от аэропорта г. Новый Уренгой.

Данные о кадастровом учете:

- Объект расположен в кадастровом квартале 89:05:020501, кадастровые номера участкам не присвоены.

Набрав в поисковой системе «Яндекс» адрес скважины №807, можно найти сайты с кадастровым номером, например:

<https://konfiskat-torgi.ru/detail/952658/skvazhina-807-glubina-4100-m-adres-rossiya-yamalo-nenetskij-ao-purovskij-rajon-osennij-litsenzionnyij-uchastok-kadastrov>

The screenshot shows a web page from konfiskat-torgi.ru. At the top, there's a navigation bar with links for 'Как купить конфискат', 'Ямало-Ненецкий АО', and 'Получать новые конкурсы'. Below the header, there's a breadcrumb trail: 'Аукционы конфиската / Ямало-Ненецкий АО / Строения / Скважина № 807, глубина 4100 м., адрес: Россия, Ямало-Ненецкий А.О., Пуровский район, Осенний лицензионный участок... №952658, продается конфискат б/у, аукци...'. The main content area displays the auction details for Skvazhina № 807, including its location (Russia, Yamalo-Nenets Autonomous Okrug, Purovsky district, Autumn license plot), description (drilled well No. 807, depth 4100 m), and auction number (952658). It also lists the auction date (April 25, 2023, at 19:59), time (18:00), and bidder (AKTSIONERNOE OBSHCHESTVO «SBERBANK - AUTOMATIZIROVANNAYA SISTEMA TORGOV»). On the right side, there's a sidebar titled 'Навигация' with links to 'Общие сведения', 'Контакты', 'Старт', 'Предмет торгов', 'Порядок участия в торгах', 'Документы', and 'Похожие лоты'. At the bottom of the page, there's a footer with activation information for Windows.

Также находим информацию об объекте строительства на сайте ГИС Торги:

<https://torgi.gov.ru/new/public/notices/view/21000034510000000319>

The screenshot shows the 'View announcement' page for notice number 21000034510000000319. The page title is 'Извещение № 21000034510000000319' (Announcement No. 21000034510000000319) and it is marked as 'Завершено' (Completed). The main content area displays various details about the auction, including the reason for changes (decision of the auction organizer), publication date (04.04.2023), and closing date (05.04.2023). It also lists the type of auction (electronic auction), bidding platform (Additional service for conducting electronic auctions via electronic trading platforms), and the name of the plot (Plot 705, depth 4050 m, address: Russia, Yamalo-Nenets Autonomous Okrug, Purinskaya district, Samburgskiy licensing area, plot 89:05:010306:2390 (RFNI P12720007667)). A sidebar on the left provides links to 'Основные сведения' (General information), 'Организатор торгов' (Auction organizer), 'Сведения о правообладателе/инициаторе торгов' (Information about the owner/initiator of the auction), 'Информация о лотах' (Information about lots), 'Требования к заявкам' (Requirements for bids), 'Условия проведения процедуры' (Procedure implementation conditions), 'Документы извещения' (Announcement documents), and 'Протоколы' (Protocols). A right sidebar includes links to 'История версий' (Version history), 'Подписано ЭП' (Signed with E-signature), and 'Журнал событий' (Event log). The bottom of the screen shows a Windows taskbar with various pinned icons and the date/time as 08.11.2023.

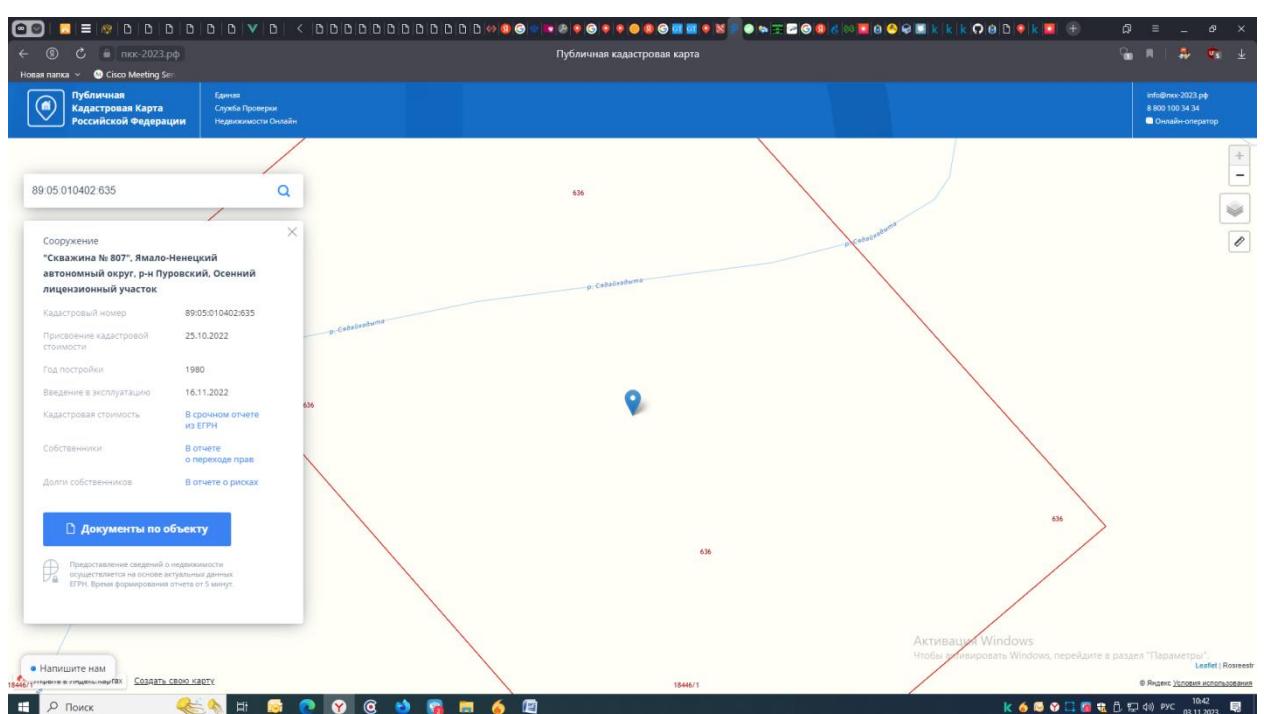
The screenshot shows the 'Information about lots' section for the announcement. It lists two lots: Lot 1 (Plot 705, depth 4050 m, address: Russia, Yamalo-Nenets Autonomous Okrug, Purinskaya district, Samburgskiy licensing area, plot 89:05:010306:2390 (RFNI P12720007667)) and Lot 2 (Plot 807, depth 4100 m, address: Russia, Yamalo-Nenets Autonomous Okrug, Purinskaya district, Oseninii licensing area, plot 89:05:010402:635 (RFNI P12720007668)). The 'Основная информация' (Main information) for Lot 2 includes the subject of ownership (Yamalo-Nenets Autonomous Okrug), ownership status (AO Yamalo-Nenets Autonomous Okrug), object category (Buildings), and object name (Buildings).

Лот 2: Скважина № 807, глубина 4100 м., адрес: Россия, Ямало-Ненецкий А.О., Пуровский район, Осенний лицензионный участок, кадастровый номер 89:05:010402:635 (РНФИ П12720007668)

Эта информация подтверждает кадастровый номер: 89:05:010402:635

Пробуем найти на сайте информацию по кадастровому номеру:

<https://пкк-2023.рф/?cadNumber=89%3A05%3A010402%3A635&lat=66.80035786146756&lng=78.38975066524623&zoom=18>



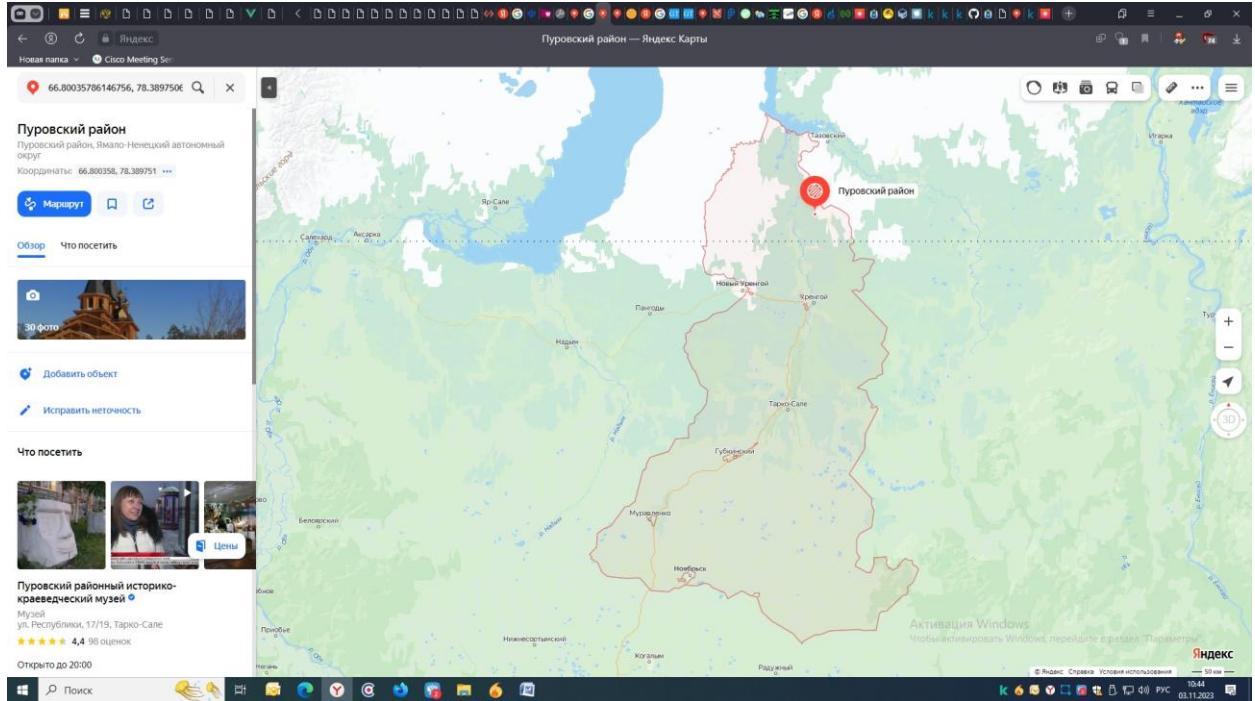
Находим точку на карте и координаты в URL адресе:

lat=66.80035786146756 (Широта)

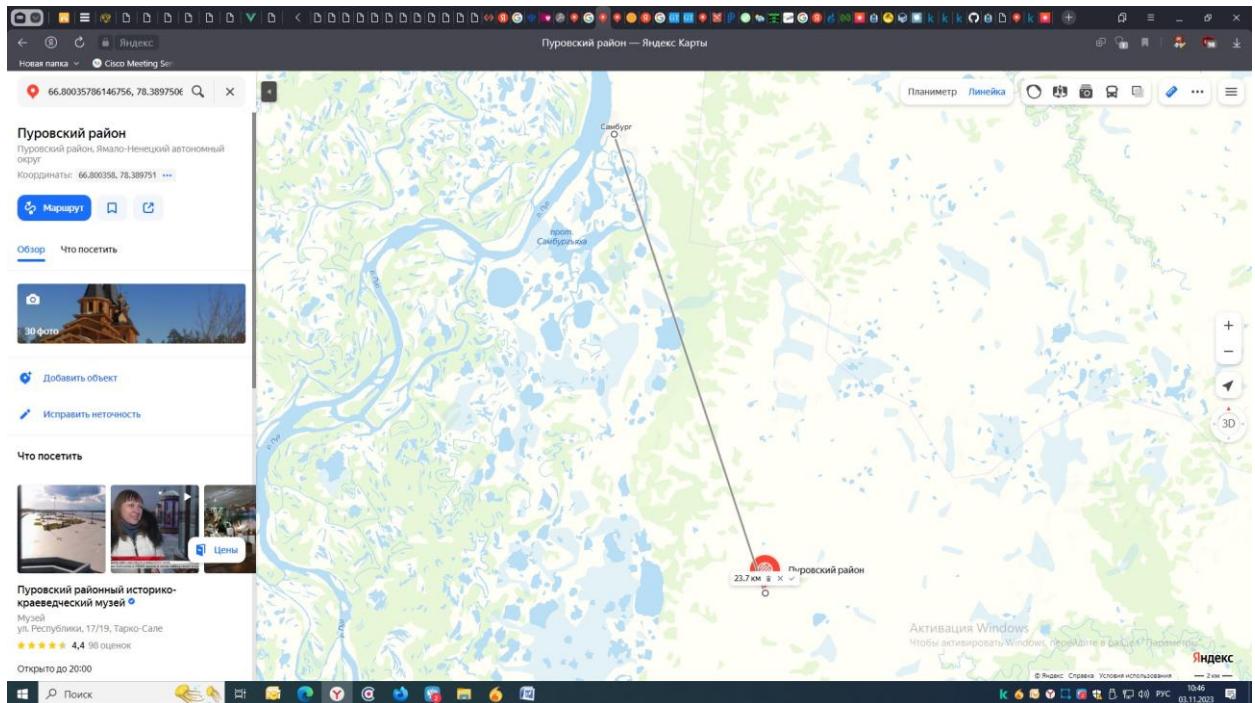
lng=78.38975066524623 (Долгота)

По координатам ищем скважину на Яндекс карте:

https://yandex.ru/maps/?ll=78.337713%2C66.894379&mode=search&sll=78.389751%2C66.800358&source=serp_navig&text=66.800358%2C78.389751&z=10.32



Находим близлежащее к скважине поселение - село Самбург



Село Самбург находится в 24 км от скважины № 807

Можем взять архив погоды Ямало-Ненецкий автономный округ – Пуровский район – село Самбург и распарсить погоду, получить датасет:

<https://pogoda1.ru/samburg/arkhiv/>

The screenshot shows the POGODA1.RU website interface for Pyrovsk. At the top, there's a navigation bar with links for 'Погода' (Weather), 'Карты' (Maps), 'Мир' (World), 'Новости' (News), 'О погоде' (About Weather), and 'Информеры' (Informers). A weather forecast for Pyrovsk is displayed with a temperature of -10°C and a wind direction of Самбург. Below the header, a banner reads 'Архив погоды в Пуровске, Пуровский район, Ямало-Ненецкий автономный округ - дневник погоды за 2017 - 2023гг.' A sidebar on the right contains sections for 'Избранное' (Favorites) and 'Новости погоды' (Weather News). The main content area shows a large empty space for the weather history, with a note: 'Для просмотра статистики погоды выберите диапазон.' (To view weather statistics, select a range.)

This screenshot shows the detailed weather archive for Pyrovsk in 2017. The page title is 'Погода в Пуровске в 2017 году'. It includes a brief description of the data: 'Точный прогноз погоды в Пуровске, Пуровский район, Ямало-Ненецкий автономный округ в 2017 году от Погода 1 - это информация об атмосферном давлении, направлении и силе ветра и других характеристиках погоды. Прогноз на 2017 год по г. Пуровск, Ямало-Ненецкий автономный округ предоставляется бесплатно. Хотите всегда быстро узнавать какая погода в вашем регионе? Добавьте его в избранное и прогноз всегда будет под рукой.' Below this, a table shows monthly weather summaries for 2017. The table has two rows: the first row shows 'Июнь' through 'Декабрь' with temperatures '+27° +2°', and the second row shows 'Июль' through 'Декабрь' with temperatures '+28° -8°', '+24° +4°', '+14° -6°', '+5° -11°', '+1° -32°', and '0° -30°'. To the right, there's a sidebar with 'Избранное' and 'Новости погоды' sections, and a note about adding cities to the favorites list.

Так как:

- архив погоды с. Самбург с 07.06.2017 года по настоящее время (ноябрь 2023);
- архив параметров работы нефтяной скважины (скважина №807) с 01.01.2013 по 18.01.2021.

Можем взять среднее между двумя датасетами 07.06.2017 по 18.01.2021, итого 1315 строки данных.

Итого:

- Есть данные о скважине №807 в размере 1315 строк и форматеxlsx (Excel) – скачиваем датасет как есть в локальное хранилище в виде файла, так как данные не обновляются;
- Есть данные о погоде с. Самбург в размере 1315 строк и форматеhtml (Web) – строим парсер и собираем данные с сайта.

На ум пришли 2 задачи:

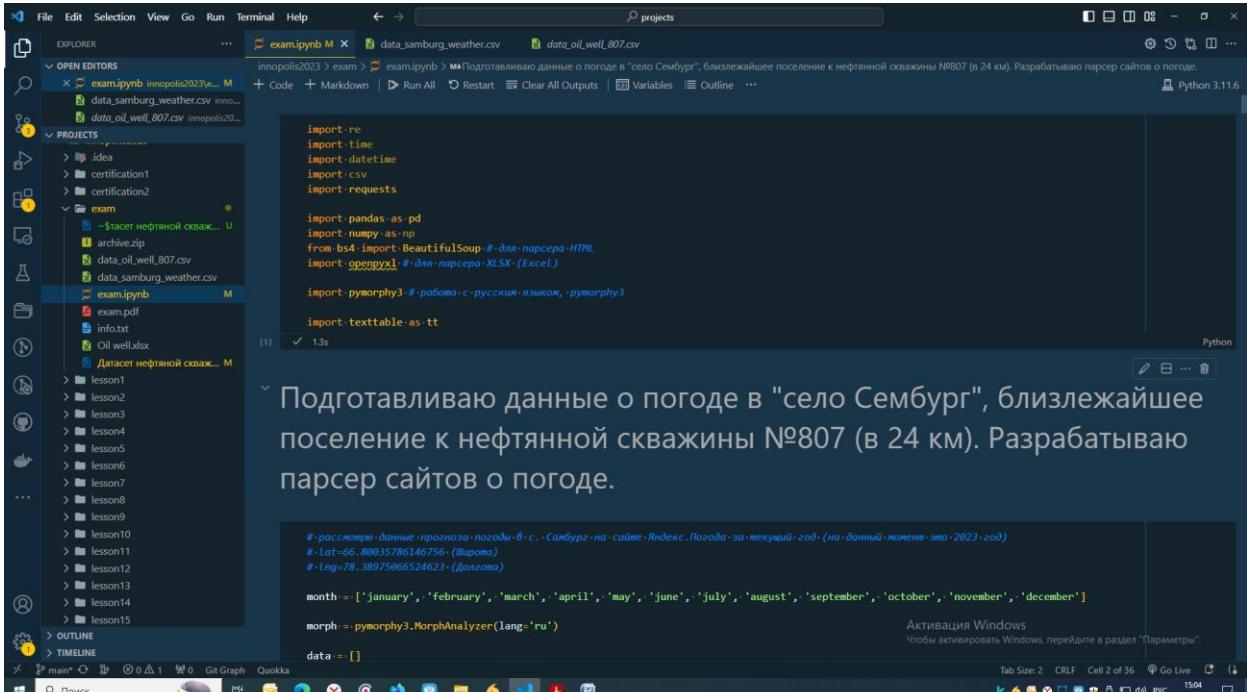
1. предсказывать погоду, влияющую на характеристики скважины №807
 - нам нужны характеристики скважины №807 в реальном времени (так как их нет, то можно синтетические данные писать / придумывать);
2. предсказывать характеристики скважины №807, относительно погодных условий – нам нужна погода в реальном времени с. Самбург (погода есть в реальном времени на сайте, можно парсить и обновлять данные).

Решением стало выбор 2 задачи:

- прогнозирование характеристик нефтяной скважины №807.

Этап 2. Подготовка данных

Подготавливаю данные о погоде в "село Сембург", близлежащее поселение к нефтяной скважине №807 (в 24 км). Разрабатываю парсер сайтов о погоде.



The screenshot shows a Jupyter Notebook interface with the following details:

- File Bar:** File, Edit, Selection, View, Go, Run, Terminal, Help.
- Project Bar:** projects
- Explorer:** Shows a file tree with a yellow highlighted folder named "exam". Inside "exam" are files like "archive.zip", "data_oil_well_807.csv", "data_samburg_weather.csv", and "exam.ipynb".
- Code Editor:** An open notebook cell titled "exam.ipynb" contains the following Python code:

```
import re
import time
import datetime
import csv
import requests

import pandas as pd
import numpy as np
from bs4 import BeautifulSoup # для парсера HTML
import openpyxl # для парсера XLSX (Excel)

import pymorphy3 # работа с русским языком, pymorphy3
import texttable as tt
```

- Text Area:** A large text block below the code editor states: "Подготавливаю данные о погоде в "село Сембург", близлежащее поселение к нефтяной скважине №807 (в 24 км). Разрабатываю парсер сайтов о погоде."
- Terminal:** A terminal window at the bottom shows command-line output related to weather data extraction.
- Bottom Bar:** Includes icons for search, refresh, and other Jupyter functions, along with system taskbar icons for browser, file explorer, and system status.

The screenshot shows a Jupyter Notebook interface with three open files: `exam.ipynb`, `data_samburg.weather.csv`, and `data_oil_well_807.csv`. The `exam.ipynb` file contains the following Python code:

```
#-проверка - осмотр данных - прогноза погоды в с. Самбург на сайте Яндекс.Погода за текущий год (на данный момент это 2023 год)
#-Lat=66.80035786146756 ( широта )
#-Lng=78.38975066524623 ( долгота )

month = ['january', 'february', 'march', 'april', 'may', 'june', 'july', 'august', 'september', 'october', 'november', 'december']

morph = pymorphy3.MorphAnalyzer(lang='ru')

data = []
data_i = 0
for i, item in enumerate(month):
    url = 'https://yandex.ru/pogoda/month/{}?lat={}&lon={}'.format(item, 66.80035786146756, 78.38975066524623)
    print(url)
    page = requests.get(url)
    bs = BeautifulSoup(page.text, 'html.parser')

    #информация о погоде в с. Самбурге за каждый месяц
    for val in enumerate(bs.find('table', {'class': 'climate-calendar'}).find_all('td', {'class': 'climate-calendar_cell'})):
        temp = []
        if val.find('div', {'class': 'climate-calendar-day_colorless_yes'}):
            continue
        temp.append(int(data_i))
        temp.append(int(val.find('div', {'class': 'climate-calendar-day_detailed-container-center'}).find_next('h6').text.split(',', 1)[0].split('.', 1)[0])))
        temp.append(str(morph.parse(val.find('div', {'class': 'climate-calendar-day_detailed-container-center'}).find_next('h6').text.split(',', 1)[0].split('.')[1])[0]))
        temp.append(int(val.find('div', {'class': 'temp_climate-calendar-day_detailed-basic-temp-day'}).find('span').text.replace('-', '.')))
        temp.append(int(val.find('div', {'class': 'temp_climate-calendar-day_detailed-basic-temp-night'}).find('span').text.replace('-', '.')))
        temp.append(int(val.find_all('td', {'class': 'climate-calendar-day_detailed-data-table-cell_climate-calendar-day_detailed-data-table-cell_value_yes'})))
        temp.append(float(val.find_all('td', {'class': 'climate-calendar-day_detailed-data-table-cell_climate-calendar-day_detailed-data-table-cell_value_yes'})))
        temp.append(str(val.find_all('td', {'class': 'climate-calendar-day_detailed-data-table-cell_climate-calendar-day_detailed-data-table-cell_value_yes'})))
        print(temp)
        data.append(temp)
        data_i += 1
```

The screenshot shows the same Jupyter Notebook environment after the code has been run. The output cell displays the generated data as a list of lists:

```
[0, 1, 'январь', 2023, 'вс', -21, -26, 760, 0.76, 5.3, '108']
[1, 2, 'январь', 2023, 'пн', -22, -22, 763, 0.79, 5.3, '3']
[2, 3, 'январь', 2023, 'вт', -21, -22, 764, 0.79, 5.8, '107']
[3, 4, 'январь', 2023, 'ср', -17, -20, 762, 0.81, 6.5, '107']
[4, 5, 'январь', 2023, 'чт', -19, -22, 764, 0.8, 6.0, '107']
[5, 6, 'январь', 2023, 'пт', -20, -22, 766, 0.79, 5.5, '107']
[6, 7, 'январь', 2023, 'сб', -19, -21, 766, 0.76, 4.8, '107']
[7, 8, 'январь', 2023, 'пн', -20, -22, 763, 0.76, 5.3, '108']
[8, 9, 'январь', 2023, 'пн', -22, -22, 761, 0.77, 5.5, '107']
[9, 10, 'январь', 2023, 'вт', -21, -24, 760, 0.77, 5.8, '107']
[10, 11, 'январь', 2023, 'ср', -24, -24, 762, 0.76, 5.1, '107']
[11, 12, 'январь', 2023, 'чт', -21, -21, 758, 0.77, 6.5, '107']
[12, 13, 'январь', 2023, 'пт', -19, -22, 756, 0.8, 6.4, '107']
[13, 14, 'январь', 2023, 'сб', -20, -20, 757, 0.79, 5.4, '107']
[14, 15, 'январь', 2023, 'пн', -17, 20, 759, 0.8, 4.9, '107']
[15, 16, 'январь', 2023, 'пн', -20, -21, 763, 0.8, 4.5, '107']
[16, 17, 'январь', 2023, 'вт', -19, -20, 764, 0.79, 4.5, '107']
[17, 18, 'январь', 2023, 'ср', -19, -22, 764, 0.77, 5.3, '107']
[18, 19, 'январь', 2023, 'чт', -19, -19, 764, 0.8, 6.0, '107']
[19, 20, 'январь', 2023, 'пт', -16, -17, 762, 0.83, 6.9, '107']
[20, 21, 'январь', 2023, 'сб', -17, -18, 760, 0.81, 7.0, '107']
[21, 22, 'январь', 2023, 'пн', -16, -17, 760, 0.82, 6.5, '107']
[22, 23, 'январь', 2023, 'вт', -15, -16, 760, 0.82, 5.5, '107']
[23, 24, 'январь', 2023, 'вт', -15, -21, 760, 0.82, 5.5, '107']
...
[361, 28, 'декабрь', 2023, 'чт', -18, -20, 756, 0.81, 5.0, '107']
[362, 29, 'декабрь', 2023, 'пт', -19, -21, 759, 0.81, 5.5, '107']
[363, 30, 'декабрь', 2023, 'сб', -19, -20, 760, 0.8, 5.8, '107']
[364, 31, 'декабрь', 2023, 'вс', -19, -20, 760, 0.8, 5.8, '107']
```

File Edit Selection View Go Run Terminal Help

projects

EXPLORER OPEN EDITORS

exam.ipynb innopolis2023... M data_samburg_weather.csv data_oil_well_807.csv

exam.ipynb innopolis2023... M data_samburg_weather.csv innopolis2023... M data_oil_well_807.csv innopolis2023... M

PROJECTS

- idea
- certification1
- certification2
- exam
 - Старая нефтяная скважина...
 - archive.zip
 - data_oil_well_807.csv
 - data_samburg_weather.csv
 - exam.ipynb
 - exam.pdf
 - info.txt
 - Oil well.xlsx
 - Директория нефтяной скважин...
- lesson1
- lesson2
- lesson3
- lesson4
- lesson5
- lesson6
- lesson7
- lesson8
- lesson9
- lesson10
- lesson11
- lesson12
- lesson13
- lesson14
- lesson15

OUTLINE TIMELINE

exam.ipynb M data_samburg_weather.csv data_oil_well_807.csv

m > exam.ipynb > M Подготавливаю данные о погоде в "селе Сембигур", близлежащее поселение к нефтяной скважине №9807 (в 24 км). Разрабатываю парсер сайтов о погоде. # создано объект Texttable

+ Code + Markdown | Run All | Restart | Clear All Outputs | Variables | Outline ...

tab = tt.TextTable()

установил формат столбцов таблицы

tab.set_cols_align(['ц', 'ц', 'ц'])

tab.header(['id', 'День', 'Месяц', 'Год', 'День недели', 'Температура днем', 'Температура ночь', 'Давление (мм рт. ст.)', 'Влажность (%)', 'Скорость ветра'])

преобразуем Dataframe в список списков (двоичный список)

data_list = df.values.tolist()

добавляем данные в таблицу

for row in data_list:
 tab.add_row(row)

получаем отформатированную таблицу в виде строки

table_string = tab.draw()

выводим таблицу на экран

print(table_string)

[37]

| id | День | Месяц | Год | День недели | Температура днем | Температура ночь | Давление (мм рт. ст.) | Влажность (%) | Скорость ветра | Напр. |
|----|------|--------|------|-------------|------------------|------------------|-----------------------|---------------|----------------|-------|
| 0 | 1 | январь | 2023 | вс | -21 | -26 | 760 | 0.76 | 5.30 | 100 |
| 1 | 2 | январь | 2023 | пн | -22 | -22 | 763 | 0.79 | 5.30 | 3 |
| 2 | 3 | январь | 2023 | вт | -21 | -22 | 764 | 0.79 | 5.88 | 0 |
| 3 | 4 | январь | 2023 | ср | -17 | -20 | 762 | 0.81 | 6.58 | 0 |

Активация Windows
Чтобы активировать Windows, перейдите в раздел "Параметры".

Tab Size: 2 | CRLF | Cell 4 of 36 | Go Live | Total Lines: 19 | Prettier | 1505 08.11.2023

Как видно я получил данные с Яндекс.Погода за 2023 года, мне нужно больше данных. У Яндекс.Погода нет архивных данных, поэтому буду использовать другой сайт с архивными данными.

File Edit Selection View Go Run Terminal Help

projects

EXPLORER OPEN EDITORS

exam.ipynb data_samburg_weather.csv data_oil_well_807.csv

innopolis2023 > exam > exam.ipynb > **М** Как видно я получил данные с Яндекс.Погода за 2023 года, мне нужно больше данных. У Яндекс.Погода нет архивных данных, поэтому буду использовать другой сайт ...

+ Code + Markdown | Run All | Restart | Clear All Outputs | Variables | Outline ...

Python 3.11.6

PROJECTS

exam

- Stacer нефтяной скважин...
- archive.zip
- data_oil_well_807.csv
- data_samburg_weather.csv
- exam.ipynb
- exam.pdf
- info.txt
- Oil well.xlsx
- Документы нефтяной скважин...

lesson1 lesson2 lesson3 lesson4 lesson5 lesson6 lesson7 lesson8 lesson9 lesson10 lesson11 lesson12 lesson13 lesson14 lesson15

OUTLINE

TIMELINE

- рассмотрим данные прогноза погоды в с. Сабург - с сайта pogoda1 - так как в нем есть архивные данные

```
url = "https://pogoda1.ru/samburg/archiv/"
page = requests.get(url)
bs = BeautifulSoup(page.text, 'html.parser')

# - формируем список диапазона дат, где имеются архивные данные
year = bs.find('select', {'class': 'select-archive-year'}).find_all('option')
month = bs.find('select', {'class': 'select-archive-month'}).find_all('option')

# - перевод месяцев с русского на английский язык
def month_translate(month):
    month = month.lower()
    if month == "январь":
        return "january"
    elif month == "февраль":
        return "february"
    elif month == "март":
        return "march"
    elif month == "апрель":
        return "april"
    elif month == "май":
        return "may"
    elif month == "июнь":
        return "june"
    elif month == "июль":
        return "july"
```

```
base_month = []
for i, val_year in enumerate(year):
    for j, val_month in enumerate(month):
        temp_0 = []
        if j > 0:
            temp_0.append(int(val_year.text))
            temp_0.append(str(month_translate(month - val_month.text)).lower())
            base_month.append(temp_0)

# Формирую список URL-адресов с данными
url_parse = 'https://pogoda1.ru/samburg/{month}-{year}/'

for i, val in enumerate(base_month):
    parse = url_parse.format(month=val[1], year=val[0])
    base_month[i].append(parse)

df = pd.DataFrame(base_month, columns=['Год', 'Месяц', 'URL'])
df
```

Активация Windows
Чтобы активировать Windows, перейдите в раздел "Параметры".

Tab Size: 2 CRLF Cell 5 of 36 Go Live 1507 08.11.2023

```
# Пакет с парсингом ссылок с данными
base_day = []
for i, val in enumerate(base_month):
    bs = BeautifulSoup(requests.get(val[2]).text, 'html.parser')
    if (bs.find('div', {'class': 'month-calendar__calendar'})):
        print(val[2])
        for j, item in enumerate(bs.find_all('a', {'class': 'calendar-item'})):
            temp = []
            if not (item.find('span', {'class': 'no-data'})):
                temp.append(int(month_number(val[1])))
                temp.append(int(item.find('span', {'month-calendar-day'}).text))
                temp.append('https://pogoda1.ru' + str(item['href']))
                print(temp)
                base_day.append(temp)

[2017, 6, 7, 'https://pogoda1.ru/samburg/07-06-2017/']
[2017, 6, 8, 'https://pogoda1.ru/samburg/08-06-2017/']
[2017, 6, 9, 'https://pogoda1.ru/samburg/09-06-2017/']
[2017, 6, 10, 'https://pogoda1.ru/samburg/10-06-2017/']
[2017, 6, 11, 'https://pogoda1.ru/samburg/11-06-2017/']
[2017, 6, 12, 'https://pogoda1.ru/samburg/12-06-2017/']
[2017, 6, 13, 'https://pogoda1.ru/samburg/13-06-2017/']
[2017, 6, 14, 'https://pogoda1.ru/samburg/14-06-2017/']
[2017, 6, 15, 'https://pogoda1.ru/samburg/15-06-2017/']
[2017, 6, 16, 'https://pogoda1.ru/samburg/16-06-2017/']
[2017, 6, 17, 'https://pogoda1.ru/samburg/17-06-2017/']
[2017, 6, 18, 'https://pogoda1.ru/samburg/18-06-2017/']
[2017, 6, 19, 'https://pogoda1.ru/samburg/19-06-2017/']
[2017, 6, 20, 'https://pogoda1.ru/samburg/20-06-2017/']
[2017, 6, 21, 'https://pogoda1.ru/samburg/21-06-2017/']
[2017, 6, 22, 'https://pogoda1.ru/samburg/22-06-2017/']
```

Активация Windows
Чтобы активировать Windows, перейдите в раздел "Параметры".

Tab Size: 2 CRLF Cell 5 of 36 Go Live 1507 08.11.2023

The screenshot shows a Jupyter Notebook interface with two open files: `exam.ipynb` and `data_samburg.weather.csv`. The `exam.ipynb` file contains Python code for web scraping. The code uses BeautifulSoup to parse HTML from a URL. It iterates through days, finds specific weather icons and their descriptions, and extracts numerical values for wind amount and precipitation. The code includes comments explaining the logic for handling different weather types like 'асно' (cloudy) and 'пасмурно' (overcast). The output cell shows the raw HTML being parsed.

```
# Парсер погоды
data = []
for i, val in enumerate(base_day):
    if i < 1593:
        continue
    temp = []
    bs = BeautifulSoup(requests.get(val[3]).text, 'html.parser')
    if not (bs.find('div', {'class': 'panel-heading'}).text == '404. Страница не найдена'):
        # print(val[3])
        temp.append(i)
        temp.append(val[0])
        temp.append(val[1])
        temp.append(val[2])
        temp.append(str(bs.find('img', {'class': 'weather-now-col-weather-now-col-main'})).find('span', {'class': 'wind-amount'}))
        if (bs.find('div', {'class': 'weather-now-col-weather-now-col-main'})).find('span', {'class': 'wind-amount'}):
            temp.append(int(bs.find_all('span', {'class': 'wind-amount'})[0].text.split(',')[1][0]).lower())
            temp.append(int(bs.find_all('span', {'class': 'wind-amount'})[0].text.split(',')[1][1].split(',', 2)[1]))
        else:
            for j, item in enumerate(bs.find_all('div', {'row-forecast-time-of-day'})):
                if not (str(item.find('div', {'class': 'cell-forecast-wind'})).find('span', {'class': 'wind'}).text == 'нет'):
                    temp.append(str(item.find('div', {'class': 'cell-forecast-wind'})).find('img', {'class': 'icon-wind'}))['title'].split(',')[1][0].lower()
                    temp.append(int(item.find('div', {'class': 'cell-forecast-wind'})).find('span', {'class': 'wind-amount'}))
                else:
                    temp.append('')
                    temp.append('')
                    break
        temp.append(int(bs.find_all('div', {'class': 'weather-now-info'})[0].find('span', {'class': 'value'}).text.split(',')[1][0]))
        temp.append(int(bs.find_all('div', {'class': 'weather-now-info'})[1].find('span', {'class': 'value'}).text.split(',')[1][0]))
        temp.append(float(bs.find_all('div', {'class': 'weather-now-info'})[2].find('span', {'class': 'value'}).text.split(',')[1][0]))
        temp.append(str(bs.find_all('div', {'class': 'weather-now-info'})[6].find('span', {'class': 'value'}).text.split(',')[1][0]).lower())
        # temp.append(int(bs.find_all('div', {'row-forecast-time-of-day'})[6].find('span', {'class': 'value'}).text.split(',')[1][0]))
        if not (str(bs.find_all('div', {'row-forecast-time-of-day'})[2]).find('div', {'class': 'cell-forecast-prec-opened'})):
            if (str(bs.find_all('div', {'row-forecast-time-of-day'})[2]).find('div', {'class': 'cell-forecast-prec'})).text == 'без осадков':
                temp.append(0)
            else:
                temp.append(float(bs.find_all('div', {'row-forecast-time-of-day'})[2].find('div', {'class': 'cell-forecast-prec'}).text.split(',')[1][0]))
        else:
```

The screenshot shows the same Jupyter Notebook interface after running the code. The output cell displays a large list of tuples, each containing a date (e.g., 0, 2017), day of the month (e.g., 6), hour (e.g., 7), weather condition (e.g., 'пасмурно', 'северо-западный'), and various weather parameters (e.g., wind speed, precipitation). The list continues for all 1593 days in the dataset.

```
[0, 2017, 6, 7, 'пасмурно', 'северо-западный', 4, 764, 56, 10.0, 'растущая', 0, 9, 12, 'https://pogoda1.ru/samburg/07-06-2017/']
[1, 2017, 6, 8, 'асно', 'северо-западный', 3, 762, 63, 10.0, 'растущая', 0, 9, 13, 'https://pogoda1.ru/samburg/08-06-2017/']
[2, 2017, 6, 9, 'малоблочно', 'северный', 3, 756, 51, 10.0, 'полонуние', 0, 12, 15, 'https://pogoda1.ru/samburg/09-06-2017/']
[3, 2017, 6, 10, 'пасмурно', 'северо-западный', 8, 752, 60, 10.0, 'убывающая', 0, 10, 10, 'https://pogoda1.ru/samburg/10-06-2017/']
[4, 2017, 6, 11, 'пасмурно', 'западный', 6, 756, 50, 10.0, 'убывающая', 0, 4, 2, 'https://pogoda1.ru/samburg/11-06-2017/']
[5, 2017, 6, 12, 'пасмурно', 'южный', 6, 758, 70, 10.0, 'убывающая', 0, 5, 12, 'https://pogoda1.ru/samburg/12-06-2017/']
[6, 2017, 6, 13, 'пасмурно', 'юго-восточный', 2, 758, 100, 10.0, 'убывающая', 1, 1, 5, 9, 'https://pogoda1.ru/samburg/13-06-2017/']
[7, 2017, 6, 14, 'пасмурно', 'южный', 9, 759, 65, 10.0, 'убывающая', 0, 6, 15, 'https://pogoda1.ru/samburg/14-06-2017/']
[8, 2017, 6, 15, 'пасмурно', 'южный', 7, 759, 55, 10.0, 'убывающая', 0, 15, 22, 'https://pogoda1.ru/samburg/15-06-2017/']
[9, 2017, 6, 16, 'малоблочно', 'юго-западный', 8, 761, 82, 10.0, 'убывающая', 0, 15, 17, 'https://pogoda1.ru/samburg/16-06-2017/']
[10, 2017, 6, 17, 'пасмурно', 'восточный', 6, 769, 49, 10.0, 'убывающая', 0, 4, 11, 'https://pogoda1.ru/samburg/17-06-2017/']
[11, 2017, 6, 18, 'асно', 'южный', 5, 758, 100, 10.0, 'убывающая', 0, 15, 22, 'https://pogoda1.ru/samburg/18-06-2017/']
[12, 2017, 6, 19, 'дожь', 'северо-восточный', 6, 755, 96, 10.0, 'убывающая', 0, 15, 19, 'https://pogoda1.ru/samburg/19-06-2017/']
[13, 2017, 6, 20, 'пасмурно', 'южный', 8, 752, 70, 10.0, 'убывающая', 0, 16, 20, 'https://pogoda1.ru/samburg/20-06-2017/']
[14, 2017, 6, 21, 'пасмурно', 'южный', 7, 751, 74, 10.0, 'убывающая', 0, 2, 15, 18, 'https://pogoda1.ru/samburg/21-06-2017/']
[15, 2017, 6, 22, 'пасмурно', 'западный', 5, 755, 77, 10.0, 'убывающая', 0, 2, 15, 17, 'https://pogoda1.ru/samburg/22-06-2017/']
[16, 2017, 6, 23, 'малоблочно', 'южный', 2, 761, 80, 10.0, 'убывающая', 0, 13, 19, 'https://pogoda1.ru/samburg/23-06-2017/']
[17, 2017, 6, 24, 'малоблочно', 'южный', 5, 758, 60, 10.0, 'новолуние', 0, 17, 25, 'https://pogoda1.ru/samburg/24-06-2017/']
[18, 2017, 6, 25, 'пасмурно', 'южный', 7, 754, 68, 10.0, 'растущая', 0, 19, 23, 'https://pogoda1.ru/samburg/25-06-2017/']
[19, 2017, 6, 26, 'малоблочно', 'восточный', 3, 762, 67, 10.0, 'растущая', 0, 10, 18, 'https://pogoda1.ru/samburg/26-06-2017/']
[20, 2017, 6, 27, 'пасмурно', 'восточный', 7, 757, 59, 10.0, 'растущая', 0, 16, 19, 'https://pogoda1.ru/samburg/27-06-2017/']
[21, 2017, 6, 28, 'пасмурно', 'южный', 9, 751, 92, 10.0, 'растущая', 0, 4, 14, 18, 'https://pogoda1.ru/samburg/28-06-2017/']
[22, 2017, 6, 29, 'пасмурно', 'юго-западный', 8, 755, 71, 10.0, 'растущая', 0, 1, 12, 17, 'https://pogoda1.ru/samburg/29-06-2017/']
[23, 2017, 6, 30, 'пасмурно', 'западный', 6, 758, 73, 10.0, 'растущая', 0, 14, 15, 'https://pogoda1.ru/samburg/30-06-2017/']
[24, 2017, 7, 1, 'асно', 'северный', 4, 764, 63, 10.0, 'растущая', 0, 10, 14, 'https://pogoda1.ru/samburg/01-07-2017/']
```

File Edit Selection View Go Run Terminal Help projects

OPEN EDITORS exam.ipynb M data_samburg.weather.csv data_oil.well_807.csv

PROJECTS

- exam.ipynb inneapolis2023.ipynb
- data_samburg_weather.csv inno...
- data_oil.well_807.csv inno...
- exam
- idea
- certification1
- certification2
- lesson1
- lesson2
- lesson3
- lesson4
- lesson5
- lesson6
- lesson7
- lesson8
- lesson9
- lesson10
- lesson11
- lesson12
- lesson13
- lesson14
- lesson15
- archive.zip
- data_oil.well_807.csv
- data_samburg.weather.csv
- exam.ipynb M
- exam.pdf
- info.txt
- Oil well.xlsx
- Дагасет нефтяной скаж...

Code + Markdown | Run All | Restart | Clear All Outputs | Variables | Outline ... Python 3.11.6

```
[18, 2017, 6, 25, 'пасмурно', 'южный', 7, 754, 68, 10.0, 'растущая', 0, 19, 23, 'https://pogoda1.ru/samburg/25-06-2017/']
[19, 2017, 6, 26, 'малооблачно', 'восточный', 3, 762, 67, 10.0, 'растущая', 0, 10, 18, 'https://pogoda1.ru/samburg/26-06-2017/']
[20, 2017, 6, 27, 'пасмурно', 'восточный', 7, 757, 59, 10.0, 'растущая', 0, 16, 19, 'https://pogoda1.ru/samburg/27-06-2017/']
[21, 2017, 6, 28, 'пасмурно', 'южный', 9, 751, 92, 10.0, 'растущая', 0.4, 14, 18, 'https://pogoda1.ru/samburg/28-06-2017/']
[22, 2017, 6, 29, 'пасмурно', 'юго-западный', 8, 755, 71, 10.0, 'растущая', 0.1, 12, 17, 'https://pogoda1.ru/samburg/29-06-2017/']
[23, 2017, 6, 30, 'пасмурно', 'западный', 6, 758, 73, 10.0, 'растущая', 0, 14, 15, 'https://pogoda1.ru/samburg/30-06-2017/']
[24, 2017, 7, 1, 'сено', 'северный', 4, 764, 63, 10.0, 'растущая', 0, 10, 14, 'https://pogoda1.ru/samburg/01-07-2017/']
...
[2346, 2023, 11, 16, 'снег', 'юго-западный', 5, 762, 96, 10.0, 'растущая', 1.0, -3, -4, 'https://pogoda1.ru/samburg/16-11-2023/']
[2347, 2023, 11, 17, 'снег', 'северный', 8, 759, 93, 10.0, 'растущая', 1.5, -19, -4, 'https://pogoda1.ru/samburg/17-11-2023/']
[2348, 2023, 11, 18, 'пасмурно', 'северный', 7, 765, 89, 10.0, 'растущая', 0, -19, -21, 'https://pogoda1.ru/samburg/18-11-2023/']
[2349, 2023, 11, 19, 'пасмурно', 'северо-западный', 5, 774, 93, 10.0, 'растущая', 0, -20, -16, 'https://pogoda1.ru/samburg/19-11-2023/']

Output is truncated. View as a scrollable element or open in a text editor. Adjust cell output settings...
```

Aктивация Windows чтобы активировать Windows, перейдите в раздел "Параметры".

Tab Size: 2 CRLF Cell 5 of 36 Go Live 15:07 06.11.2023

File Edit Selection View Go Run Terminal Help projects

OPEN EDITORS exam.ipynb M data_samburg.weather.csv data_oil.well_807.csv

PROJECTS

- exam.ipynb inneapolis2023.ipynb
- data_samburg_weather.csv inno...
- data_oil.well_807.csv inno...
- exam
- idea
- certification1
- certification2
- lesson1
- lesson2
- lesson3
- lesson4
- lesson5
- lesson6
- lesson7
- lesson8
- lesson9
- lesson10
- lesson11
- lesson12
- lesson13
- lesson14
- lesson15
- archive.zip
- data_oil.well_807.csv
- data_samburg.weather.csv
- exam.ipynb M
- exam.pdf
- info.txt
- Oil well.xlsx
- Дагасет нефтяной скаж...
- lesson1
- lesson2
- lesson3
- lesson4
- lesson5
- lesson6
- lesson7
- lesson8
- lesson9
- lesson10
- lesson11
- lesson12
- lesson13
- lesson14
- lesson15
- outline
- Timeline

Code + Markdown | Run All | Restart | Clear All Outputs | Variables | Outline ... Python 3.11.6

```
#-Формирование DataFrame
df = pd.DataFrame(data, columns=[

    'id',
    'год',
    'месяц',
    'месяц',
    'день',
    'погодное условие',
    'направление ветра',
    'скорость ветра (м/с)',
    'давление (мм рт. ст.)',
    'влажность (%)',
    'видимость (мм)',
    'луна',
    'осадки (мм)',
    'температура днем',
    'температура ночью',
    'url'
])
# df.drop(columns=['id'], inplace=True)

#-Сохранение данных в csv
df.to_csv('data_samburg_weather.csv', sep=',', encoding='utf-8', index=False)
df
```

| год | месяц | день | погодное условие | направление ветра | скорость ветра (м/с) | давление (мм рт. ст.) | влажность (%) | видимость (мм) | луна | осадки (мм) | температура днем | температура ночью | url | |
|------|-------|------|------------------|-------------------|----------------------|-----------------------|---------------|----------------|------|-------------|------------------|-------------------|-----|--------------------|
| 0 | 2017 | 6 | 7 | пасмурно | северо-западный | 4 | 764 | 56 | 10.0 | растущая | 0.0 | 9 | 13 | https://pogoda1... |
| 1 | 2017 | 6 | 8 | ясно | северо-западный | 3 | 762 | 63 | 10.0 | растущая | 0.0 | 9 | 13 | https://pogoda1... |
| 2 | 2017 | 6 | 9 | малооблачно | северный | 3 | 756 | 51 | 10.0 | полонуние | 0.0 | 12 | 15 | https://pogoda1... |
| 3 | 2017 | 6 | 10 | пасмурно | северо-западный | 8 | 752 | 60 | 10.0 | убывающая | 0.0 | 10 | 10 | https://pogoda1... |
| 4 | 2017 | 6 | 11 | пасмурно | западный | 6 | 756 | 50 | 10.0 | убывающая | 0.0 | 4 | 2 | https://pogoda1... |
| - | - | - | - | - | - | - | - | - | - | - | - | - | - | - |
| 2345 | 2023 | 11 | 15 | снег | юго-западный | 10 | 763 | 92 | 10.0 | растущая | 0.9 | -4 | -8 | https://pogoda1... |
| 2346 | 2023 | 11 | 16 | снег | юго-западный | 5 | 762 | 96 | 10.0 | растущая | 1.0 | -3 | -4 | https://pogoda1... |
| 2347 | 2023 | 11 | 17 | снег | северный | 8 | 759 | 93 | 10.0 | растущая | 1.5 | -19 | -4 | https://pogoda1... |
| 2348 | 2023 | 11 | 18 | пасмурно | северный | 7 | 765 | 89 | 10.0 | растущая | 0.0 | -19 | -21 | https://pogoda1... |
| 2349 | 2023 | 11 | 19 | пасмурно | северо-западный | 5 | 774 | 93 | 10.0 | растущая | 0.0 | -20 | -16 | https://pogoda1... |

Aктивация Windows чтобы активировать Windows, перейдите в раздел "Параметры".

Tab Size: 2 CRLF Cell 5 of 36 Go Live 15:07 06.11.2023

Необходимые данные получил в размере 2350 строк. Сохранил данные в файл и загрузил к себе в Git репозиторий. Дальше идет подготовка данных, а именно просмотр пустот и заполнения средними или наиболее встречающимися значениями, просмотр типов данных.

```
#-Загрузка файла из Git моего репозитория в Pandas
#-data_samburg_weather = pd.read_csv("https://raw.githubusercontent.com/SotEG/innopolis2023/main/exam/data_samburg_weather.csv", sep=',', index_col=False)
data_samburg_weather = pd.read_csv("https://raw.githubusercontent.com/SotEG/innopolis2023/main/exam/data_samburg_weather.csv", sep=',', index_col=False)
```

| | id | год | месяц | день | погодное_условие | направление_ветра | скорость_ветра_(м/с) | давление_(мм рт. ст.) | влажность_(%) | видимость_(мм) | луна | осадки_(мм) | температура_днем | температура_ночью | URL |
|---|----|------|-------|------|------------------|-------------------|----------------------|-----------------------|---------------|----------------|-------------------|-------------|------------------|-------------------|--------------|
| 0 | 0 | 2017 | 6 | 7 | пасмурно | северо-западный | 4.0 | 764 | 56 | 10.0 | растущая | 0.0 | 9 | 13 | https://p... |
| 1 | 1 | 2017 | 6 | 8 | ясно | северо-западный | 3.0 | 762 | 63 | 10.0 | растущая | 0.0 | 9 | 13 | https://p... |
| 2 | 2 | 2017 | 6 | 9 | малооблачно | северный | 3.0 | 756 | 51 | 10.0 | полноуние | 0.0 | 12 | 15 | https://p... |
| 3 | 3 | 2017 | 6 | 10 | пасмурно | северо-западный | 8.0 | 752 | 60 | 10.0 | убывающая | 0.0 | 10 | 10 | https://p... |
| 4 | 4 | 2017 | 6 | 11 | пасмурно | западный | 6.0 | 756 | 50 | 10.0 | Активация Windows | 0.0 | 4 | 2 | https://p... |

```
#-Размер данных - количество строк, -колонок
data_samburg_weather.shape
```

```
(2350, 15)
```

```
#-Просмотр типов данных в DataFrame
print(data_samburg_weather.info())
```

| | Column | Non-Null Count | Dtype |
|---|-----------------------|----------------|------------------|
| 0 | Unnamed: 0 | 2350 | non-null int64 |
| 1 | год | 2350 | non-null int64 |
| 2 | месяц | 2350 | non-null int64 |
| 3 | день | 2350 | non-null int64 |
| 4 | погодное_условие | 2350 | non-null object |
| 5 | направление_ветра | 2348 | non-null object |
| 6 | скорость_ветра_(м/с) | 2348 | non-null float64 |
| 7 | давление_(мм рт. ст.) | 2350 | non-null int64 |
| 8 | влажность_(%) | 2350 | non-null int64 |
| 9 | видимость_(мм) | 2350 | non-null float64 |

The screenshot shows a Jupyter Notebook interface with several tabs open. The active tab is 'exam.ipynb' which contains the following code:

```
#-Проверка - данных
#-Количество - пустых - значек
data_samburg_weather.isnull().sum()
#-Количество - неопределенные - значений - (неправильно - считанные)
data_samburg_weather.isna().sum()
```

The output pane displays the results of the executed code:

| Название | Значение |
|-----------------------|----------|
| год | 0 |
| месяц | 0 |
| день | 0 |
| погодное условие | 0 |
| направление ветра | 2 |
| скорость ветра (м/с) | 2 |
| давление (мм рт. ст.) | 0 |
| влажность (%) | 0 |
| видимость (мм) | 0 |
| луна | 0 |
| осадки (мм) | 0 |
| температура днем | 0 |
| температура ночью | 0 |
| url | 0 |

Below the table, the status bar indicates 'Ln 2, Col 27' and 'Python 3.11.6'.

The screenshot shows a Jupyter Notebook interface with several tabs open. The active tab is 'exam.ipynb' which contains the following code:

```
#-Количество - неопределенные - значений - (неправильно - считанные)
data_samburg_weather.isna().sum()
#-Количество - пустых - строк
(data_samburg_weather == "").sum()
```

The output pane displays the results of the executed code:

| Название | Значение |
|-----------------------|----------|
| год | 0 |
| месяц | 0 |
| день | 0 |
| погодное условие | 0 |
| направление ветра | 0 |
| скорость ветра (м/с) | 0 |
| давление (мм рт. ст.) | 0 |
| влажность (%) | 0 |
| видимость (мм) | 0 |
| луна | 0 |
| осадки (мм) | 0 |
| температура днем | 0 |
| температура ночью | 0 |
| url | 0 |

Below the table, the status bar indicates 'Ln 2, Col 22' and 'Python 3.11.6'.

The screenshot shows a Jupyter Notebook interface with the following code in the cell:

```
# Заполнение пустых значений -- наиболее встречающимся классом
# df['направление ветра'] = df['направление ветра'].replace('', str(df['направление ветра'].value_counts().index[0]))
data_samburg_weather['направление ветра'].fillna(str(data_samburg_weather['направление ветра'].value_counts().index[0]), inplace=True)

# Заполнение пустых значений -- наиболее распространенного значения
# data_samburg_weather['скорость ветра (м/с)'] = data_samburg_weather['скорость ветра (м/с)'].replace('', float(data_samburg_weather['скорость ветра (м/с)'].value_counts().idxmax()))
data_samburg_weather['скорость ветра (м/с)'].fillna(float(data_samburg_weather['скорость ветра (м/с)'].value_counts().idxmax()), inplace=True)

data_samburg_weather = data_samburg_weather.astype({'скорость ветра (м/с)': 'float64'})
```

Output:

```
✓ 0.0s
```

Code cell 122:

```
# Просмотр типов данных & датасета
print(data_samburg_weather.info())
```

Output:

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2350 entries, 0 to 2349
Data columns (total 15 columns):
 #   Column      Non-Null Count  Dtype  
--- 
 0   id          2350 non-null   int64  
 1   год         2350 non-null   int64  
 2   месяц       2350 non-null   int64  
 3   день         2350 non-null   int64  
 4   погодное условие  2350 non-null   object  
 5   направление ветра (м/с) 2350 non-null   object  
 6   скорость ветра (м/с) 2350 non-null   float64 
 7   давление (мм рт. ст.) 2350 non-null   int64  
 8   влажность (%)    2350 non-null   int64  
 9   видимость (м)    2350 non-null   float64 
 10  луна        2350 non-null   object  
 11  осадки (мм)   2350 non-null   float64 
```

Code cell 132:

```
Активация Windows
чтобы активировать Windows, перейдите в раздел "Параметры".
```

Bottom status bar: Ln 2, Col 22 Tab Size: 2 CRLF Cell 10 of 36 Go Live 15:09 06.11.2023

The screenshot shows a Jupyter Notebook interface with the following code in the cell:

```
13 температура ночные 2350 non-null int64
14 url      2350 non-null object
memroy usage: 275.5+ KB
None
```

Code cell 133:

```
#-Побортная проверка данных
```

Code cell 134:

```
#-Количество пустых ячеек
data_samburg_weather.isnull().sum()
```

Output:

```
... id          0
год         0
месяц       0
день         0
погодное условие  0
направление ветра  0
скорость ветра (м/с) 0
давление (мм рт. ст.) 0
влажность (%)    0
видимость (м)    0
луна        0
осадки (мм)   0
температура днем   0
температура ночные 0
url         0
dtype: int64
```

Code cell 144:

```
#-Количество неопределенных значений (неправильно считанные)
data_samburg_weather.isna().sum()
```

Output:

```
... id          0
год         0

```

Code cell 145:

```
Активация Windows
чтобы активировать Windows, перейдите в раздел "Параметры".
```

Bottom status bar: Ln 4, Col 21 Tab Size: 2 CRLF Cell 10 of 36 Go Live 15:09 06.11.2023

File Edit Selection View Go Run Terminal Help projects

OPEN EDITORS exam.ipynb M x data_samburg.weather.csv data_oil_well_807.csv

PROJECTS

exam.ipynb M

```
год 0
месяц 0
день 0
погодное условие 0
направление ветра 0
скорость ветра (м/с) 0
давление (мм рт. ст.) 0
влажность (%) 0
видимость (мм) 0
луна 0
осадки (мм) 0
температура днем 0
температура ночью 0
url 0
dtype: int64
```

Количество пустых строк
(data_samburg_weather == '').sum()

```
[143] ✓ 0.0s
```

... Unnamed: 0 0
год 0
месяц 0
день 0
погодное условие 0
направление ветра 0
скорость ветра (м/с) 0
давление (мм рт. ст.) 0
влажность (%) 0
видимость (мм) 0
луна 0
осадки (мм) 0
температура днем 0
температура ночью 0
url 0
dtype: int64

Активация Windows
чтобы активировать Windows, перейдите в раздел "Параметры".

Ln 2, Col 21 (20 selected) Tab Size: 2 CRLF Cell 10 of 36 ⌂ Go Live ⌂ 15:09 06.11.2023

File Edit Selection View Go Run Terminal Help projects

OPEN EDITORS exam.ipynb M x data_samburg.weather.csv data_oil_well_807.csv

PROJECTS

exam.ipynb M

```
луна 0
осадки (мм) 0
температура днем 0
температура ночью 0
url 0
dtype: int64
```

Описательная статистика
data_samburg_weather.describe(include='all', percentiles=[0.1, 0.25, 0.5, 0.75, 0.9]).T

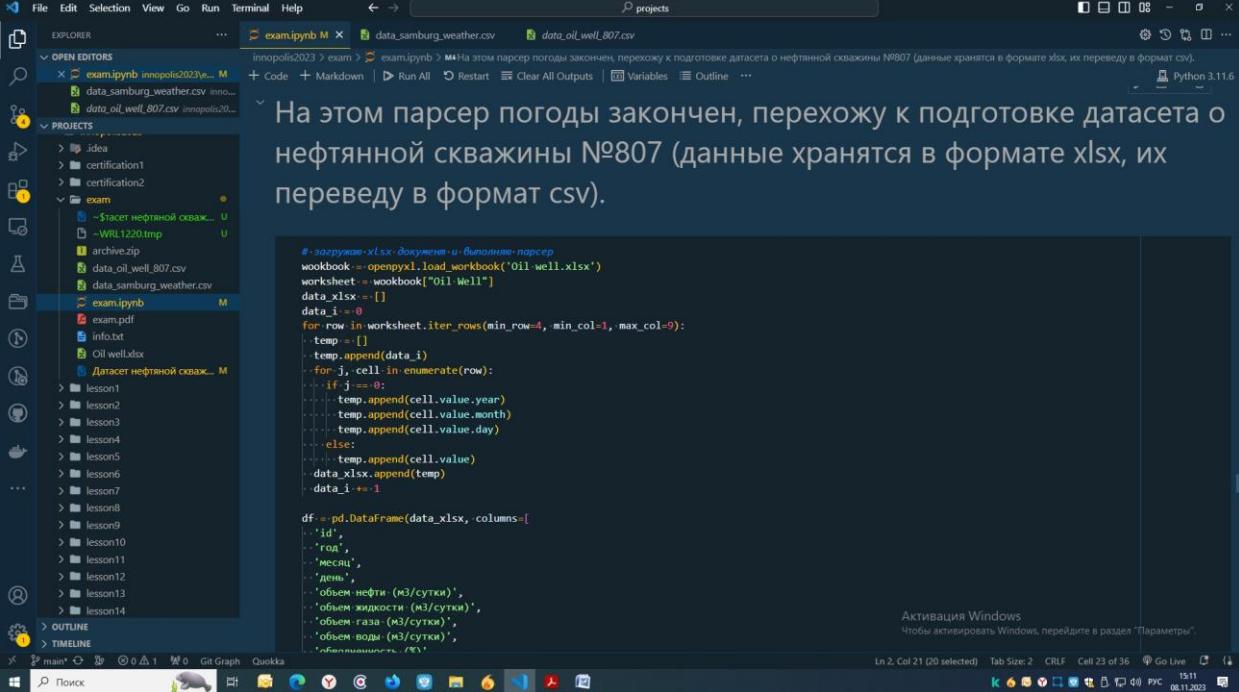
```
[35] ✓ 0.0s
```

| | count | unique | top | freq | mean | std | min | 10% | 25% | 50% | 75% | 90% | max |
|-----------------------|--------|--------|--|------|-------------|------------|-----------|-----------|--------|--------|---------|--------|--------|
| id | 2350.0 | NaN | NaN | NaN | 1174.5 | 678.530889 | 0.0 | 234.9 | 587.25 | 1174.5 | 1761.75 | 2114.1 | 2349.0 |
| год | 2350.0 | NaN | NaN | NaN | 2020.155319 | 1.881358 | 2017.0 | 2018.0 | 2019.0 | 2020.0 | 2022.0 | 2023.0 | 2023.0 |
| месяц | 2350.0 | NaN | NaN | NaN | 6.659149 | 3.390316 | 1.0 | 2.0 | 4.0 | 7.0 | 10.0 | 11.0 | 12.0 |
| день | 2350.0 | NaN | NaN | NaN | 15.701702 | 8.793305 | 1.0 | 4.0 | 8.0 | 16.0 | 23.0 | 28.0 | 31.0 |
| погодное условие | 2350 | 6 | пасмурно | 1549 | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN |
| направление ветра | 2350 | 8 | южный | 445 | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN |
| скорость ветра (м/с) | 2350.0 | NaN | NaN | NaN | 5.511489 | 2.76689 | 1.0 | 2.0 | 4.0 | 5.0 | 7.0 | 9.0 | 44.0 |
| давление (мм рт. ст.) | 2350.0 | NaN | NaN | NaN | 759.973617 | 9.550216 | 729.0 | 748.0 | 754.0 | 759.0 | 765.0 | 772.0 | 806.0 |
| влажность (%) | 2350.0 | NaN | NaN | NaN | 79.794468 | 17.196529 | 0.0 | 52.0 | 70.0 | 85.0 | 93.0 | 97.0 | 100.0 |
| видимость (мм) | 2350.0 | NaN | NaN | NaN | NaN | NaN | 23.398 | 86.264654 | 1.0 | 10.0 | 10.0 | 10.0 | 992.0 |
| луна | 2350 | 4 | расступа | 1111 | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN |
| осадки (мм) | 2350.0 | NaN | NaN | NaN | 0.140085 | 0.704188 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.1 | 13.4 |
| температура днем | 2350.0 | NaN | NaN | NaN | NaN | NaN | -4.195745 | 13.979581 | -46.0 | -24.0 | -14.0 | -1.0 | 13.0 |
| температура ночью | 2350.0 | NaN | NaN | NaN | NaN | NaN | -2.524255 | 14.767881 | -46.0 | -23.0 | -13.0 | 0.0 | 10.0 |
| url | 2350 | 2350 | https://pogoda1.ru/samburg/07-06-2017/ | 1 | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN |

Активация Windows
чтобы активировать Windows, перейдите в раздел "Параметры".

Ln 2, Col 21 (20 selected) Tab Size: 2 CRLF Cell 10 of 36 ⌂ Go Live ⌂ 15:09 06.11.2023

На этом парсер погоды закончен, перехожу к подготовке датасета о нефтяной скважины №807 (данные хранятся в формате xlsx, их переведу в формат csv).



```
# загружаем xlsx-документ и фиксируем парсер
workbook = openpyxl.load_workbook('Oil Well.xlsx')
worksheet = workbook['Oil Well']
data_xlsx = []
data_i = 0
for row in worksheet.iter_rows(min_row=4, min_col=1, max_col=9):
    temp = []
    temp.append(data_i)
    for j, cell in enumerate(row):
        if j == 0:
            temp.append(cell.value.year)
            temp.append(cell.value.month)
            temp.append(cell.value.day)
        else:
            temp.append(cell.value)
    data_xlsx.append(temp)
    data_i += 1

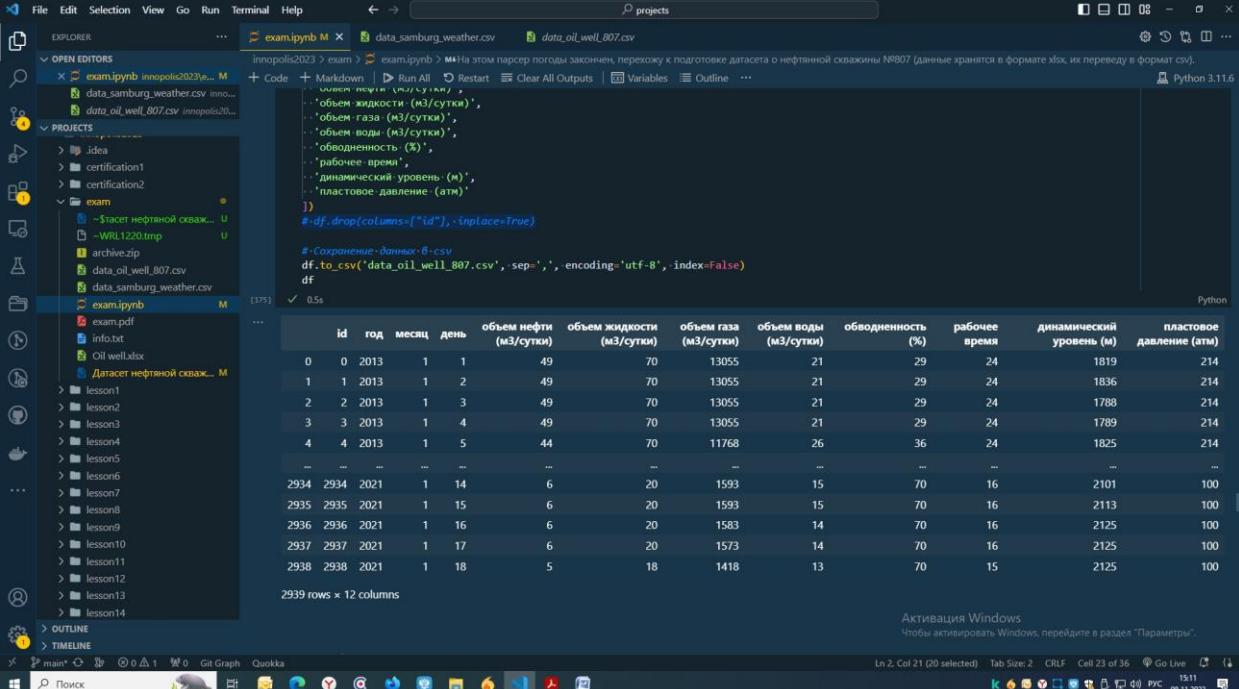
df = pd.DataFrame(data_xlsx, columns=[

    'id',
    'год',
    'месяц',
    'день',
    'объем нефти (м³/сутки)',
    'объем жидкости (м³/сутки)',
    'объем газа (м³/сутки)',
    'объем воды (м³/сутки)',
    'обводненность (%)'
])
# df.drop(columns=['id'], inplace=True)

# Сохранение данных в csv
df.to_csv('data_oil_well_807.csv', sep=',', encoding='utf-8', index=False)
df
```

Активация Windows
Чтобы активировать Windows, перейдите в раздел "Параметры".

Ln 2, Col 21 (20 selected) Tab Size: 2 CRLF Cell 23 of 36 Go Live 15:11 08.11.2023



| | id | год | месяц | день | объем нефти (м³/сутки) | объем жидкости (м³/сутки) | объем газа (м³/сутки) | объем воды (м³/сутки) | обводненность (%) | рабочее время | динамический уровень (м) | пластовое давление (атм) |
|------|-----------|------------|--------------|-------------|-------------------------------|----------------------------------|------------------------------|------------------------------|--------------------------|----------------------|---------------------------------|---------------------------------|
| 0 | 0 | 2013 | 1 | 1 | 49 | 70 | 13055 | 21 | 29 | 24 | 1819 | 214 |
| 1 | 1 | 2013 | 1 | 2 | 49 | 70 | 13055 | 21 | 29 | 24 | 1836 | 214 |
| 2 | 2 | 2013 | 1 | 3 | 49 | 70 | 13055 | 21 | 29 | 24 | 1788 | 214 |
| 3 | 3 | 2013 | 1 | 4 | 49 | 70 | 13055 | 21 | 29 | 24 | 1789 | 214 |
| 4 | 4 | 2013 | 1 | 5 | 44 | 70 | 11768 | 26 | 36 | 24 | 1825 | 214 |
| ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| 2934 | 2934 | 2021 | 1 | 14 | 6 | 20 | 1593 | 15 | 70 | 16 | 2101 | 100 |
| 2935 | 2935 | 2021 | 1 | 15 | 6 | 20 | 1593 | 15 | 70 | 16 | 2113 | 100 |
| 2936 | 2936 | 2021 | 1 | 16 | 6 | 20 | 1583 | 14 | 70 | 16 | 2125 | 100 |
| 2937 | 2937 | 2021 | 1 | 17 | 6 | 20 | 1573 | 14 | 70 | 16 | 2125 | 100 |
| 2938 | 2938 | 2021 | 1 | 18 | 5 | 18 | 1418 | 13 | 70 | 15 | 2125 | 100 |

2939 rows × 12 columns

Активация Windows
Чтобы активировать Windows, перейдите в раздел "Параметры".

Ln 2, Col 21 (20 selected) Tab Size: 2 CRLF Cell 23 of 36 Go Live 15:11 08.11.2023

Необходимые данные получил в размере 2939 строк. Сохранил данные в файл и загрузил к себе в Git репозиторий. Дальше идет подготовка данных, а именно просмотр пустот и заполнения средними или наиболее встречающимися значениями, просмотр типов данных.

```
# Загрузка файла из Git-моего репозитория-6-Pandas
# data_oil_well_807=pd.read_csv('https://raw.githubusercontent.com/SotGE/innopolis2023/main/exam/data_oil_well_807.csv',sep=',',index_col=False)
data_oil_well_807 = pd.read_csv('https://raw.githubusercontent.com/SotGE/innopolis2023/main/exam/data_oil_well_807.csv', sep=',', index_col=False)
```

| | id | год | месяц | день | объем нефти (м3/сутки) | объем жидкости (м3/сутки) | объем газа (м3/сутки) | объем воды (м3/сутки) | обводненность (%) | рабочее время | динамический уровень (м) | пластовое давление (атм) |
|------|-----------|------------|--------------|-------------|-------------------------------|----------------------------------|------------------------------|------------------------------|--------------------------|----------------------|---------------------------------|---------------------------------|
| 0 | 0 | 2013 | 1 | 1 | 49 | 70 | 13055 | 21 | 29 | 24 | 1819 | 214 |
| 1 | 1 | 2013 | 1 | 2 | 49 | 70 | 13055 | 21 | 29 | 24 | 1836 | 214 |
| 2 | 2 | 2013 | 1 | 3 | 49 | 70 | 13055 | 21 | 29 | 24 | 1788 | 214 |
| 3 | 3 | 2013 | 1 | 4 | 49 | 70 | 13055 | 21 | 29 | 24 | 1789 | 214 |
| 4 | 4 | 2013 | 1 | 5 | 44 | 70 | 11768 | 26 | 36 | 24 | 1825 | 214 |
| ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| 2934 | 2934 | 2021 | 1 | 14 | 6 | 20 | 1593 | 15 | 70 | 16 | 2101 | 100 |
| 2935 | 2935 | 2021 | 1 | 15 | 6 | 20 | 1593 | 15 | 70 | 16 | 2113 | 100 |
| 2936 | 2936 | 2021 | 1 | 16 | 6 | 20 | 1583 | 14 | 70 | 16 | 2125 | 100 |

```
# Размер - данных - (количество - строк, - колонок)
data_oil_well_807.shape
```

```
(2939, 12)
```

```
# Просмотр - типов - данных - 6- данным
print(data_oil_well_807.info())
```

```
[1]:
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2939 entries, 0 to 2938
Data columns (total 12 columns):
 #   Column           Non-Null Count  Dtype  
--- 
 0   id              2939 non-null   int64  
 1   год             2939 non-null   int64  
 2   месяц          2939 non-null   int64  
 3   день            2939 non-null   int64  
 4   объем нефти (м3/сутки)  2939 non-null   int64  
 5   объем жидкости (м3/сутки) 2939 non-null   int64  
 6   объем газа (м3/сутки)   2939 non-null   int64  
 7   объем воды (м3/сутки)   2939 non-null   int64  
 8   обводненность (%)     2939 non-null   int64  
 9   рабочее время        2939 non-null   int64  
 10  динамический уровень (м) 2939 non-null   int64  
 11  пластовое давление (атм) 2939 non-null   int64  
dtypes: int64(12)
memory usage: 275.7 KB
None
```

File Edit Selection View Go Run Terminal Help projects

OPEN EDITORS exam.ipynb M data_samburg.weather.csv data_oil_well_807.csv

PROJECTS

- exam.ipynb imopolis2023y... M
- data_samburg_weather.csv imo...
- data_oil_well_807.csv imopoli...
- archive.zip
- exam.pdf
- info.txt
- Oil well.xlsx
- Дагестан нефтяной скваж... M
- lesson1
- lesson2
- lesson3
- lesson4
- lesson5
- lesson6
- lesson7
- lesson8
- lesson9
- lesson10
- lesson11
- lesson12
- lesson13
- lesson14

OUTLINE

TIMELINE

Python 3.11.6

```
#-Проверка данных
#-Количество нулевых значений (неправильно считанные)
data_oil_well_807.isnull().sum()
```

0.0s

| | id | год | месяц | день | объем нефти (м3/сутки) | объем жидкости (м3/сутки) | объем газа (м3/сутки) | объем воды (м3/сутки) | обводненность (%) | рабочее время | динамический уровень (м) | пластовое давление (атм) |
|--------|-------|-----|-------|------|------------------------|---------------------------|-----------------------|-----------------------|-------------------|---------------|--------------------------|--------------------------|
| dtype: | int64 | | | | | | | | | | | |


```
#-Количество неопределенные значения (неправильно считанные)
data_oil_well_807.isna().sum()
```

0.0s

| | id | год | месяц | день | объем нефти (м3/сутки) | объем жидкости (м3/сутки) | объем газа (м3/сутки) | объем воды (м3/сутки) | обводненность (%) | рабочее время | динамический уровень (м) | пластовое давление (атм) |
|--------|-------|-----|-------|------|------------------------|---------------------------|-----------------------|-----------------------|-------------------|---------------|--------------------------|--------------------------|
| dtype: | int64 | | | | | | | | | | | |

Активация Windows
чтобы активировать Windows, перейдите в раздел "Параметры".

3 selections Tab Size: 2 CRLF Cell 25 of 36 Go Live

15:12 06.11.2023

File Edit Selection View Go Run Terminal Help projects

OPEN EDITORS exam.ipynb M data_samburg.weather.csv data_oil_well_807.csv

PROJECTS

- exam.ipynb imopolis2023y... M
- data_samburg_weather.csv imo...
- data_oil_well_807.csv imopoli...
- archive.zip
- exam.pdf
- info.txt
- Oil well.xlsx
- Дагестан нефтяной скваж... M
- lesson1
- lesson2
- lesson3
- lesson4
- lesson5
- lesson6
- lesson7
- lesson8
- lesson9
- lesson10
- lesson11
- lesson12
- lesson13
- lesson14

OUTLINE

TIMELINE

Python 3.11.6

```
#-Описательная статистика
data_oil_well_807.describe(include='all', percentiles=[0.1, 0.25, 0.5, 0.75, 0.9]).T
```

0.0s

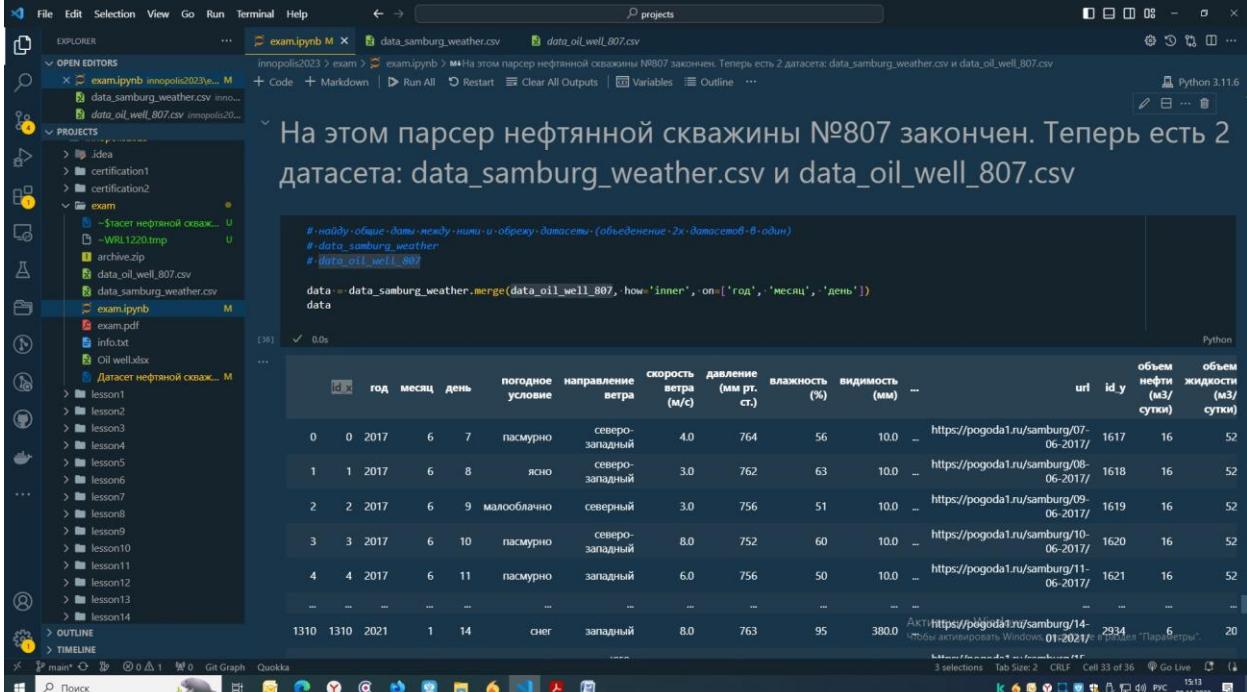
| | count | mean | std | min | 10% | 25% | 50% | 75% | 90% | max |
|---------------------------|--------|-------------|-------------|--------|--------|--------|--------|--------|--------|---------|
| id | 2939.0 | 1469.000000 | 848.560546 | 0.0 | 293.8 | 734.5 | 1469.0 | 2203.5 | 2644.2 | 2938.0 |
| год | 2939.0 | 2016.529092 | 2.311826 | 2013.0 | 2013.0 | 2015.0 | 2017.0 | 2019.0 | 2020.0 | 2021.0 |
| месяц | 2939.0 | 6.489622 | 3.466087 | 1.0 | 2.0 | 3.0 | 7.0 | 10.0 | 11.0 | 12.0 |
| день | 2939.0 | 15.686288 | 8.794391 | 1.0 | 4.0 | 8.0 | 16.0 | 23.0 | 28.0 | 31.0 |
| объем нефти (м3/сутки) | 2939.0 | 17.624362 | 9.689026 | 0.0 | 8.0 | 11.0 | 15.0 | 22.0 | 32.0 | 49.0 |
| объем жидкости (м3/сутки) | 2939.0 | 59.464103 | 18.634101 | 12.0 | 32.0 | 50.0 | 58.0 | 74.0 | 83.0 | 113.0 |
| объем газа (м3/сутки) | 2939.0 | 4730.146308 | 2598.888524 | 4.0 | 2134.8 | 3041.5 | 3909.0 | 5843.5 | 8511.0 | 13113.0 |
| объем воды (м3/сутки) | 2939.0 | 41.828853 | 13.056625 | 9.0 | 23.0 | 33.0 | 43.0 | 50.0 | 58.0 | 99.0 |
| обводненность (%) | 2939.0 | 70.694794 | 9.534203 | 29.0 | 59.0 | 69.0 | 73.0 | 76.0 | 80.0 | 100.0 |
| рабочее время | 2939.0 | 22.344675 | 3.039553 | 7.0 | 17.0 | 22.0 | 24.0 | 24.0 | 24.0 | 24.0 |
| динамический уровень (м) | 2939.0 | 1930.383464 | 114.543752 | 1529.0 | 1823.0 | 1855.0 | 1890.0 | 2008.0 | 2122.0 | 2137.0 |
| пластовое давление (атм) | 2939.0 | 157.019054 | 32.917150 | 100.0 | 111.0 | 129.0 | 157.0 | 185.5 | 203.0 | 214.0 |

Активация Windows
чтобы активировать Windows, перейдите в раздел "Параметры".

3 selections Tab Size: 2 CRLF Cell 25 of 36 Go Live

15:12 06.11.2023

На этом парсер нефтяной скважины №807 закончен. Теперь есть 2 датасета: data_samburg_weather.csv и data_oil_well_807.csv

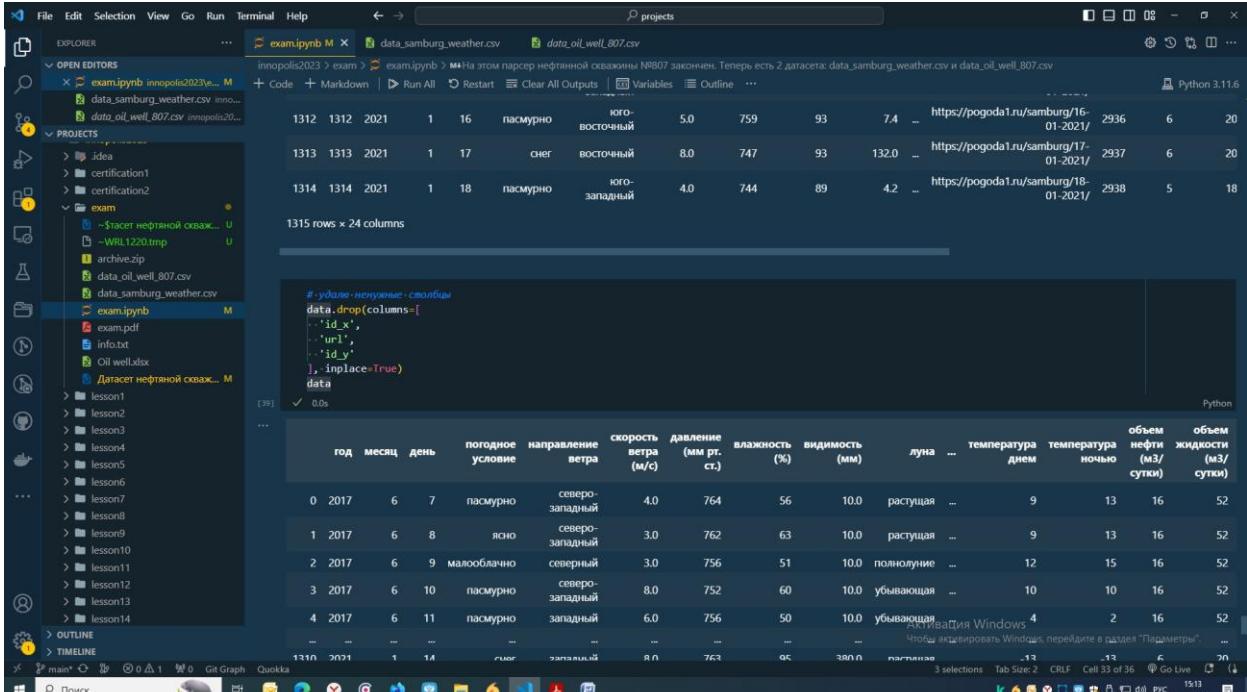


```
#найду общие даты между ними и обрежу датасеты (объединение - 2х датасетов - 8 один)
#data_samburg_weather
#data_oil_well_807

data = data_samburg_weather.merge(data_oil_well_807, how='inner', on=['год', 'месяц', 'день'])

data
```

| id_x | год | месяц | день | погодное условие | направление ветра | скорость ветра (м/с) | давление (мм рт. ст.) | влажность (%) | видимость (мм) | url | id_y | объем нефти (м3/ сутки) | объем жидкости (м3/ сутки) |
|------|------|-------|------|---------------------|----------------------|----------------------------|-----------------------------|------------------|-------------------|---|------|----------------------------------|-------------------------------------|
| 0 | 2017 | 6 | 7 | пасмурно | северо- западный | 4.0 | 764 | 56 | 10.0 | https://pogoda1.ru/samburg/07-06-2017/ | 1617 | 16 | 52 |
| 1 | 2017 | 6 | 8 | ясно | северо- западный | 3.0 | 762 | 63 | 10.0 | https://pogoda1.ru/samburg/08-06-2017/ | 1618 | 16 | 52 |
| 2 | 2017 | 6 | 9 | малооблачно | северный | 3.0 | 756 | 51 | 10.0 | https://pogoda1.ru/samburg/09-06-2017/ | 1619 | 16 | 52 |
| 3 | 2017 | 6 | 10 | пасмурно | северо- западный | 8.0 | 752 | 60 | 10.0 | https://pogoda1.ru/samburg/10-06-2017/ | 1620 | 16 | 52 |
| 4 | 2017 | 6 | 11 | пасмурно | западный | 6.0 | 756 | 50 | 10.0 | https://pogoda1.ru/samburg/11-06-2017/ | 1621 | 16 | 52 |
| ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| 1310 | 2021 | 1 | 14 | снег | западный | 8.0 | 763 | 95 | 380.0 | Активы https://pogoda1.ru/samburg/14-01-2021/ | 2934 | 6 | 20 |



```
#удаляю ненужные столбцы
data.drop(columns=[  
    'id_x',  
    'url',  
    'id_y',  
], inplace=True)  
data
```

| год | месяц | день | погодное условие | направление ветра | скорость ветра (м/с) | давление (мм рт. ст.) | влажность (%) | видимость (мм) | луна | температура днем | температура ночью | объем нефти (м3/ сутки) | объем жидкости (м3/ сутки) | |
|------|-------|------|---------------------|----------------------|----------------------------|-----------------------------|------------------|-------------------|------------|---------------------|----------------------|----------------------------------|-------------------------------------|-----|
| 2017 | 6 | 7 | пасмурно | северо- западный | 4.0 | 764 | 56 | 10.0 | растущая | ... | 9 | 13 | 16 | 52 |
| 2017 | 6 | 8 | ясно | северо- западный | 3.0 | 762 | 63 | 10.0 | растущая | ... | 9 | 13 | 16 | 52 |
| 2017 | 6 | 9 | малооблачно | северный | 3.0 | 756 | 51 | 10.0 | полнолуние | ... | 12 | 15 | 16 | 52 |
| 2017 | 6 | 10 | пасмурно | северо- западный | 8.0 | 752 | 60 | 10.0 | убывающая | ... | 10 | 10 | 16 | 52 |
| 2017 | 6 | 11 | пасмурно | западный | 6.0 | 756 | 50 | 10.0 | убывающая | ... | 4 | 2 | 16 | 52 |
| ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| 1310 | 2021 | 1 | 14 | снег | западный | 8.0 | 763 | 95 | 380.0 | Активы Windows | 4 | ... | ... | ... |

Данные для исследования подготовил.

Этап 3. Обработка данных перед загрузкой в модель

The screenshot shows a Jupyter Notebook interface with the file `exam.ipynb` open. In the code cell, the command `data.info()` is run, displaying the following information:

```
#-Проверка -значений- на -0
(data == 0).sum()

#-Как видно, -представленные- параметры- могут- находиться- в- значении-0

#-Заполнение -нулевых- значений- -median() -не- требуется
#-data -> data.replace(0, data.median())

... ГОД 0
месяц 0
день 0
погодное условие 0
направление ветра 0
скорость ветра (м/с) 0
давление (мм рт. ст.) 0
влажность (%) 5
видимость (км) 0
луна 0
осадки (мм) 1033
температура днем 83
температура ночью 77
объем нефти (м3/сутки) 1
объем жидкости (м3/сутки) 0
объем газа (м3/сутки) 0
объем воды (м3/сутки) 0
обводненность (%) 0
рабочее время 0
динамический уровень (м) 0
пластовое давление (атм) 0
dtype: int64
```

Below the code cell, the message "#-Проверка типов данных в датасете" is visible. The status bar at the bottom right shows "13.11.2023 12:06".

The screenshot shows a Jupyter Notebook interface with the file `exam.ipynb` open. In the code cell, the command `data.info()` is run, displaying the following detailed information about the DataFrame structure:

```
#-Проверка типов данных в датасете
data.info()
```

| # | Column | Non-Null Count | Dtype |
|----|---------------------------|----------------|---------|
| 0 | год | 1315 | int64 |
| 1 | месяц | 1315 | int64 |
| 2 | день | 1315 | int64 |
| 3 | погодное условие | 1315 | object |
| 4 | направление ветра | 1315 | object |
| 5 | скорость ветра (м/с) | 1315 | float64 |
| 6 | давление (мм рт. ст.) | 1315 | int64 |
| 7 | влажность (%) | 1315 | int64 |
| 8 | видимость (км) | 1315 | float64 |
| 9 | луна | 1315 | object |
| 10 | осадки (мм) | 1315 | float64 |
| 11 | температура днем | 1315 | int64 |
| 12 | температура ночью | 1315 | int64 |
| 13 | объем нефти (м3/сутки) | 1315 | int64 |
| 14 | объем жидкости (м3/сутки) | 1315 | int64 |
| 15 | объем газа (м3/сутки) | 1315 | int64 |
| 16 | объем воды (м3/сутки) | 1315 | int64 |
| 17 | обводненность (%) | 1315 | int64 |
| 18 | рабочее время | 1315 | int64 |
| 19 | динамический уровень (м) | 1315 | int64 |
| 20 | пластовое давление (атм) | 1315 | int64 |

```
memory usage: 215.9+ KB
```

Below the code cell, the message "#-Переведем горизонтальные данные в строковые - в- числовые - с- сортировкой по - алфавиту" is visible. The status bar at the bottom right shows "13.11.2023 12:07".

File Edit Selection View Go Run Terminal Help projects

OPEN EDITORS exam.ipynb ×

exam.ipynb innopolis2023exam

PROJECTS

- idea
- certification1
- certification2
- certification3
- exam
- catboost_info
- archive.zip
- data_oil_well_807.csv
- data_samburg_weather.csv
- exam.ipynb
- exam.pdf
- info.txt
- Oil well.xlsx
- Датасет нефтяной скважины...
- lesson1
- lesson2
- lesson3
- lesson4
- lesson5
- lesson6
- lesson7
- lesson8
- lesson9
- lesson10
- lesson11
- catboost.info
- data.csv
- lesson11.1.pdf
- lesson11.2.docx

#-Перевел -категориальные -данные -> строковые -& -числовые -> сортировкой по -алфавиту

```
data['погодные условия'] = pd.factorize(data['погодные условия'], astype='category'), sort=True)[0]+1
data['направление ветра'] = pd.factorize(data['направление ветра'], astype='category'), sort=True)[0]+1
data['луна'] = pd.factorize(data['луна'], astype='category'), sort=True)[0]+1
data
```

0.0s

| год | месяц | день | погодное условие | направление ветра | скорость ветра (м/с) | давление (мм рт. ст.) | влажность (%) | видимость (мм) | луна | температура днем | температура ночью | объем нефти (м3/сутки) | объем жидкости (м3/сутки) | объем газа (м3/сутки) | | |
|------|-------|------|------------------|-------------------|----------------------|-----------------------|---------------|----------------|-------|------------------|-------------------|------------------------|---------------------------|-----------------------|-----|------|
| 0 | 2017 | 6 | 7 | 4 | 5 | 4.0 | 764 | 56 | 10.0 | 3 | ... | 9 | 13 | 16 | 52 | 4210 |
| 1 | 2017 | 6 | 8 | 6 | 5 | 3.0 | 762 | 63 | 10.0 | 3 | ... | 9 | 13 | 16 | 52 | 4210 |
| 2 | 2017 | 6 | 9 | 2 | 3 | 3.0 | 756 | 51 | 10.0 | 2 | ... | 12 | 15 | 16 | 52 | 4210 |
| 3 | 2017 | 6 | 10 | 4 | 5 | 8.0 | 752 | 60 | 10.0 | 4 | ... | 10 | 10 | 16 | 52 | 4210 |
| 4 | 2017 | 6 | 11 | 4 | 2 | 6.0 | 756 | 50 | 10.0 | 4 | ... | 4 | 2 | 16 | 52 | 4210 |
| ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| 1310 | 2021 | 1 | 14 | 5 | 2 | 8.0 | 763 | 95 | 380.0 | 3 | ... | -13 | -13 | 6 | 20 | 1593 |
| 1311 | 2021 | 1 | 15 | 4 | 7 | 3.0 | 770 | 87 | 10.0 | 3 | ... | -20 | -21 | 6 | 20 | 1593 |
| 1312 | 2021 | 1 | 16 | 4 | 6 | 5.0 | 759 | 93 | 7.4 | 3 | ... | -17 | -16 | 6 | 20 | 1583 |
| 1313 | 2021 | 1 | 17 | 5 | 1 | 8.0 | 747 | 93 | 132.0 | 3 | ... | -19 | -19 | 6 | 20 | 1573 |
| 1314 | 2021 | 1 | 18 | 4 | 7 | 4.0 | 744 | 89 | 4.2 | 3 | ... | -18 | -24 | 5 | 18 | 1418 |

1315 rows × 21 columns

#-Просмотр -типов -данных -& -датасете

```
data.info()
```

Активация Windows
Чтобы активировать Windows, перейдите в раздел "Параметры".

Ln 2, Col 1 (441 selected) Tab Size: 2 CRLF Cell 65 of 84 Go Live 12:07 13.11.2023

File Edit Selection View Go Run Terminal Help projects

OPEN EDITORS exam.ipynb ×

exam.ipynb innopolis2023exam

PROJECTS

- idea
- certification1
- certification2
- certification3
- exam
- catboost_info
- archive.zip
- data_oil_well_807.csv
- data_samburg_weather.csv
- exam.ipynb
- exam.pdf
- info.txt
- Oil well.xlsx
- Датасет нефтяной скважины...
- lesson1
- lesson2
- lesson3
- lesson4
- lesson5
- lesson6
- lesson7
- lesson8
- lesson9
- lesson10
- lesson11
- catboost.info
- data.csv
- lesson11.1.pdf
- lesson11.2.docx

#-Просмотр -типов -данных -& -датасете

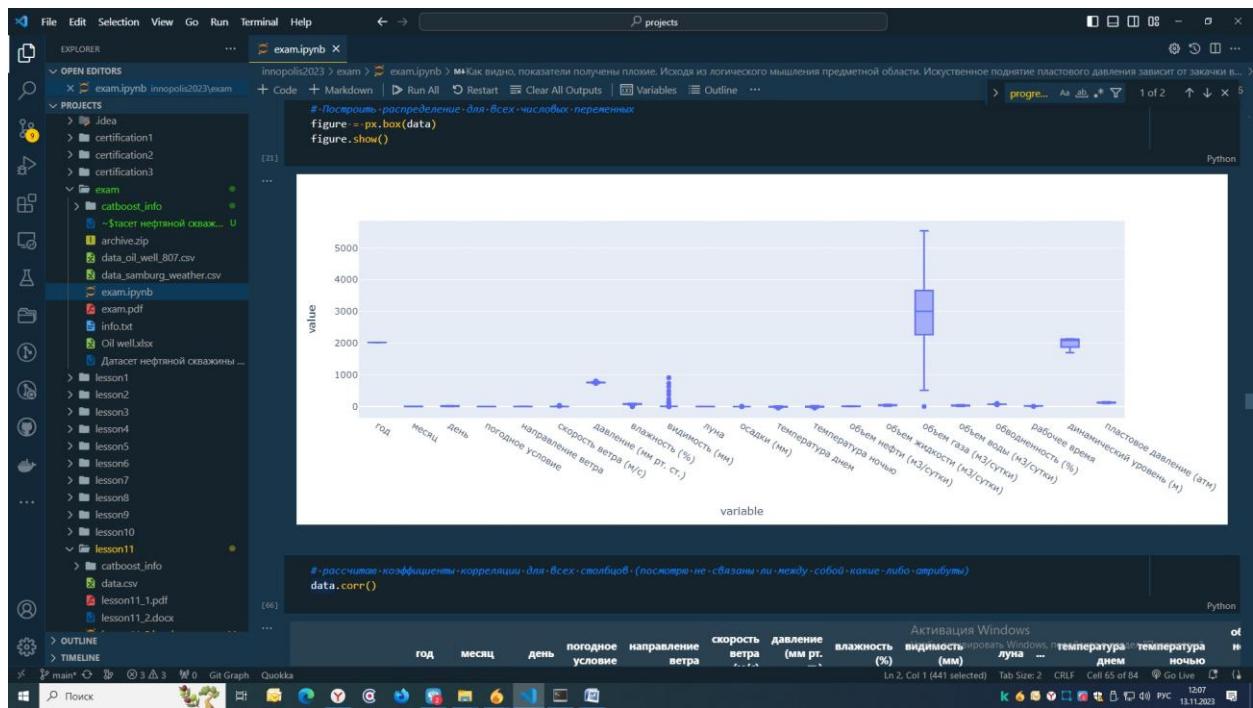
```
#-Получил -все -числовые -данные
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1315 entries, 0 to 1314
Data columns (total 21 columns):
 #   Column           Non-Null Count  Dtype  
 ---  -- 
 0   год              1315 non-null   int64  
 1   месяц            1315 non-null   int64  
 2   день              1315 non-null   int64  
 3   погодное условие 1315 non-null   int64  
 4   направление ветра 1315 non-null   int64  
 5   скорость ветра (м/с) 1315 non-null   float64
 6   давление (мм рт. ст.) 1315 non-null   int64  
 7   влажность (%)    1315 non-null   int64  
 8   видимость (мм)   1315 non-null   float64
 9   луна              1315 non-null   int64  
 10  осадки (мм)     1315 non-null   float64
 11  температура днем 1315 non-null   int64  
 12  температура ночью 1315 non-null   int64  
 13  объем нефти (м3/сутки) 1315 non-null   int64  
 14  объем жидкости (м3/сутки) 1315 non-null   int64  
 15  объем газа (м3/сутки) 1315 non-null   int64  
 16  объем воды (м3/сутки) 1315 non-null   int64  
 17  обводненность (%) 1315 non-null   int64  
 18  рабочее время    1315 non-null   int64  
 19  динамический уровень (м) 1315 non-null   int64  
 20  пластовое давление (атм) 1315 non-null   int64  
dtypes: float64(3), int64(18)
memory usage: 215.9 KB
```

#-Построить -распределение -для -всех -числовых -переменных

Активация Windows
Чтобы активировать Windows, перейдите в раздел "Параметры".

Ln 2, Col 1 (441 selected) Tab Size: 2 CRLF Cell 65 of 84 Go Live 12:07 13.11.2023



File Edit Selection View Go Run Terminal Help projects

OPEN EDITORS exam.ipynb ×

exam.ipynb innopolis2023exam

PROJECTS

- idea
- certification1
- certification2
- certification3
- exam
- catboost_info
- archive.zip
- data_oil_well_807.csv
- data_samburg_weather.csv
- exam.ipynb
- exam.pdf
- info.txt
- Oil well.xlsx
- Датасет нефтяной скважин...
- lesson1
- lesson2
- lesson3
- lesson4
- lesson5
- lesson6
- lesson7
- lesson8
- lesson9
- lesson10
- lesson11
- catboost.info
- data.csv
- lesson11.1.pdf
- lesson11.2.docx

рассчитаем - коэффициенты - корреляции - для - всех - столбцов - (посмотрю - не - связаны - ли - между - собой - какие - либо - атрибуты)
data.corr()

| | год | месяц | день | погодное условие | направление ветра | скорость ветра (м/с) | давление (мм рт. ст.) | влажность (%) | видимость (мм) | луна | температура днем | температура ночью | | |
|---------------------------|-----------|-----------|-----------|---------------------|----------------------|----------------------------|-----------------------------|------------------|-------------------|-----------|---------------------|----------------------|-----------|-------|
| год | 1.000000 | -0.237267 | -0.026652 | -0.046422 | 0.024324 | 0.101737 | -0.323560 | -0.052083 | 0.094247 | 0.002791 | - | -0.083252 | -0.079080 | -0.75 |
| месяц | -0.237267 | 1.000000 | 0.024301 | 0.024254 | -0.020146 | -0.003688 | 0.043810 | 0.098486 | 0.079023 | -0.13040 | - | 0.176811 | 0.146865 | -0.05 |
| день | -0.026652 | 0.024301 | 1.000000 | 0.034970 | -0.045750 | -0.000921 | -0.026426 | -0.009375 | 0.020795 | -0.105912 | - | -0.028802 | -0.028967 | 0.01 |
| погодное условие | -0.046422 | 0.024254 | 0.034970 | 1.000000 | 0.049172 | -0.052509 | 0.140615 | -0.100295 | 0.025039 | -0.028246 | - | -0.172482 | -0.158432 | 0.01 |
| направление ветра | 0.024324 | -0.020146 | -0.045750 | 0.049172 | 1.000000 | 0.087580 | -0.011782 | 0.117353 | -0.045021 | 0.039956 | - | -0.091349 | -0.076476 | 0.00 |
| скорость ветра (м/с) | 0.101737 | -0.003688 | -0.000921 | -0.052509 | 0.087580 | 1.000000 | -0.266876 | 0.023140 | -0.000429 | -0.026514 | - | 0.146523 | 0.159741 | -0.10 |
| давление (мм рт. ст.) | -0.323560 | 0.043810 | -0.026426 | 0.140615 | -0.011782 | -0.266876 | 1.000000 | -0.099229 | 0.001019 | 0.045537 | - | -0.430273 | -0.426717 | 0.27 |
| влажность (%) | -0.052083 | 0.098486 | -0.009375 | -0.100295 | 0.117353 | 0.023140 | -0.099229 | 1.000000 | 0.031578 | 0.016435 | - | -0.161586 | -0.218488 | 0.01 |
| видимость (мм) | 0.094247 | 0.079023 | 0.020795 | 0.025039 | -0.045021 | -0.000429 | 0.001019 | 0.031578 | 1.000000 | -0.084598 | - | -0.056957 | -0.056661 | -0.12 |
| луна | 0.002791 | -0.013040 | -0.105912 | -0.028246 | 0.039956 | -0.026514 | 0.045537 | 0.016435 | -0.084598 | 1.000000 | - | 0.015826 | 0.012682 | 0.01 |
| осадки (мм) | -0.097533 | 0.028005 | -0.012482 | -0.162065 | 0.018718 | 0.089236 | -0.190706 | 0.172020 | -0.019356 | -0.011811 | - | 0.117444 | 0.097749 | 0.07 |
| температура днем | -0.083252 | 0.176811 | -0.028802 | -0.172482 | -0.091349 | 0.146523 | -0.430273 | -0.161586 | -0.056957 | 0.015826 | - | 1.000000 | 0.977463 | -0.01 |
| температура ночью | -0.079080 | 0.146865 | -0.028967 | -0.158432 | -0.076476 | 0.159741 | -0.426717 | -0.218488 | -0.056661 | 0.012682 | - | 0.977463 | 1.000000 | -0.01 |
| объем нефти (м3/сутки) | -0.759238 | -0.057944 | 0.016719 | 0.016720 | 0.007556 | -0.103005 | 0.278294 | 0.014231 | 0.122641 | 0.012014 | 0.0010971 | 0.010984 | 1.00 | |

File Edit Selection View Go Run Terminal Help projects

OPEN EDITORS exam.ipynb x

exam.ipynb innopolis2023exam

Code Markdown Run All Restart Clear All Outputs Variables Outline

#-посмотреть-пары-с-высокой-корреляцией
df = data.corr().abs().unstack().sort_values(ascending=False).drop_duplicates()

#-сброс-ограниченный-на-10000-таблицами-данных
#pd.set_option('display.max_rows', None)
#pd.set_option('display.max_columns', None)
#pd.set_option('display.max_colwidth', None)

#-df

print(df.to_markdown())

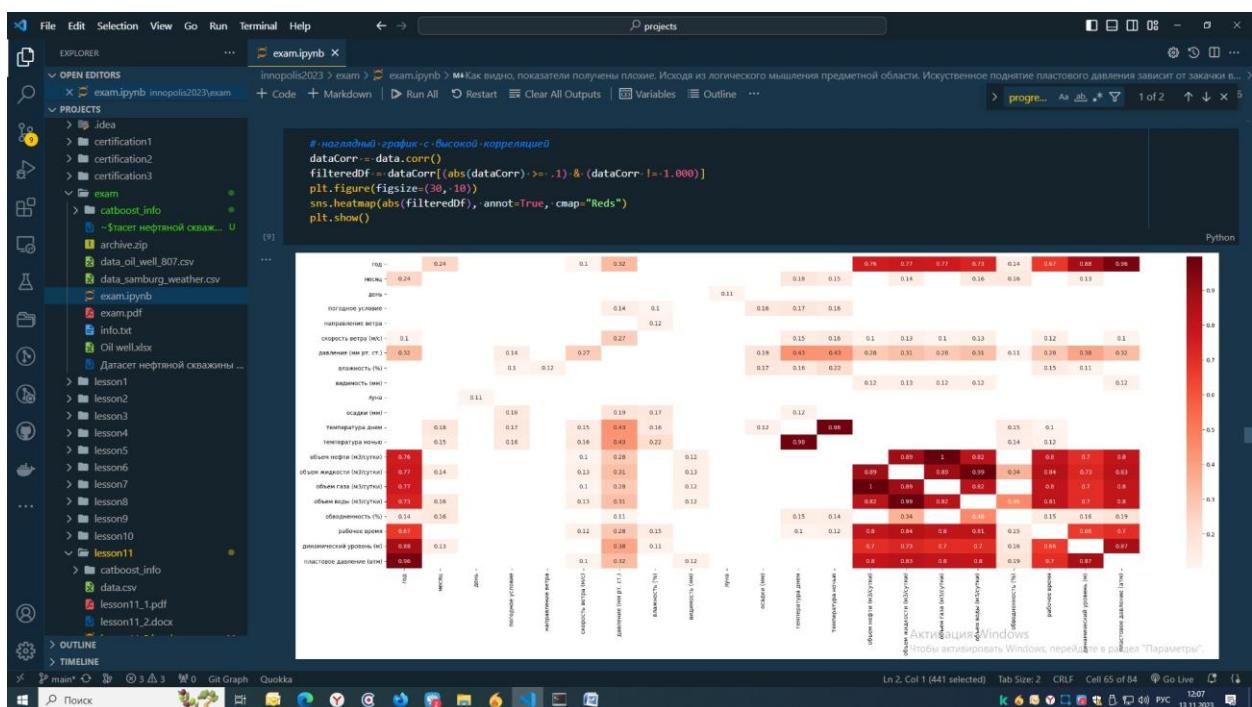
#display(df.to_string())

Python

| | | 0 |
|---|---|---|
| (‘год’, ‘год’) | (‘объем газа (м3/сутки)’, ‘объем нефти (м3/сутки)’) | 1 |
| (‘объем воды (м3/сутки)’, ‘объем жидкости (м3/сутки)’) | 0.996308 | |
| (‘температура днем’, ‘температура ночью’) | 0.977463 | |
| (‘пластовое давление (атм)’, ‘год’) | 0.96275 | |
| (‘объем жидкости (м3/сутки)’, ‘объем газа (м3/сутки)’) | 0.893276 | |
| (‘объем жидкости (м3/сутки)’, ‘объем нефти (м3/сутки)’) | 0.889227 | |
| (‘год’, ‘динамический уровень (%)’) | 0.882661 | |
| (‘динамический уровень (%)’, ‘пластовое давление (атм)’) | 0.873967 | |
| (‘объем жидкости (м3/сутки)’, ‘рабочее время’) | 0.838319 | |
| (‘объем жидкости (м3/сутки)’, ‘пластовое давление (атм)’) | 0.826001 | |
| (‘объем воды (м3/сутки)’, ‘объем газа (м3/сутки)’) | 0.824716 | |
| (‘объем воды (м3/сутки)’, ‘объем нефти (м3/сутки)’) | 0.820455 | |
| (‘объем воды (м3/сутки)’, ‘рабочее время’) | 0.814131 | |
| (‘объем газа (м3/сутки)’, ‘рабочее время’) | 0.804058 | |
| (‘рабочее время’, ‘объем нефти (м3/сутки)’) | 0.802173 | |
| (‘объем газа (м3/сутки)’, ‘пластовое давление (атм)’) | 0.801352 | |
| (‘объем воды (м3/сутки)’, ‘пластовое давление (атм)’) | 0.79939 | |
| (‘объем нефти (м3/сутки)’, ‘пластовое давление (атм)’) | 0.796269 | |
| (‘объем газа (м3/сутки)’, ‘год’) | 0.76557 | |
| (‘объем газа (м3/сутки)’, ‘0’) | 0.75060 | |

Активация Windows
Чтобы активировать Windows, перейдите в раздел “Параметры”.

Ln 2, Col 1 (441 selected) Tab Size: 2 CRLF Cell 65 of 84 Go Live 12/07 13.11.2023



Я не учитываю в корреляции ['погодное условие', 'направление ветра', 'луна'] так как это категориальные данные, переведенные в число.

File Edit Selection View Go Run Terminal Help projects

OPEN EDITORS exam.ipynb M

PROJECTS

- idea
- certification1
- certification2
- certification3
- exam
- catboost_info
- archive.zip
- data_oil_well_807.csv
- data_samburg_weather.csv
- exam.ipynb M
- exam.pdf
- info.txt
- Oil well.xlsx
- Датасет нефтяной скважины...
- lesson1
- lesson2
- lesson3
- lesson4
- lesson5
- lesson6
- lesson7
- lesson8
- lesson9
- lesson10
- lesson11
- catboost.info
- data.csv
- lesson11.1.pdf
- lesson11.2.docx

OUTLINE

TIMELINE

MAIN

Поиск

exam.ipynb

```
#-Разделение-для-задачи-классификации-по-X-(эндогенные-переменные,-т.е.-регрессоры-или-независимые)-и-у-(эндогенные-переменные-или-зависимые)

#Убираю-не-нужные-данные-с-датой,-так-как-я-не-буду-исследовать-данные-б-зависимости-от-времени-года
data_ml = data.drop(['год', 'месяц', 'день'], axis=1)

#['']
#...погодное-усладие',
#...направление-ветра',
#...скорость-ветра-(м/с)',
#...давление-(мм-рт.-ст.)',
#...влагность-(%)',
#...видимость-(мм)',
#...луга',
#...осадки-(мм)',
#...температура-днем',
#...температура-ночью'
#]

#['']
#...объем-нефти-(м3/сумки)',
#...объем-жидкости-(м3/сумки)',
#...объем-газа-(м3/сумки)',
#...объем-воды-(м3/сумки)',
#...обводненность-(%)',
#...рабочее-время',
#...динамический-уровень-(м)',
#...пластовое-давление-(атм)'
#]

X = data_ml.drop([
    'объем-нефти-(м3/сумки)',
    'объем-жидкости-(м3/сумки)',
    'объем-газа-(м3/сумки)',
    'объем-воды-(м3/сумки)',
    'обводненность-(%)',
    'рабочее-время',
    'динамический-уровень-(м)',
    'пластовое-давление-(атм)'
])

#Подготовка-данных

#Нормализация-(MinMaxScaler)
scalar = MinMaxScaler()
features = scalar.fit_transform(X, y)
X_normalised = pd.DataFrame(features, columns=X.columns)
X_normalised
```

Активация Windows
Чтобы активировать Windows, перейдите в раздел "Параметры".

Ln 2, Col 1 (441 selected) Tab Size: 2 CRLF Cell 45 of 84 Go Live 12:09 13.11.2023

File Edit Selection View Go Run Terminal Help projects

OPEN EDITORS exam.ipynb M

PROJECTS

- idea
- certification1
- certification2
- certification3
- exam
- catboost_info
- archive.zip
- data_oil_well_807.csv
- data_samburg_weather.csv
- exam.ipynb M
- exam.pdf
- info.txt
- Oil well.xlsx
- Датасет нефтяной скважины...
- lesson1
- lesson2
- lesson3
- lesson4
- lesson5
- lesson6
- lesson7
- lesson8
- lesson9
- lesson10
- lesson11
- catboost.info
- data.csv
- lesson11.1.pdf
- lesson11.2.docx

OUTLINE

TIMELINE

MAIN

Поиск

exam.ipynb

```
#...объем-воды-(м3/сумки)',
#...обводненность-(%)',
#...рабочее-время',
#...динамический-уровень-(м)',
#...пластовое-давление-(атм)'
#]

#0.0s

#Подготовка-данных

#Нормализация-(MinMaxScaler)
scalar = MinMaxScaler()
features = scalar.fit_transform(X, y)
X_normalised = pd.DataFrame(features, columns=X.columns)
X_normalised
```

| | погодное условие | направление ветра | скорость ветра (м/с) | давление (мм рт. ст.) | влагность (%) | видимость (мм) | луга | осадки (мм) | температура днем | температура ночью |
|------|------------------|-------------------|----------------------|-----------------------|---------------|----------------|----------|-------------|------------------|-------------------|
| 0 | 0.6 | 0.571429 | 0.069767 | 0.454545 | 0.56 | 0.009738 | 0.666667 | 0.0 | 0.820896 | 0.776316 |
| 1 | 1.0 | 0.571429 | 0.046512 | 0.428571 | 0.63 | 0.009738 | 0.666667 | 0.0 | 0.820896 | 0.776316 |
| 2 | 0.2 | 0.285714 | 0.046512 | 0.350649 | 0.51 | 0.009738 | 0.333333 | 0.0 | 0.865672 | 0.802632 |
| 3 | 0.6 | 0.571429 | 0.162791 | 0.298701 | 0.60 | 0.009738 | 1.000000 | 0.0 | 0.835821 | 0.736842 |
| 4 | 0.6 | 0.142857 | 0.116279 | 0.350649 | 0.50 | 0.009738 | 1.000000 | 0.0 | 0.746269 | 0.631579 |
| ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| 1310 | 0.8 | 0.142857 | 0.162791 | 0.441558 | 0.95 | 0.414597 | 0.666667 | 0.0 | 0.492537 | 0.434211 |
| 1311 | 0.6 | 0.857143 | 0.046512 | 0.532468 | 0.87 | 0.009738 | 0.666667 | 0.0 | 0.388060 | 0.328947 |
| 1312 | 0.6 | 0.714286 | 0.093023 | 0.389610 | 0.93 | 0.006894 | 0.666667 | 0.0 | 0.432836 | 0.394737 |
| 1313 | 0.8 | 0.000000 | 0.162791 | 0.233766 | 0.93 | 0.143232 | 0.666667 | 0.0 | 0.402985 | 0.355263 |
| 1314 | 0.6 | 0.857143 | 0.069767 | 0.194805 | 0.89 | 0.003392 | 0.666667 | 0.0 | 0.417910 | 0.289474 |

Активация Windows
Чтобы активировать Windows, перейдите в раздел "Параметры".

Ln 2, Col 1 (441 selected) Tab Size: 2 CRLF Cell 45 of 84 Go Live 12:09 13.11.2023

File Edit Selection View Go Run Terminal Help projects

OPEN EDITORS exam.ipynb innopolis2023y... M

PROJECTS

- idea
- certification1
- certification2
- certification3
- exam
- catboost_info
- archive.zip
- data_oil_well_807.csv
- data_samburg_weather.csv
- exam.ipynb
- exam.pdf
- info.txt
- Oil well.xlsx
- Датасет нефтяной скважины ...
- lesson1
- lesson2
- lesson3
- lesson4
- lesson5
- lesson6
- lesson7
- lesson8
- lesson9
- lesson10
- lesson11
- catboost.info
- data.csv
- lesson11.1.pdf
- lesson11.2.docx

OUTLINE

TIMELINE

Python

```
# Построить распределение для всех числовых нормализованных переменных
figure = px.box(X_normalised)
figure.show()
```

Aктивация Windows
чтобы активировать Windows, перейдите в раздел "Параметры".

Ln 2, Col 1 (441 selected) Tab Size: 2 CRLF Cell 45 of 84 Go Live 12:10 13.11.2023

Этап 4. Разделение данных на тренировочную, тестовую и валидационную. Обучение и тестирование моделей

File Edit Selection View Go Run Terminal Help projects

OPEN EDITORS exam.ipynb innopolis2023y... M

PROJECTS

- idea
- certification1
- certification2
- certification3
- exam
- catboost_info
- archive.zip
- data_oil_well_807.csv
- data_samburg_weather.csv
- exam.ipynb
- exam.pdf
- info.txt
- Oil well.xlsx
- Датасет нефтяной скважины ...
- lesson1
- lesson2
- lesson3
- lesson4
- lesson5
- lesson6
- lesson7
- lesson8
- lesson9
- lesson10
- lesson11
- catboost.info
- data.csv
- lesson11.1.pdf
- lesson11.2.docx

OUTLINE

TIMELINE

Python

```
# Разделение на тренировочную (80%), тестовую (10%) и валидационную (10%)
X_train, X_test, y_train, y_test = train_test_split(X_normalised, y, test_size=0.2, random_state=1024, shuffle=True)
X_test, X_val, y_test, y_val = train_test_split(X_test, y_test, test_size=0.5, random_state=1024, shuffle=True)

# Рассмотрим разные алгоритмы
models = [
    LinearRegression(), # Метод наименьших квадратов
    RandomForestRegressor(n_estimators=100, max_features='sqrt'), # Случайный лес
    KNeighborsRegressor(n_neighbors=6), # Метод ближайших соседей
    SVC(kernel='linear'), # Метод опорных векторов с линейным ядром (для одномерного массива у)
    LogisticRegression() # Логистическая регрессия (для одномерного массива у)
    DecisionTreeClassifier(max_depth=4, random_state=42), # Деревья решений
    RandomForestClassifier(min_samples_split=5, n_estimators=1000), # Ассоциации деревьев решений
    GradientBoostingClassifier(max_depth=4, random_state=42), # Ассоциации градиентного спуска
    KNeighborsClassifier(n_neighbors=5) # Обучение моделей К-ближайших соседей
]

# Строки: графики тренировочных данных
fig, axes = plt.subplots(nrows=1, ncols=2, figsize=(12, 4))
sns.histplot(data=X_train, ax=axes[0], palette='dark')
boxplot = sns.boxplot(data=X_train, ax=axes[1], palette='pastel')
boxplot.tick_params(axis='x', rotation=30)
plt.tight_layout()
```

Count

Луна

Видимость (мм)

Давление (мм рт. ст.)

Скорость ветра (м/с)

Направление ветра

Погодное условие

Осадки (мм)

Активация Windows
чтобы активировать Windows, перейдите в раздел "Параметры".

Ln 2, Col 1 (441 selected) Tab Size: 2 CRLF Cell 49 of 84 Go Live Total Lines: 16 Prettier 12:12 13.11.2023

File Edit Selection View Go Run Terminal Help projects

OPEN EDITORS exam.ipynb M

exam.ipynb innopolis2023y... M

PROJECTS

- idea
- certification1
- certification2
- certification3
- exam
- catboost_info
- archive.zip
- data_oil_well_807.csv
- data_samburg_weather.csv
- exam.ipynb
- exam.pdf
- info.txt
- Oil well.xlsx
- Датасет нефтяной скважин...
- lesson1
- lesson2
- lesson3
- lesson4
- lesson5
- lesson6
- lesson7
- lesson8
- lesson9
- lesson10
- lesson11
- catboost.info
- data.csv
- lesson11.1.pdf
- lesson11.2.docx

Code + Markdown | ▶ Run All ⌘ Restart ⌘ Clear All Outputs Variables Outline ...

Симплекс-диаграмма-треугольниковых-данных
fig, axes = plt.subplots(rows=1, ncols=2, figsize=(12, 4))
sns.histplot(data=X_train, ax=axes[0], palette='dark')
boxplot = sns.boxplot(data=X_train, ax=axes[1], palette='pastel')
boxplot.tick_params(axis='x', rotation=30)
plt.tight_layout()

Count

погодное условие
направление ветра
скорость ветра (м/с)
давление (мм рт. ст.)
влажность (%)
видимость (мм)
осадки (мм)
температура днем
температура ночью

Создаю временную структуру для графика
models_test = pd.DataFrame()
models_temp = {}

Для каждой модели из списка
for model in models:

Aктивация Windows
Чтобы активировать Windows, перейдите в раздел "Параметры".

Ln 2, Col 1 (441 selected) Tab Size: 2 CRLF Cell 49 of 84 Go Live

File Edit Selection View Go Run Terminal Help projects

OPEN EDITORS exam.ipynb M

exam.ipynb innopolis2023y... M

PROJECTS

- idea
- certification1
- certification2
- certification3
- exam
- catboost_info
- archive.zip
- data_oil_well_807.csv
- data_samburg_weather.csv
- exam.ipynb
- exam.pdf
- info.txt
- Oil well.xlsx
- Датасет нефтяной скважин...
- lesson1
- lesson2
- lesson3
- lesson4
- lesson5
- lesson6
- lesson7
- lesson8
- lesson9
- lesson10
- lesson11
- catboost.info
- data.csv
- lesson11.1.pdf
- lesson11.2.docx

Code + Markdown | ▶ Run All ⌘ Restart ⌘ Clear All Outputs Variables Outline ...

Создаю временную структуру для графика
models_test = pd.DataFrame()
models_temp = {}

Для каждой модели из списка
for model in models:

Модель LinearRegression:
Правильность на обучаемом наборе: 0.128
Правильность на тестовом наборе: 0.079

Модель LinearRegression:
Правильность на обучаемом наборе: 0.142
Правильность на тестовом наборе: 0.122

Модель LinearRegression:
Правильность на обучаемом наборе: 0.129
Правильность на тестовом наборе: 0.075

Модель LinearRegression:
Правильность на обучаемом наборе: 0.139
Правильность на тестовом наборе: 0.122

Модель LinearRegression:
Правильность на обучаемом наборе: 0.044
Правильность на тестовом наборе: -0.011

Aктивация Windows
Чтобы активировать Windows, перейдите в раздел "Параметры".

Ln 2, Col 1 (441 selected) Tab Size: 2 CRLF Cell 49 of 84 Go Live

File Edit Selection View Go Run Terminal Help projects

OPEN EDITORS exam.ipynb M

exam.ipynb innopolis2023y... M

PROJECTS

- idea
- certification1
- certification2
- certification3
- exam
- catboost_info
- archive.zip
- data_oil_well_807.csv
- data_samburg_weather.csv
- exam.ipynb
- exam.pdf
- info.txt
- Oil well.xlsx
- Датасет нефтяной скважин...
- lesson1
- lesson2
- lesson3
- lesson4
- lesson5
- lesson6
- lesson7
- lesson8
- lesson9
- lesson10
- lesson11
- catboost.info
- data.csv
- lesson11.1.pdf
- lesson11.2.docx

OUTLINE

TIMELINE

SEARCH

Python

```
# Кoefficientsы-дeterminants
models_test_abs = models_test.abs()
models_test_abs
```

| | R2_y1 | R2_y2 | R2_y3 | R2_y4 | R2_y5 | R2_y6 | R2_y7 | R2_y8 |
|---|----------|-------------|--------------|-----------|------------|-----------|---------------|-------------|
| 0 | 0.078734 | 136.617551 | 1.003104e-06 | 63.064032 | 488.72200 | 12.731727 | 475800.692359 | 1552.887861 |
| 1 | 0.204729 | 139.553802 | 0.024188e-06 | 64.350407 | 488.103624 | 12.624907 | 476573.033082 | 1557.507793 |
| 2 | 0.083182 | 141.1563418 | 0.031262e+06 | 65.850907 | 488.892107 | 12.980761 | 473280.184347 | 1576.112447 |

```
# Максимизация данных -o- > 1- до 1- относительно каждого столбца
models_test_normalized = (models_test_abs - models_test_abs.min()) / (models_test_abs.max() - models_test_abs.min())
models_test_normalized
```

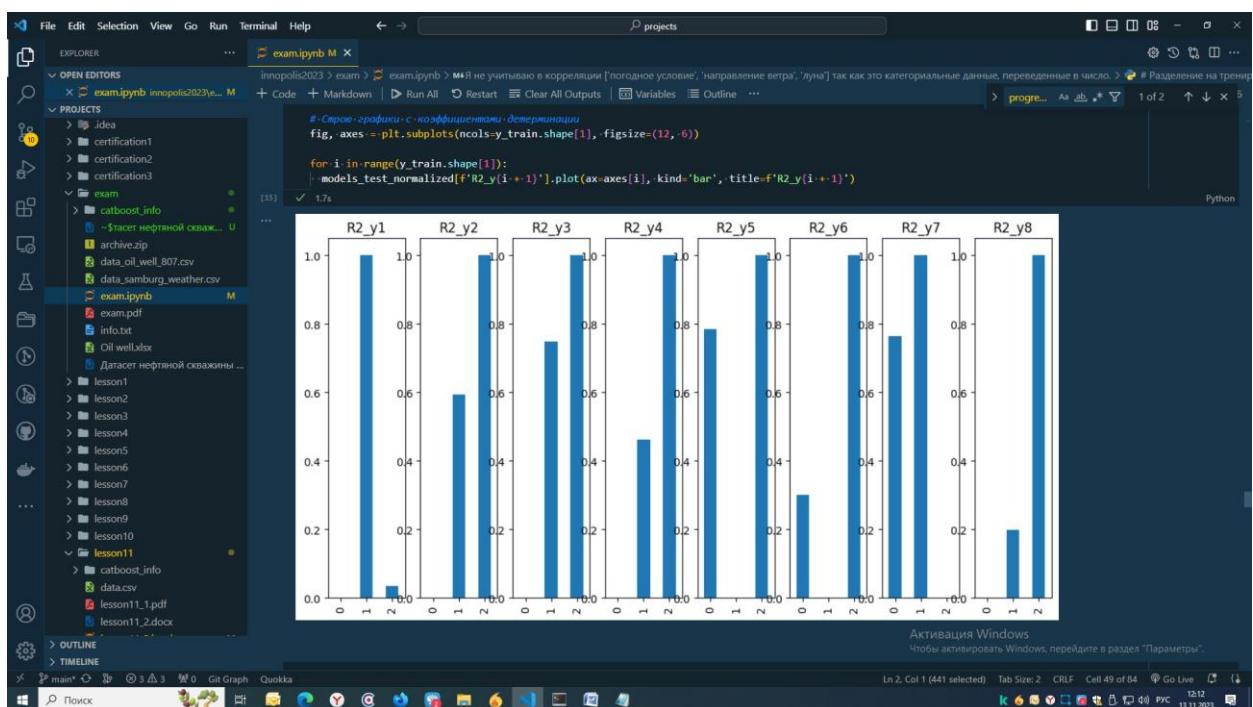
| | R2_y1 | R2_y2 | R2_y3 | R2_y4 | R2_y5 | R2_y6 | R2_y7 | R2_y8 |
|---|----------|----------|----------|----------|----------|----------|----------|----------|
| 0 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.784515 | 0.300179 | 0.765449 | 0.000000 |
| 1 | 1.000000 | 0.593678 | 0.748785 | 0.461583 | 0.000000 | 0.000000 | 1.000000 | 0.198924 |
| 2 | 0.035301 | 1.000000 | 1.000000 | 1.000000 | 1.000000 | 1.000000 | 0.000000 | 1.000000 |

```
# Столбцы-графики с коэффициентами-дeterminants
fig, axes = plt.subplots(ncols=y_train.shape[1], figsize=(12, 6))

for i in range(y_train.shape[1]):
    models_test_normalized[f'R2_y_{i+1}'].plot(ax=axes[i], kind='bar', title=f'R2_y_{i+1}')
```

Активация Windows
чтобы активировать Windows, перейдите в раздел "Параметры".

Ln 2, Col 1 (441 selected) Tab Size: 2 CRLF Cell 49 of 84 Go Live



File Edit Selection View Go Run Terminal Help projects

OPEN EDITORS exam.ipynb M

PROJECTS

- idea
- certification1
- certification2
- certification3
- exam
- catboost_info
- > archive.zip
- > data_oil_well_807.csv
- > data_samburg_weather.csv
- > exam.ipynb M
- > exam.pdf
- > info.txt
- > Oil well.xlsx
- > Датасет нефтяной скважины...
- lesson1
- lesson2
- lesson3
- lesson4
- lesson5
- lesson6
- lesson7
- lesson8
- lesson9
- lesson10
- lesson11
- > catboost.info
- > data.csv
- > lesson11.1.pdf
- > lesson11.2.docx

#-Проверку-анализ-лучших-признаков

```
#-Для-каждого-столбца-результатующего-набора
for i in range(y_train.shape[1]):
    # Выбор 2 лучших признаков из 8 по Ху-квадрату
    selector = SelectKBest(chi2, k=2)
    X_train_best = selector.fit_transform(X_train, y_train.iloc[:, i])
    # Принт(X_train_best.shape)
    X.columns(selector.get_support(indices=True))
    vector_names = list(X.columns(selector.get_support(indices=True)))
    print(f'Для {[y.columns[i]}] лучшие признаки: {[vector_names]}')
    # X_train_best_df = pd.DataFrame(X_train_best, columns=selector.get_support(indices=True))
    # print(X_train_best_df)

#-Для ['объем нефти (м3/сутки)'] лучшие признаки: ['видимость (mm)', 'осадки (mm)']
Для ['объем жидкости (м3/сутки)'] лучшие признаки: ['видимость (mm)', 'осадки (mm)']
Для ['объем газа (м3/сутки)'] лучшие признаки: [' направление ветра', 'осадки (mm)']
Для ['объем воды (м3/сутки)'] лучшие признаки: ['видимость (mm)', 'температура ночью']
Для ['обводненность (%)'] лучшие признаки: ['температура днем', 'температура ночью']
Для ['рабочее время'] лучшие признаки: ['давление (мм рт. ст.)', 'видимость (mm)']
Для ['динамический уровень (м)'] лучшие признаки: ['осадки (mm)', 'температура днем']
Для [' пластовое давление (атм)'] лучшие признаки: ['температура днем', 'температура ночью']

#-обучение-лучшей-модели-с-лучшими-признаками
model_best = models[2]
model_best.fit(X_train, y_train)

#-качество-модели
score = model.score(X_test,y_test)
print("Accuracy: ", score*100)

#-предсказание-на-новой-выборке
data_best = pd.DataFrame(model_best.predict(X_test), columns=[

    'объем нефти (м3/сутки)',
    'объем жидкости (м3/сутки)',
    'объем газа (м3/сутки)',
    'объем воды (м3/сутки)',
    'обводненность (%)',
    'рабочее время',
    'динамический уровень (м)',
    ' пластовое давление (атм)'
])
data_best
```

Python

Активация Windows чтобы активировать Windows, перейдите в раздел "Параметры".

Ln 2, Col 1 (441 selected) Tab Size: 2 CRLF Cell 49 of 84 Go Live 12.12 13.11.2023

File Edit Selection View Go Run Terminal Help projects

OPEN EDITORS exam.ipynb M

PROJECTS

- idea
- certification1
- certification2
- certification3
- exam
- catboost_info
- > archive.zip
- > data_oil_well_807.csv
- > data_samburg_weather.csv
- > exam.ipynb M
- > exam.pdf
- > info.txt
- > Oil well.xlsx
- > Датасет нефтяной скважины...
- lesson1
- lesson2
- lesson3
- lesson4
- lesson5
- lesson6
- lesson7
- lesson8
- lesson9
- lesson10
- lesson11
- > catboost.info
- > data.csv
- > lesson11.1.pdf
- > lesson11.2.docx

#-обучение-лучшей-модели-с-лучшими-признаками

```
model_best = models[2]
model_best.fit(X_train, y_train)

#-качество-модели
score = model.score(X_test,y_test)
print("Accuracy: ", score*100)

#-предсказание-на-новой-выборке
data_best = pd.DataFrame(model_best.predict(X_test), columns=[

    'объем нефти (м3/сутки)',
    'объем жидкости (м3/сутки)',
    'объем газа (м3/сутки)',
    'объем воды (м3/сутки)',
    'обводненность (%)',
    'рабочее время',
    'динамический уровень (м)',
    ' пластовое давление (атм)'
])
data_best
```

| | объем нефти (м3/сутки) | объем жидкости (м3/сутки) | объем газа (м3/сутки) | объем воды (м3/сутки) | обводненность (%) | рабочее время | динамический уровень (м) | пластовое давление (атм) |
|---|------------------------|---------------------------|-----------------------|-----------------------|-------------------|---------------|--------------------------|--------------------------|
| 0 | 9.500000 | 38.833333 | 2512.833333 | 29.500000 | 75.000000 | 18.333333 | 2097.000000 | 116.000000 |
| 1 | 10.666667 | 42.833333 | 2854.666667 | 32.166667 | 74.333333 | 20.000000 | 2011.833333 | 123.333333 |
| 2 | 12.333333 | 50.000000 | 3295.333333 | 38.000000 | 74.666667 | 22.666667 | 1906.166667 | 128.833333 |
| 3 | 13.000000 | 49.500000 | 3484.833333 | 36.666667 | 73.333333 | 21.833333 | 1938.000000 | 137.166667 |
| 4 | 8.833333 | 37.666667 | 2415.666667 | 28.666667 | 76.000000 | 18.166667 | 2029.666667 | 118.500000 |

Python

Активация Windows чтобы активировать Windows, перейдите в раздел "Параметры".

Ln 2, Col 1 (441 selected) Tab Size: 2 CRLF Cell 49 of 84 Go Live 12.12 13.11.2023

```

#-score модели - (методы для классификации)
def model_report(model, X_train, y_train, X_test, y_test, average='weighted'):
    # Делаем предсказания на тренировочном наборе
    y_pred_train = model.predict(X_train)
    # Делаем предсказания на тестовом наборе
    y_pred_test = model.predict(X_test)

    # Оцениваем точность модели на тренировочном наборе
    print(f"Тренировочный набор - Матрица ошибок (confusion matrix): \n{confusion_matrix(y_train, y_pred_train)}")
    print(f"Тренировочный набор - Правильность (accuracy) модели: {accuracy_score(y_train, y_pred_train)}")
    print(f"Тренировочный набор - Точность (precision) модели: {precision_score(y_train, y_pred_train, average=average)}")
    print(f"Тренировочный набор - Полнота (recall) модели: {recall_score(y_train, y_pred_train, average=average)}")
    print(f"Тренировочный набор - F1-мера модели: {f1_score(y_train, y_pred_train, average=average)}")
    print(f"Тренировочный набор - Средняя абсолютная ошибка (mean absolute error): {mean_absolute_error(y_train, y_pred_train)}\n\n")

    # Оцениваем точность модели на тестовом наборе
    print(f"Тестовый набор - Матрица ошибок (confusion matrix): \n{confusion_matrix(y_test, y_pred_test)}")
    print(f"Тестовый набор - Правильность (accuracy) модели: {accuracy_score(y_test, y_pred_test)}")
    print(f"Тестовый набор - Точность (precision) модели: {precision_score(y_test, y_pred_test, average=average)}")
    print(f"Тестовый набор - Полнота (recall) модели: {recall_score(y_test, y_pred_test, average=average)}")
    print(f"Тестовый набор - F1-мера модели: {f1_score(y_test, y_pred_test, average=average)}")
    print(f"Тестовый набор - Средняя абсолютная ошибка (mean absolute error): {mean_absolute_error(y_test, y_pred_test)}\n\n")

#-Сделал обучение - (KNeighborsRegressor) Влияние температуры на обводненность
KNeighbors = KNeighborsRegressor(n_neighbors=6)
KNeighbors.fit(X_train[['температура днем', 'температура ночью']], y_train[['обводненность (%)']])
print(f"Тренировочный набор: {KNeighbors.score(X_train[['температура днем', 'температура ночью']], y_train[['обводненность (%)']])*100}")
print(f"Тестовый набор: {KNeighbors.score(X_test[['температура днем', 'температура ночью']], y_test[['обводненность (%)']])*100}")

#-Сделал обучение - (KNeighborsRegressor) Влияние температуры на рабочее время
KNeighbors = KNeighborsRegressor(n_neighbors=6)
KNeighbors.fit(X_train[['температура днем', 'температура ночью']], y_train[['рабочее время']])
print(f"Тренировочный набор: {KNeighbors.score(X_train[['температура днем', 'температура ночью']], y_train[['рабочее время']])*100}")
print(f"Тестовый набор: {KNeighbors.score(X_test[['температура днем', 'температура ночью']], y_test[['рабочее время']])*100}")

#-Сделал обучение - (KNeighborsRegressor) Влияние температуры на динамический уровень
KNeighbors = KNeighborsRegressor(n_neighbors=6)
KNeighbors.fit(X_train[['температура днем', 'температура ночью']], y_train[['динамический уровень (м)']])
print(f"Тренировочный набор: {KNeighbors.score(X_train[['температура днем', 'температура ночью']], y_train[['динамический уровень (м)']])*100}")
print(f"Тестовый набор: {KNeighbors.score(X_test[['температура днем', 'температура ночью']], y_test[['динамический уровень (м)']])*100}")

#-Сделал обучение - (KNeighborsRegressor) Влияние температуры на пластовое давление
KNeighbors = KNeighborsRegressor(n_neighbors=6)
KNeighbors.fit(X_train[['температура днем', 'температура ночью']], y_train[['пластовое давление (атм)']])
print(f"Тренировочный набор: {KNeighbors.score(X_train[['температура днем', 'температура ночью']], y_train[['пластовое давление (атм)']])*100}")
print(f"Тестовый набор: {KNeighbors.score(X_test[['температура днем', 'температура ночью']], y_test[['пластовое давление (атм)']])*100}")

```

```

#-Сделал обучение - (KNeighborsRegressor) Влияние температуры на рабочее время
KNeighbors = KNeighborsRegressor(n_neighbors=6)
KNeighbors.fit(X_train[['температура днем', 'температура ночью']], y_train[['рабочее время']])
print(f"Тренировочный набор: {KNeighbors.score(X_train[['температура днем', 'температура ночью']], y_train[['рабочее время']])*100}")
print(f"Тестовый набор: {KNeighbors.score(X_test[['температура днем', 'температура ночью']], y_test[['рабочее время']])*100}")

#-Сделал обучение - (KNeighborsRegressor) Влияние температуры на динамический уровень
KNeighbors = KNeighborsRegressor(n_neighbors=6)
KNeighbors.fit(X_train[['температура днем', 'температура ночью']], y_train[['динамический уровень (м)']])
print(f"Тренировочный набор: {KNeighbors.score(X_train[['температура днем', 'температура ночью']], y_train[['динамический уровень (м)']])*100}")
print(f"Тестовый набор: {KNeighbors.score(X_test[['температура днем', 'температура ночью']], y_test[['динамический уровень (м)']])*100}")

#-Сделал обучение - (KNeighborsRegressor) Влияние температуры на пластовое давление
KNeighbors = KNeighborsRegressor(n_neighbors=6)
KNeighbors.fit(X_train[['температура днем', 'температура ночью']], y_train[['пластовое давление (атм)']])
print(f"Тренировочный набор: {KNeighbors.score(X_train[['температура днем', 'температура ночью']], y_train[['пластовое давление (атм)']])*100}")
print(f"Тестовый набор: {KNeighbors.score(X_test[['температура днем', 'температура ночью']], y_test[['пластовое давление (атм)']])*100}")

```

Как видно, показатели получены плохие. Исходя из логического мышления предметной области. Искусственное поднятие пластового давления зависит от закачки воды. Естественное поднятие пластового давления зависит от водоносного слоя, который подпирает нефть, что приводит к увеличению давления. Сам же водоносный слой зависит от осадков, чем больше осадков, тем выше водоносный слой.

File Edit Selection View Go Run Terminal Help

projects

OPEN EDITORS

exam.ipynb M Копия блокнота_ensemble_ml_algorithms_bagging_boosting_voting.ipynb simple_knn_algorithm_demo.ipynb

Изменения получены плюсике. Исходы из логического мышления предметной области. Искусственное поднятие пластового давления зависит от закачки ... # Сгруппирую данные по дням, месяцам и годам (среднее)

Code + Markdown Run All Restart Clear All Outputs Variables Outline

Rando... Аа ab * 6 of 20

PROJECTS

exam.ipynb

```
#-Сгруппирую данные по дням, месяцам и годам (среднее)
dataml_group = data.groupby(by=[data['месяц'], data['год']]).mean()

#-Разделение для задачи классификации на X (известные переменные, т.е., -рекрессоры или независимые) и Y (издогенные переменные или -забисимые)
X = dataml_group[[ 'объем нефти (м3/сутки)', 'объем жидкости (м3/сутки)', 'объем газа (м3/сутки)', 'объем воды (м3/сутки)', 'оводненность (%)', 'рабочее время', 'динамический уровень (м)', 'пластовое давление (атм)'], axis=1]

y = dataml_group[[ 'объем нефти (м3/сутки)', 'объем жидкости (м3/сутки)', 'объем газа (м3/сутки)', 'объем воды (м3/сутки)', 'оводненность (%)', 'рабочее время', 'динамический уровень (м)', 'пластовое давление (атм)' ]]

#-Разделение на тренировочную (80%), тестовую (10%) и валидационную (10%)
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=1024, shuffle=True)
X_val, X_test, y_val = train_test_split(X, y, test_size=0.5, random_state=32, shuffle=True)
X_val = [
    'погодное условие',
    'направление ветра',
    'скорость ветра (м/с)',
    'давление (жм. рт. ст.)',
    'влажность (%)',
    'видимость (м)'
]
```

Активация Windows
Чтобы активировать Windows, перейдите в раздел "Параметры".

Ln 5. Col 16 (6 selected) Tab Size: 2 CRLF Cell 65 of 91 Go Live Total Lines: 57 ✓ Prettier 14:33 13.11.2023

File Edit Selection View Go Run Terminal Help

projects

OPEN EDITORS

exam.ipynb M Копия блокнота_ensemble_ml_algorithms_bagging_boosting_voting.ipynb simple_knn_algorithm_demo.ipynb

Изменения получены плюсике. Исходы из логического мышления предметной области. Искусственное поднятие пластового давления зависит от закачки ... # XGBoost

Code + Markdown Run All Restart Clear All Outputs Variables Outline

Rando... Аа ab * 6 of 20

PROJECTS

exam.ipynb

```
#-Случайный лес
RandomForest = RandomForestRegressor(n_estimators=100)
RandomForest.fit(X_train[X_values], y_train[y_values])
print("Тренировочный набор -- (RandomForest.score(X_train[X_values], y_train[y_values])*100)")
print("Тестовый набор -- (RandomForest.score(X_test[X_values], y_test[y_values])*100)")

#-Как видно, качество побилось, благодаря группировке данных и изучению предметной области
[468] ✓ 0.2s
...
c:\Python311\lib\site-packages\sklearn\base.py:115: DataConversionWarning:
```

A column-vector `y` was passed when a 1d array was expected. Please change the shape of `y` to `(n_samples,)`, for example using `ravel()`.

```
Тренировочный набор - 91.3069429516709
Тестовый набор - 53.437838027831084

#-Bagging
Bagging = BaggingRegressor()
Bagging.fit(X_train[X_values], y_train[y_values])
print("Тренировочный набор -- (Bagging.score(X_train[X_values], y_train[y_values])*100)")
print("Тестовый набор -- (Bagging.score(X_test[X_values], y_test[y_values])*100)")

[469] ✓ 0.0s
...
Тренировочный набор - 85.7371119844786
Тестовый набор - 45.253817042952946
c:\Python311\lib\site-packages\sklearn\ensemble\_bagging.py:59: DataConversionWarning:
```

A column-vector `y` was passed when a 1d array was expected. Please change the shape of `y` to `(n_samples,)`, for example using `ravel()`.

Активация Windows
Чтобы активировать Windows, перейдите в раздел "Параметры".

Ln 5. Col 16 (6 selected) Tab Size: 2 CRLF Cell 69 of 91 Go Live 14:34 13.11.2023

The screenshot shows the PyCharm IDE interface with two code editors open:

- AdaBoost:** The code defines an AdaBoostRegressor with 100 estimators, fits it to training data, and prints scores for both training and test sets. It also handles a DataConversionWarning.
- CatBoost:** The code defines a CatBoostRegressor with 100 estimators, fits it to training data, and prints scores for both training and test sets.

Both code snippets include explanatory comments in Russian. The bottom status bar indicates "Активация Windows" and "Чтобы активировать Windows, перейдите в раздел 'Параметры'".

The screenshot shows a Jupyter Notebook interface with several open files:

- `exam.ipynb` (active tab):
 - Code cell output:

```
# -XGBoost
XGBBoost = XGBRegressor(n_estimators=100)
XGBBoost.fit(X_train[X_values], y_train[y_values])
print(f"Тренировочный набор - {XGBBoost.score(X_train[X_values], y_train[y_values])*100}")
print(f"Тестовый набор - {XGBBoost.score(X_test[X_values], y_test[y_values])*100}")
```
 - Output:

```
Тренировочный набор - 99.999998637433
Тестовый набор - 31.968238998896464
```
- `exam.ipynb M`:
 - Code cell output:

```
learn: 1.4145319 total: 94.1ms remaining: 99.4ms
learn: 1.3619965 total: 95.9ms remaining: 88.5ms
learn: 1.3122717 total: 96.7ms remaining: 85.8ms
learn: 1.2660553 total: 97.5ms remaining: 83.1ms
learn: 1.2190874 total: 98.3ms remaining: 80.4ms
learn: 1.1791946 total: 99.2ms remaining: 77.9ms
learn: 1.1413853 total: 153ms remaining: 119ms
learn: 1.0869912 total: 154ms remaining: 118ms
learn: 1.0529958 total: 155ms remaining: 107ms
learn: 1.0156760 total: 155ms remaining: 104ms
learn: 0.9845905 total: 156ms remaining: 100ms
learn: 0.9341569 total: 158ms remaining: 96.5ms
learn: 0.8880357 total: 158ms remaining: 93ms
learn: 0.8466353 total: 159ms remaining: 89.5ms
learn: 0.8052187 total: 161ms remaining: 86.9ms
learn: 0.7654095 total: 163ms remaining: 84.2ms
learn: 0.7294471 total: 164ms remaining: 80.9ms
learn: 0.6973214 total: 165ms remaining: 77.8ms
learn: 0.6838678 total: 166ms remaining: 74.7ms
...
learn: 0.2518822 total: 245ms remaining: 2.47ms
learn: 0.2468034 total: 245ms remaining: 0us
```
 - Text:

Тренировочный набор - 99.9942511995243
Тестовый набор - 38.59452456980723
- `Копия_блокнота_ensemble_ml.ipynb`:
 - Code cell output:

```
Output is truncated. View as a scrollable element or open in a text editor. Adjust cell output settings.
```
- `simple_knn_algorithm_demo.ipynb`:
 - Code cell output:

```
Активация Windows
Чтобы активировать Windows, перейдите в раздел "Параметры".
```

На данный момент, лучшие результаты показывают алгоритмы: RandomForest и AdaBoost. Использую Feature engineering.

На данный момент, лучшие результаты показывают алгоритмы: RandomForest и AdaBoost. Использую Feature engineering.

```

plt.rcParams["figure.figsize"]=(12, 6)
plt.style.use('fivethirtyeight')
X.plot(subplots=True, sharex=True, figsize=(20, -15))
plt.show()

```

Минимальный параметр

| | count | mean | std | min | 25% | 50% | 75% | max |
|-----------------------|--------|------------|------------|-------|-------|-------|-------|--------|
| погодное условие | 1315.0 | 3.774144 | 1.310861 | 1.0 | 4.0 | 4.0 | 4.0 | 6.0 |
| направление ветра | 1315.0 | 5.161977 | 2.315407 | 1.0 | 3.0 | 5.0 | 7.0 | 8.0 |
| скорость ветра (м/с) | 1315.0 | 5.283650 | 2.805626 | 1.0 | 3.0 | 5.0 | 7.0 | 44.0 |
| давление (мм рт. ст.) | 1315.0 | 760.999240 | 10.345662 | 729.0 | 754.0 | 761.0 | 767.0 | 806.0 |
| влажность (%) | 1315.0 | 80.827376 | 15.196526 | 0.0 | 73.5 | 85.0 | 92.0 | 100.0 |
| видимость (мм) | 1315.0 | 13.486616 | 47.630362 | 1.1 | 10.0 | 10.0 | 10.0 | 915.0 |
| лун | 1315.0 | 3.387072 | 0.679097 | 1.0 | 3.0 | 3.0 | 4.0 | 4.0 |
| осадки (мм) | 1315.0 | 0.241521 | 0.914084 | 0.0 | 0.0 | 0.0 | 0.0 | 13.4 |
| температура днем | 1315.0 | -4.150570 | 13.499766 | -46.0 | -14.0 | -2.0 | 7.0 | 21.0 |
| температура ночью | 1315.0 | -2.285932 | 14.540095 | -46.0 | -12.0 | -1.0 | 10.0 | 30.0 |
| id | 1315.0 | 657.000000 | 379.752112 | 0.0 | 328.5 | 657.0 | 985.5 | 1314.0 |

Минимальный параметр

```

settings_min = settings.MinimalFCParameters()
settings_min

```

extracted_features = extract_features(X_drop, column_id="id", impute_function=impute, default_fc_parameters=settings_min)

Активация Windows
Чтобы активировать Windows, перейдите в раздел "Параметры".

File Edit Selection View Go Run Terminal Help projects

OPEN EDITORS exam.ipynb M Копия блокнота_ensemble_ml_algorithms_bagging_boosting_voting.ipynb simple_knn_algorithm_demo.ipynb

PROJECTS

exam.ipynb M

extracted_features = extract_features(X_drop, column_id="id", impute_function=impute, default_fc_parameters=settings_min)

Feature Extraction: 100% | 13150/13150 [00:05<00:00, 2628.42it/s]

extracted_features

| | погодное условие_sum_values | погодное условие_median | погодное условие_mean | погодное условие_length | погодное условие_standard deviation | погодное условие_variance | погодное условие_root mean square | погодное условие_m |
|------|-----------------------------|-------------------------|-----------------------|-------------------------|-------------------------------------|---------------------------|-----------------------------------|--------------------|
| 0 | 4.0 | 4.0 | 4.0 | 1.0 | 0.0 | 0.0 | 4.0 | |
| 1 | 6.0 | 6.0 | 6.0 | 1.0 | 0.0 | 0.0 | 6.0 | |
| 2 | 2.0 | 2.0 | 2.0 | 1.0 | 0.0 | 0.0 | 2.0 | |
| 3 | 4.0 | 4.0 | 4.0 | 1.0 | 0.0 | 0.0 | 4.0 | |
| 4 | 4.0 | 4.0 | 4.0 | 1.0 | 0.0 | 0.0 | 4.0 | |
| ... | ... | ... | ... | ... | ... | ... | ... | |
| 1310 | 5.0 | 5.0 | 5.0 | 1.0 | 0.0 | 0.0 | 5.0 | |
| 1311 | 4.0 | 4.0 | 4.0 | 1.0 | 0.0 | 0.0 | 4.0 | |
| 1312 | 4.0 | 4.0 | 4.0 | 1.0 | 0.0 | 0.0 | 4.0 | |
| 1313 | 5.0 | 5.0 | 5.0 | 1.0 | 0.0 | 0.0 | 5.0 | |
| 1314 | 4.0 | 4.0 | 4.0 | 1.0 | 0.0 | 0.0 | 4.0 | |

1315 rows × 100 columns

extracted_features.describe().T

Активация Windows чтобы активировать Windows, перейдите в раздел "Параметры" Python

File Edit Selection View Go Run Terminal Help projects

OPEN EDITORS exam.ipynb M Копия блокнота_ensemble_ml_algorithms_bagging_boosting_voting.ipynb simple_knn_algorithm_demo.ipynb

PROJECTS

exam.ipynb M

extracted_features.describe().T

| | count | mean | std | min | 25% | 50% | 75% | max |
|-------------------------------------|--------|-----------|-----------|-------|-------|------|------|------|
| погодное условие_sum_values | 1315.0 | 3.774144 | 1.310861 | 1.0 | 4.0 | 4.0 | 4.0 | 6.0 |
| погодное условие_median | 1315.0 | 3.774144 | 1.310861 | 1.0 | 4.0 | 4.0 | 4.0 | 6.0 |
| погодное условие_mean | 1315.0 | 3.774144 | 1.310861 | 1.0 | 4.0 | 4.0 | 4.0 | 6.0 |
| погодное условие_length | 1315.0 | 1.000000 | 0.000000 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 |
| погодное условие_standard deviation | 1315.0 | 0.000000 | 0.000000 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| ... | ... | ... | ... | ... | ... | ... | ... | ... |
| температура ночью_variance | 1315.0 | 0.000000 | 0.000000 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| температура ночью_mean_square | 1315.0 | 11.923954 | 8.622931 | 0.0 | 5.0 | 11.0 | 17.0 | 46.0 |
| температура ночью_maximum | 1315.0 | -2.285932 | 14.540095 | -46.0 | -12.0 | -1.0 | 10.0 | 30.0 |
| температура ночью_absolute_maximum | 1315.0 | 11.923954 | 8.622931 | 0.0 | 5.0 | 11.0 | 17.0 | 46.0 |
| температура ночью_minimum | 1315.0 | -2.285932 | 14.540095 | -46.0 | -12.0 | -1.0 | 10.0 | 30.0 |

100 rows × 8 columns

#-доля-пропусков
#-проверка-параметры-на-0.-Если-вся-строка-0---офильтрована-

extracted_features.isnull().sum().sum()

0

#-сматр-пропуски
extracted_features.isnull().mean()

Активация Windows чтобы активировать Windows, перейдите в раздел "Параметры" Python

The screenshot shows a Jupyter Notebook interface with several open files. The current file, `exam.ipynb`, contains Python code related to weather data analysis. The code includes calculations for various weather conditions like temperature and wind speed, and a section for training and testing a machine learning model. The output pane displays the results of these operations, including numerical values and printed statements.

```
#-словарь-пропуски
extracted_features.isnull().mean()
✓ 0.0s

... погодное_условие_sum_values 0.0
погодное_условие_median 0.0
погодное_условие_mean 0.0
погодное_условие_length 0.0
погодное_условие_standard_deviation 0.0

температура_ночье_variance ... 0.0
температура_ночье_root_mean_square 0.0
температура_ночье_maximum 0.0
температура_ночье_absolute_maximum 0.0
температура_ночье_minimum 0.0
Length: 100, dtype: float64

#-БЕЗ-ГРУППИРОВКИ
#-разделение-данных - на-тестовую-(20%) и-тренировочную-выборки -(80%)
X_train_ext, X_test_ext, y_train_ext, y_test_ext = train_test_split(extracted_features, y_original_1, test_size=0.2, random_state=1024, shuffle=True)
X_test_ext, X_val_ext, y_test_ext, y_val_ext = train_test_split(extracted_features, y_original_1, test_size=0.5, random_state=32, shuffle=True)

#-X_values--{
#...погодное-условие',
#...направление-ветра',
#...скорость-ветра-(м/с)',
#...давление-(мм.рт.-см.г'),
#...влажность-(%),
#...видимость-(мм),
#...луна',
#...осадки-(мм),
#...температура-днем',
#...температура-ночью'
#}
y_values = [
    ...' пластовое давление -(атм)'
]

Активация Windows
Чтобы активировать Windows, перейдите в раздел "Параметры".
```

The screenshot shows a Jupyter Notebook interface with several open files. The current file, `exam.ipynb`, contains Python code related to weather data analysis. The code includes calculations for various weather conditions like temperature and wind speed, and a section for training and testing a machine learning model. The output pane displays the results of these operations, including numerical values and printed statements.

```
#-Случайный лес
forest = RandomForestRegressor(n_estimators=100)
forest.fit(X_train_ext, y_train_ext[y_values])
print(f"Тренировочный набор - {forest.score(X_train_ext, y_train_ext[y_values])*100}")
print(f"Тестовый набор - {forest.score(X_test_ext, y_test_ext[y_values])*100}")

A column-vector y was passed when a 1d array was expected. Please change the shape of y to (n_samples,), for example using ravel().

Тренировочный набор - 91.25749337651345
Тестовый набор - 79.65666408586509

#-С-ГРУППИРОВКОЙ
X_drop = X
X_drop['id'] = X_drop.index
X_drop = X_drop.drop(['день'], axis=1)

extracted_features = extract_features(X_drop, column_id='id', impute_function=impute, default_fc_parameters=settings_min)

#-разделение-данных - на-тестовую-(20%) и-тренировочную-выборки -(80%)
X_train_ext, X_test_ext, y_train_ext, y_test_ext = train_test_split(extracted_features, y, test_size=0.2, random_state=1024, shuffle=True)
X_test_ext, X_val_ext, y_test_ext, y_val_ext = train_test_split(extracted_features, y, test_size=0.5, random_state=32, shuffle=True)

#-X_values--{
#...погодное-условие',
#...направление-ветра',
#...скорость-ветра-(м/с)',
#...давление-(мм.рт.-см.г'),
#...влажность-(%),
#...видимость-(мм),
#...луна',
#...осадки-(мм),
#...температура-днем',
#...температура-ночью'
#}
y_values = [
    ...
```

The screenshot shows a Jupyter Notebook interface with several open files. The current file, `exam.ipynb`, contains Python code for feature engineering and machine learning. The code includes imports for `RandomForestRegressor`, data loading from CSV files, and printing scores for training and testing datasets. A warning message about column-vector conversion is visible.

```
#-X_values = [
#    ...погодные условия',
#    ...направление ветра',
#    ...скорость ветра (м/с)',
#    ...давление (мм рт. см.)',
#    ...влажность (%)',
#    ...видимость (мкм)',
#    ...луна',
#    ...осадки (мм)',
#    ...температура днем',
#    ...температура ночью'
#]

y_values = [
    'давление давление (атм)'
]

X_train_original_2 = X_train_ext
y_train_original_2 = y_train_ext
X_test_original_2 = X_test_ext
y_test_original_2 = y_test_ext

# Случайный лес
forest = RandomForestRegressor(n_estimators=100)
forest.fit(X_train_ext, y_train_ext[y_values])
print("Тренировочный набор -- (forest.score(X_train_ext, y_train_ext[y_values])*100)")
print("Тестовый набор -- (forest.score(X_test_ext, y_test_ext[y_values])*100)")

0.5s

... Feature Extraction: 100% | 440/440 [00:00:00, 3098.50it/s]
c:\Python311\lib\site-packages\sklearn\base.py:115: DataConversionWarning:
A column-vector y was passed when a 1d array was expected. Please change the shape of y to (n_samples,), for example using ravel().
Тренировочный набор - 91.01880118964779
Тестовый набор - 86.77089427483158
```

Получил показатели благодаря изучению предметной области, группировки данных по году и месяцу, а также применению Feature engineering:

Тренировочный набор - 90.34289677151644

Тестовый набор - 67.62513386375784

The screenshot shows a Jupyter Notebook interface with several open files. The current file, `exam.ipynb`, contains Python code for feature extraction and model evaluation. The code includes a list of extracted features and prints the score for the test dataset.

```
#-Список признаков с бесами
feature_names_str = [f"col{col}" for col in extracted_features.columns.tolist()]
cols = [col.replace("col", "") for col in feature_names_str[12]]
cols

['погодное условие_sum_values',
 'погодное условие_median',
 'погодное условие_mean',
 'погодное условие_length',
 'погодное условие_standard deviation',
 'погодное условие_variance',
 'погодное условие_root_mean_square',
 'погодное условие_maximum',
 'погодное условие_absolute_maximum',
 'погодное условие_minimum',
 'направление ветра_sum_values',
 'направление ветра_median']
```

File Edit Selection View Go Run Terminal Help projects

OPEN EDITORS exam.ipynb M Копия блокнота_ensemble_ml_algorithms_bagging_boosting_voting.ipynb simple_knn_algorithm_demo.ipynb

PROJECTS

exam.ipynb

```
# Вес каждого фактора в итоговой модели
forest.feature_importances_
```

array([0.00709237, 0.01445549, 0.00860452, 0., 0., 0., 0.00370888, 0.0041815, 0.0128175, 0.00660522, 0.0080761, 0.00263896, 0.00654576, 0., 0., 0.00715512, 0.00610137, 0.00186508, 0.00179876, 0.00902458, 0.00242369, 0.02332305, 0., 0., 0., 0.00847517, 0.00388107, 0.00381333, 0.01510916, 0.07937283, 0.06207352, 0.04651378, 0., 0., 0., 0.05530108, 0.07372314, 0.02197713, 0.10749412, 0.00342952, 0.00349145, 0.0054071, 0., 0., 0., 0.00431554, 0.00498524, 0.00342452, 0.00381631, 0.01561651, 0.01702696, 0.01272034, 0., 0., 0., 0.00436425, 0.00406651, 0.01472823, 0.01136417, 0.00478165, 0.00679205, 0.00589771, 0., 0., 0., 0.011247281, 0.0055905, 0.00964953, 0.01036224, 0.01616308, 0.00692845, 0.02885331, 0., 0., 0., 0.02038833, 0.01429548, 0.0182876, 0.02421432, 0.00611068, 0.004089541, 0.00770327, 0., 0., 0., 0., 0.00814167, 0.00965339, 0.00999757, 0.00452676, 0.00152346, 0.01166093, 0.00637792, 0., 0., 0., 0., 0.08862226, 0.08870803, 0.01606922, 0.00957782])

```
plot = pd.Series(data=forest.feature_importances_).plot(kind='bar')
plot.tick_params(axis='x', rotation=90)
```

OUTLINE

TIMELINE

File Edit Selection View Go Run Terminal Help projects

OPEN EDITORS exam.ipynb M Копия блокнота_ensemble_ml_algorithms_bagging_boosting_voting.ipynb simple_knn_algorithm_demo.ipynb

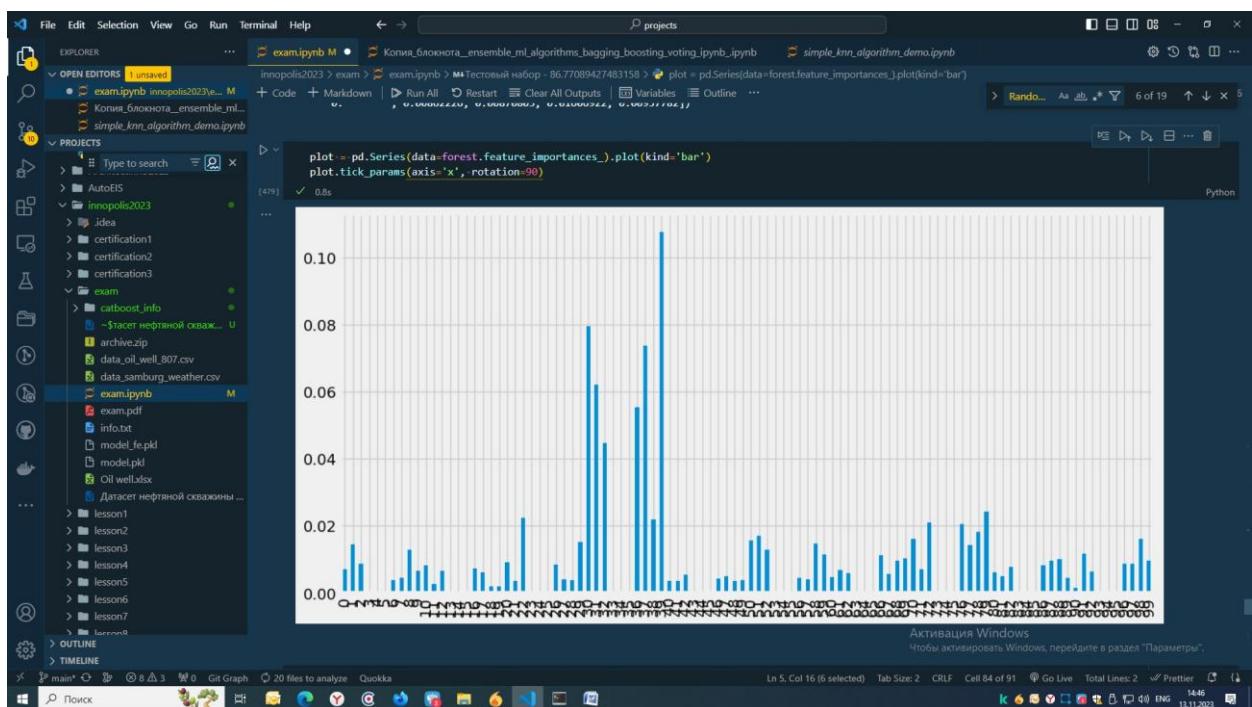
PROJECTS

exam.ipynb

plot = pd.Series(data=forest.feature_importances_).plot(kind='bar')

plot.tick_params(axis='x', rotation=90)

Активация Windows
Чтобы активировать Windows, перейдите в раздел "Параметры".



Этап 5. Сохранение и загрузка моделей.

```
# Сохранение модели
joblib.dump(RandomForest, 'model.pkl') #-без Feature engineering
joblib.dump(forest, 'model_fe.pkl') #-с Feature engineering

# Загрузку модели
RandomForest = joblib.load('model.pkl') #-без Feature engineering
forest = joblib.load('model_fe.pkl') #-с Feature engineering

# Используя загруженные модели для предсказания
#-без Feature engineering
#-Тренировочный набор - 91.2553105344086
#-Тестовый набор - 88.85372762794663
X_test_pred = X_test_original_1.drop(['день'], axis=1)
y_pred = RandomForest.predict(X_test_pred)
y_pred

# Используя загруженные модели для предсказания
#-с Feature engineering
#-Тренировочный набор - 91.2553105344086
#-Тестовый набор - 88.85372762794663
y_pred = forest.predict(X_test_original_2)
y_pred
```

```
#-Прикрепляю обработка собственных данных
#-без Feature engineering
y_pred = RandomForest.predict([[
    -1, #посадочное устье
    -1, #напоротечение - вода
    -1, #скорость - вода (м/с)
    -1, #давление - (мм рт. - см.)
    -1, #влажность - (%)
    -1, #видимость - (м)
    -1, #луна
    -1, #осадки - (мм)
    -1, #температура - днем
    -1, #температура - ночь
]])
y_pred

... c:\Python311\lib\site-packages\sklearn\base.py:465: UserWarning:
X does not have valid feature names, but RandomForestRegressor was fitted with feature names

... array([120.63871613])
```

Итог

Задав новые данные погодных условий. Получаю результат пластового давления. Это давление влияет на увеличение давления нефти и может привести к поломке оборудования нефтяной скважины №807.

Так как село Самбург находится в 24 км от скважины №807, результаты могут быть с отклонениями. Для более точного анализа и построения модели, необходимо снимать погодные показания непосредственно рядом с нефтяной скважиной.

Список приложенных файлов:

«exam.ipynb» – для Google Colab

«exam.py» – исходный код для Python

«Oil well.xlsx» – датасет нефтяной скважины №807 (оригинал)

«data_oil_well_807.csv» – датасет нефтяной скважины №807 (парсер)

«data_samburg_weather.csv» – датасет погоды с. Самбург (парсер)

«model.pkl» – обученная модель без Feature engineering

«model_fe.pkl» – обученная модель с Feature engineering

Ссылка на Notebook Colab:

https://colab.research.google.com/drive/1hMPW8lfmeuqPYqFc0jSy_gpjgVFBLkR9?usp=sharing

Ссылка на GitHub:

<https://github.com/SotGE/innopolis2023/tree/main/exam>