



UMCS

CENTRUM SZTUCZNEJ INTELIGENCJI
I MODELOWANIA KOMPUTEROWEGO

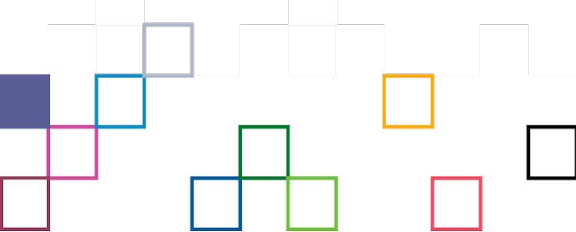
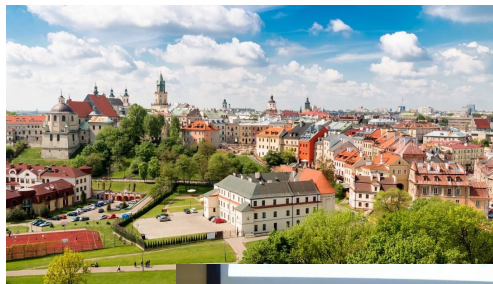
Large Language Models (LLMs) in Science: Pros and Cons

dr Kamil Filipek

Centre for Artificial Intelligence and Computational Modeling
Marii Curie-Skłodowska University in Lublin

OSLO, 3-4.10.2024

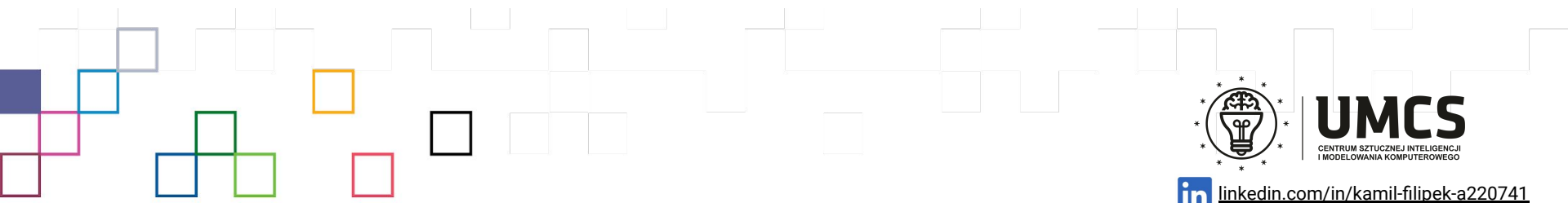
Our place



Plan

1. LLMs Now and Future
2. Setting LLM's Envi
3. Prompting Effectively
 - a. General Tips
 - b. Specific Tips
4. Quantitative Analysis
 - a. Open Source Data
 - b. Code Assessment and Interpretation
5. Future Use

LLMs: Now and Future



LLMs: Now and Future

LLM - Large Language Model

1. Generative: This refers to the model's ability to generate new content
2. Pre-trained: The model is initially trained on a large corpus of text data
3. Transformer: This refers to the specific architecture of the neural network used in GPT models

2017

Attention Is All You Need

Ashish Vaswani*
Google Brain
avaswani@google.com

Noam Shazeer*
Google Brain
noam@google.com

Niki Parmar*
Google Research
nikip@google.com

Jakob Uszkoreit*
Google Research
uszko@google.com

Llion Jones*
Google Research
llion@google.com

Aidan N. Gomez*¹
University of Toronto
aidan@cs.toronto.edu

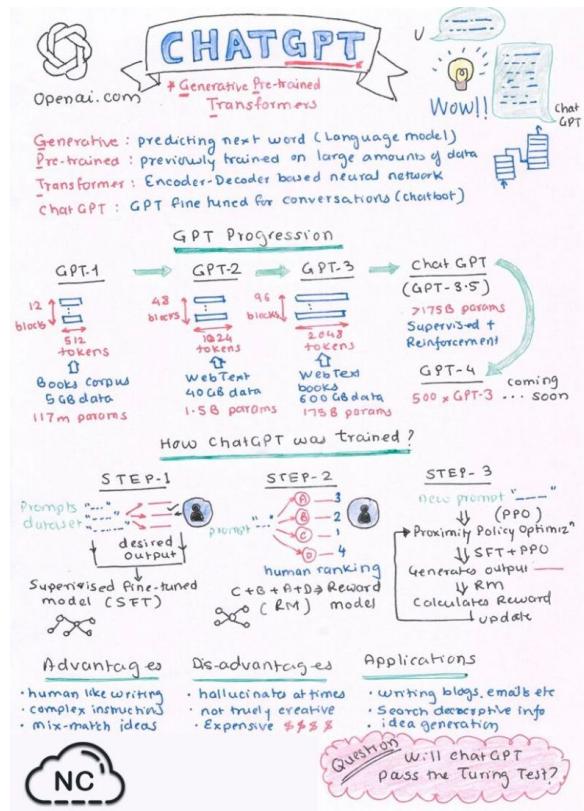
Lukas Kaiser*
Google Brain
lukasz.kaiser@google.com

Ilya Polosukhin*¹
iliiia.polosukhin@gmail.com

BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding

Jacob Devlin Ming-Wei Chang Kenton Lee Kristina Toutanova
Google AI Language
{jacobdevlin,mingweichang,kentonl,kristout}@google.com

2018



LLMs: Now and Future

Close-source LLMs



<https://chatgpt.com/>



<https://gemini.google.com/app>



<https://claude.ai/new>

BY ANTHROPIC



UMCS
CENTRUM SZTUCZNEJ INTELIGENCJI
I MODELOWANIA KOMPUTEROWEGO



linkedin.com/in/kamil-filipek-a220741

LLMs: Now and Future

Open-source LLMs



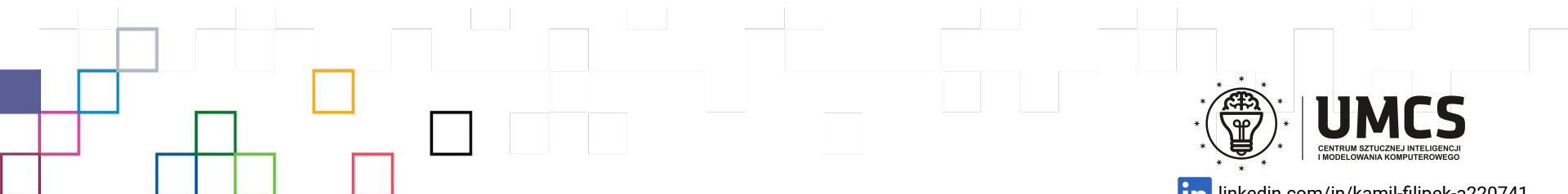
Hugging Face

<https://huggingface.co/models?sort=trending&search=LLM>






Example
implementations

https://hub.analytics.umcs.pl/user/kfilipek/lab/tree/shared/Chat_GPT/LLM/Dolly.ipynb



LLMs: Now and Future

Close-source LLMs vs. open-source LLMs

Features	Open-Source LLMs	Close-Source LLMs
		
Cost	Lower initial cost but possibly higher maintenance.	Higher initial cost but lower maintenance.
Innovation	Collaborative and fast innovation from a global community.	Slower innovation directed by a single entity.
Support	Reliant on community-based support, which can be extensive but inconsistent.	Consistent and quality support offered through dedicated channels.
Customization	Highly customizable with access to source code.	Customization possible within the platform's limits.
IP	Shared IP under open-source licenses.	IP is closely held, providing competitive advantage.

SOURCE: <https://datasciencedojo.com/blog/open-source-llm/>

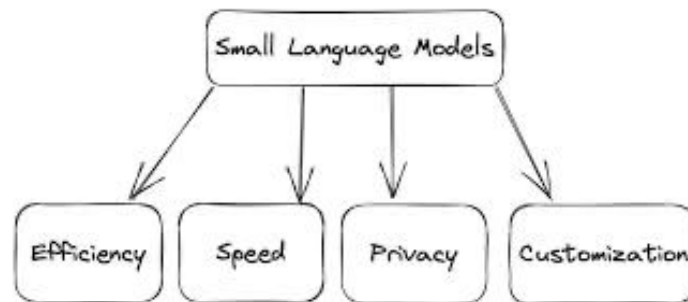
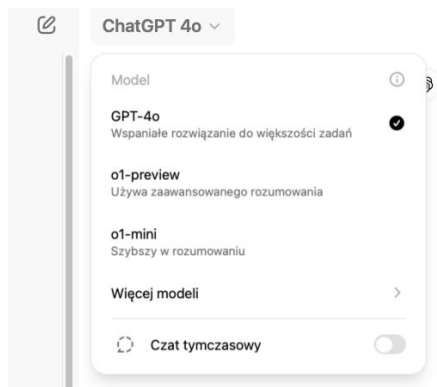


UMCS
CENTRUM SZTUCZNEJ INTELIGENCJI
I MODELOWANIA KOMPUTEROWEGO



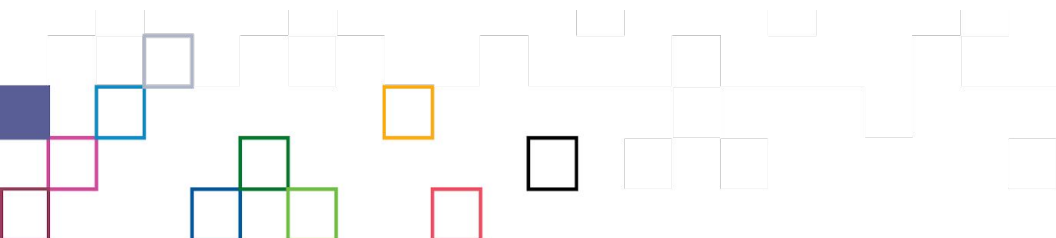
[linkedin.com/in/kamil-filipek-a220741](https://www.linkedin.com/in/kamil-filipek-a220741)

LLMs: Now and Future



MINI: <https://chatgpt.com/share/66f5505a-2c88-8002-95db-b46800aa3bd9>

GPTo: <https://chatgpt.com/share/66f550ca-8474-8002-b17e-0752f079f97e>

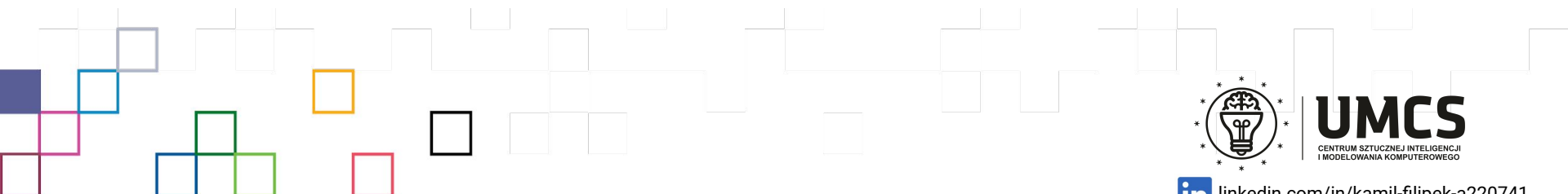


LLMs: Now and Future

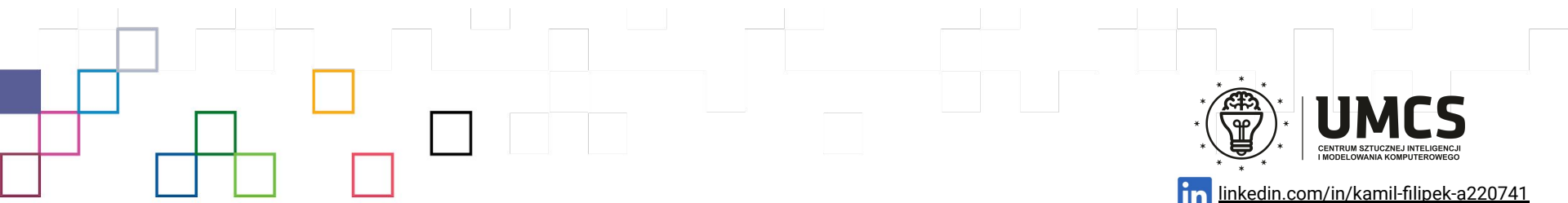
.... Please propose prompt to compare the models.

MINI: <https://chatgpt.com/share/66f5505a-2c88-8002-95db-b46800aa3bd9>

GPTo: <https://chatgpt.com/share/66f550ca-8474-8002-b17e-0752f079f97e>



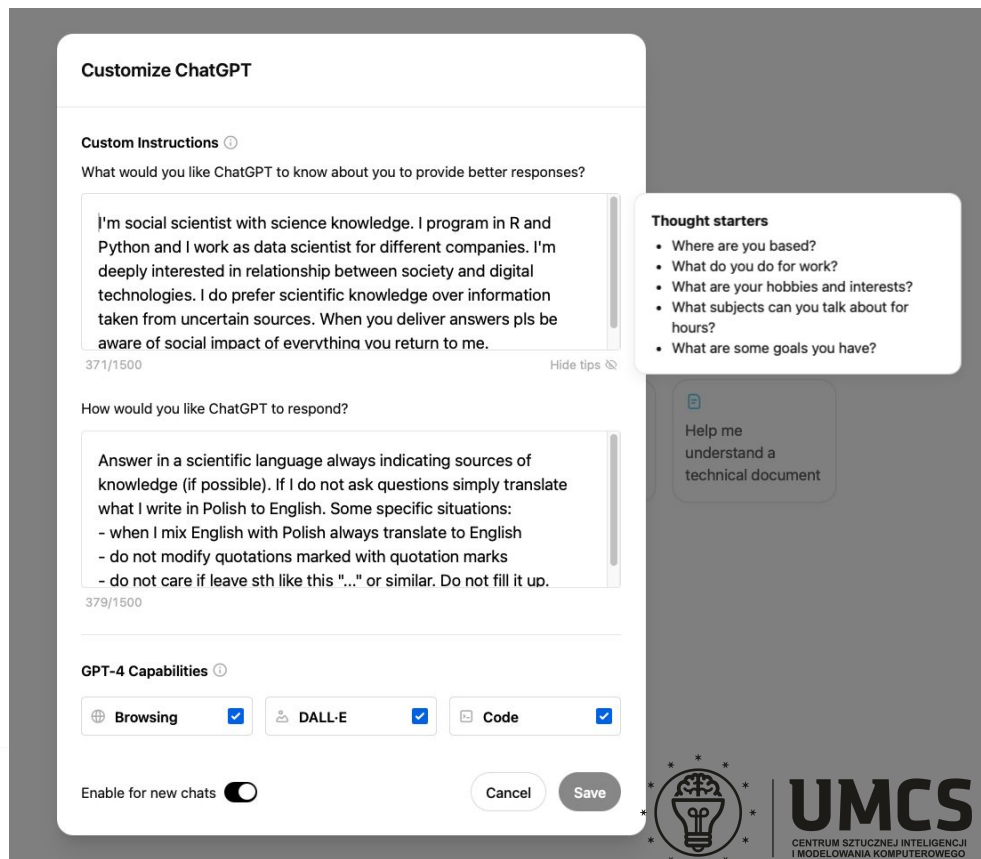
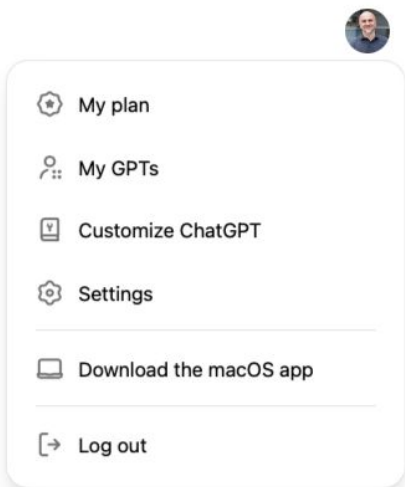
Setting LLM's Envi



UMCS
CENTRUM SZTUCZNEJ INTELIGENCJI
I MODELOWANIA KOMPUTEROWEGO


ChatGPT Envi

Customize ChatGPT



ChatGPT Envi

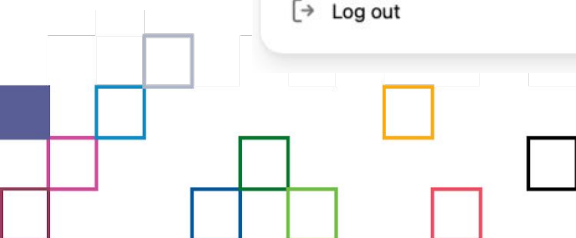
Settings



- My plan
- My GPTs
- Customize ChatGPT
- Settings
- Download the macOS app
- Log out

Settings

- General**
 - Theme: System
- Personalization
 - Always show code when using data analyst:
- Speech
- Data controls
 - Language: English (US)
- Builder profile
 - Archived chats: [Manage](#)
- Connected apps
- Security
 - Archive all chats: [Archive all](#)
 - Delete all chats: [Delete all](#)



ChatGPT Envi

☰ Explore GPTs

Plugins/Extensions

GPTs







Discover and create custom versions of ChatGPT that combine instructions, extra knowledge, and any combination of skills.

🔍 Search GPTs

Najczęściej wybierane Writing Productivity Research & Analysis Education Lifestyle Program →

Research & Analysis

Find, evaluate, interpret, and visualize information

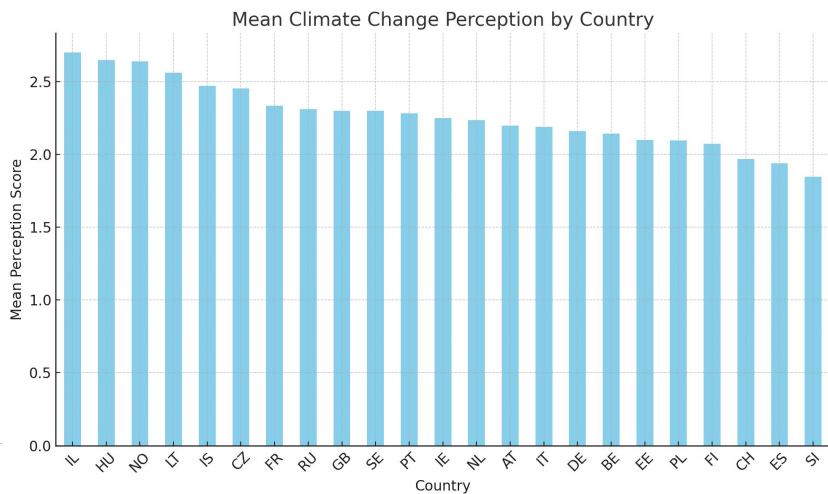
-  **Scholar GPT**
Enhance research with 200M+ resources and built-in critical reading skills. Access Google Scholar, PubMed, JSTOR, Arxiv...
By awesomegpts.ai
-  **Consensus**
Ask the research, chat directly with the world's scientific literature. Search references, get simple explanations, ...
By consensus.app
-  **SciSpace**
Do hours worth of research in minutes. Instantly access 287M+ papers, analyze papers at lightning speed, and...
By scispace.com
-  **Excel AI**
🌱 The worlds most powerful data analysis assistant. 🌱
By pulsar.co.uk
-  **Scholar AI**
AI Innovator — search and review 200M+ scientific papers, patents, and books. Research literature, discover insights, ...
By scholarai.io
-  **Video Summarizer**
Youtube Video summarizer | video summaries, chat with Youtube video
By thegeneralmind.com

See more



ChatGPT Envi

Plugins/Extensions



Data Analysis & Report AI

By Kenneth G Bastian

Limitless, detailed data analysis & reporting with charts, graphs, and insights.



ESS Data Portal

Search for ESS data e.g. trust politicians, election



Log in

Menu

Datafile Builder (Wizard)

Create your own datafile. Select rounds and countries, plus the variables you want to download.

Select Country/year

Select Variables

Download

	1	2	3	4	5	6	7	8	9	10	10SC	11	All
All	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

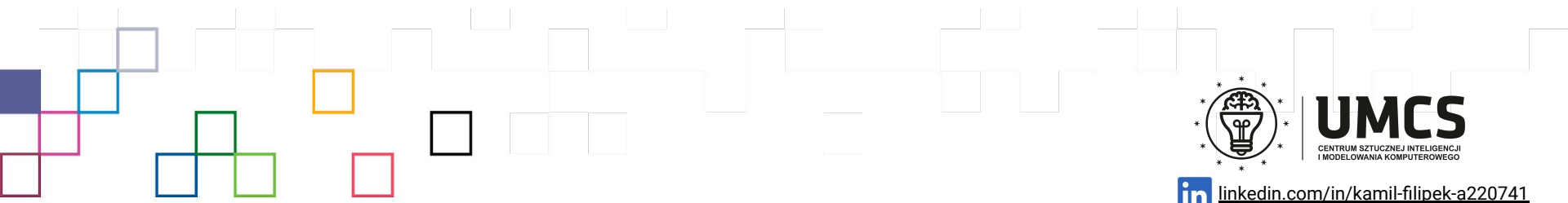


UMCS
CENTRUM SZTUCZNEJ INTELIGENCJI
I MODELOWANIA KOMPUTEROWEGO



[linkedin.com/in/kamil-filipek-a220741](https://www.linkedin.com/in/kamil-filipek-a220741)

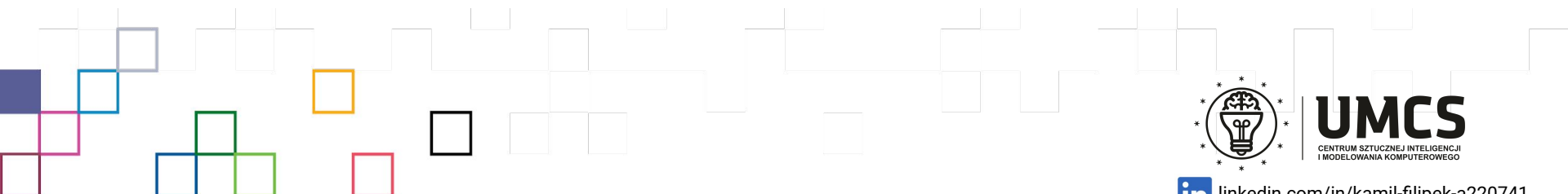
Prompting Effectively



UMCS
CENTRUM SZTUCZNEJ INTELIGENCJI
I MODELOWANIA KOMPUTEROWEGO

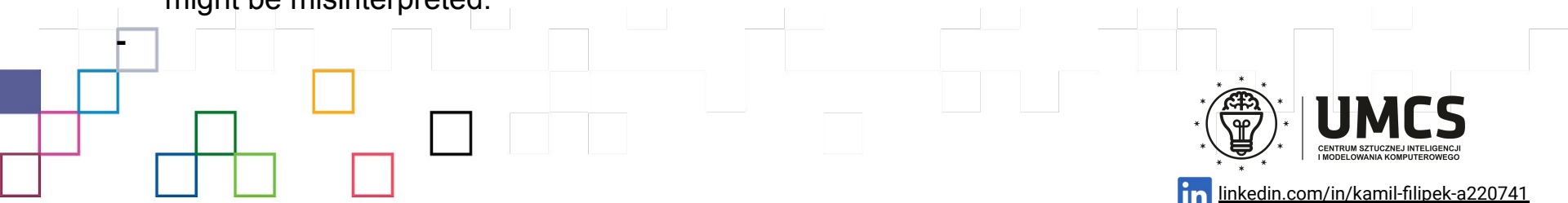
General Tips

- **Be specific and clear:** Provide detailed instructions and context for what you want. The more specific you are, the better the results.
- **Break complex tasks into smaller steps:** For multi-part tasks, break them down and ask for one part at a time.
- **Use examples:** Providing examples of the kind of output you're looking for can be very helpful.
- **Specify the desired format:** If you want the response in a particular format (e.g. bullet points, paragraphs, code), state that explicitly.
- **Ask for reasoning:** If you want to understand the thought process, ask the AI to explain its reasoning.
- **Be iterative:** If the first response isn't quite what you wanted, refine your prompt and try again.



General Tips

- **Use system prompts or role-playing:** Setting up a specific role or context can help guide the AI's responses.
- **Encourage creativity when appropriate:** For creative tasks, explicitly state if you want novel or unusual ideas.
- **Specify constraints:** If there are limitations or rules the response should follow, state them clearly.
- **Ask for alternative viewpoints:** To get a balanced perspective, you can ask the AI to consider different angles on a topic.
- **Use clear language:** Avoid ambiguity and use straightforward language in your prompts.
- **Proofread your prompts:** Ensure your prompts are free of typos and grammatical errors that might be misinterpreted.



Good prompt

Imagine you are a climate scientist preparing a report on the impacts of rapid climate change on Earth's ecosystems. Your goal is to communicate the severity of these changes to both policymakers and the general public in a clear, scientific manner. For this, you will create a section of your report that focuses on a specific example of ecosystem disruption due to rising global temperatures.

Instructions: Step 1: Choose one ecosystem (e.g., coral reefs, Arctic tundra, Amazon rainforest). Step 2: Describe in detail how rising global temperatures are disrupting this ecosystem. Be specific about both the direct (e.g., habitat loss, temperature stress) and indirect (e.g., migration patterns, food web changes) impacts. Step 3: Explain the broader consequences for the planet if this disruption continues unchecked (e.g., biodiversity loss, impacts on human populations). Step 4: Offer one solution or adaptation strategy that could help mitigate the impact on this ecosystem, and explain the scientific reasoning behind it.

Desired Format: Please provide this information in a structured report format with bullet points or numbered lists for clarity. Encouraged Creativity: Feel free to include potential novel ideas for adaptation strategies that are currently being researched or proposed by scientists. Constraint: Use only scientific research and avoid speculative or pseudoscientific claims. Be sure to back up your points with examples of real-world data and findings when possible.



UMCS
CENTRUM SZTUCZNEJ INTELIGENCJI
I MODELOWANIA KOMPUTEROWEGO



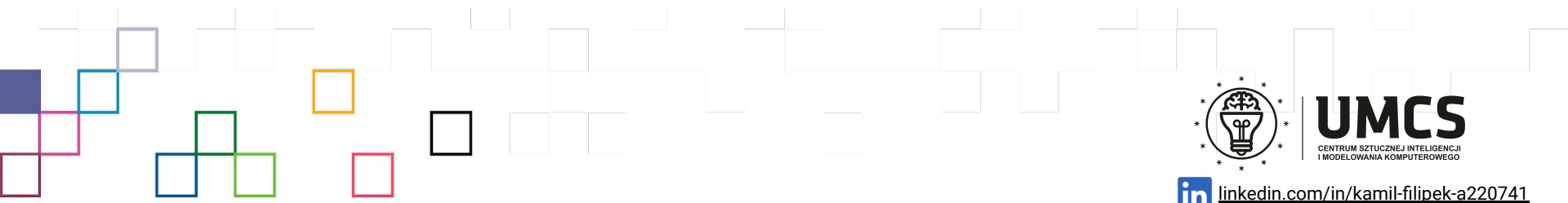
[linkedin.com/in/kamil-filipek-a220741](https://www.linkedin.com/in/kamil-filipek-a220741)

Good prompt

Imagine you are a climate scientist preparing a report on the impacts of rapid climate change on Earth's ecosystems. Your goal is to communicate the severity of these changes to both policymakers and the general public in a clear, scientific manner. For this, you will create a section of your report that focuses on a specific example of ecosystem disruption due to rising global temperatures. Instructions: Step 1: Choose one ecosystem (e.g., coral reefs, Arctic tundra, Amazon rainforest). Step 2: Describe in detail how rising global temperatures are disrupting this ecosystem. Be specific about both the direct (e.g., habitat loss, temperature stress) and indirect (e.g., migration patterns, food web changes) impacts. Step 3: Explain the broader consequences for the planet if this disruption continues unchecked (e.g., biodiversity loss, impacts on human populations). Step 4: Offer one solution or adaptation strategy that could help mitigate the impact on this ecosystem, and explain the scientific reasoning behind it. Desired Format: Please provide this information in a structured report format with bullet points or numbered lists for clarity. Encouraged Creativity: Feel free to include potential novel ideas for adaptation strategies that are currently being researched or proposed by scientists. Constraint: Use only scientific research and avoid speculative or pseudoscientific claims. Be sure to back up your points with examples of real-world data and findings when possible.

Good prompt

Let's type the prompt...



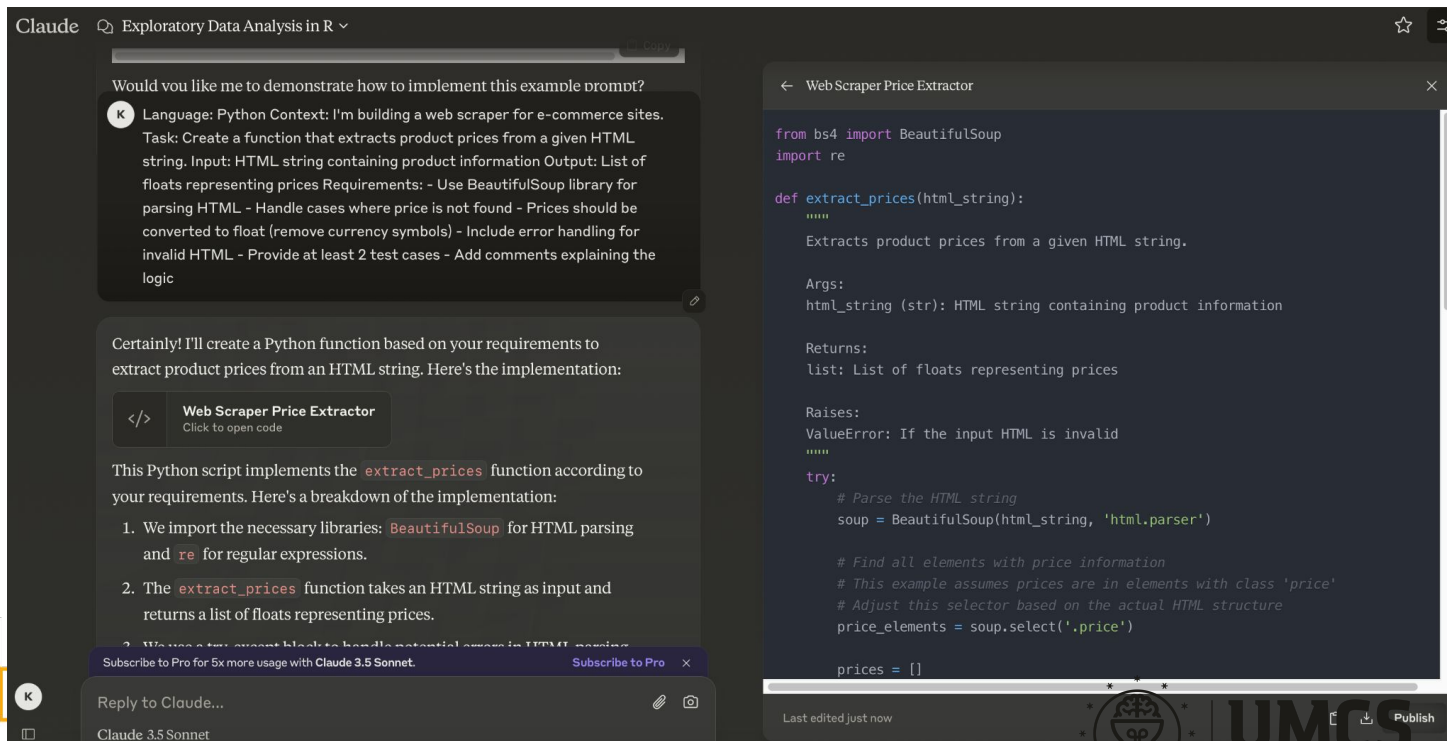
Specific Tips

For programming

- Language: Python Context: I'm building a web scraper for e-commerce sites.
- Task: Create a function that extracts product prices from a given HTML string.
- Input: HTML string containing product information
- Output: List of floats representing prices
- Requirements:
 - Use BeautifulSoup library for parsing HTML
 - Handle cases where price is not found
 - Prices should be converted to float (remove currency symbols)
 - Include error handling for invalid HTML
 - Provide at least 2 test cases
 - Add comments explaining the logic

Specific Tips

Real tests



Claude Exploratory Data Analysis in R

Would you like me to demonstrate how to implement this example prompt?

K Language: Python Context: I'm building a web scraper for e-commerce sites. Task: Create a function that extracts product prices from a given HTML string. Input: HTML string containing product information Output: List of floats representing prices Requirements: - Use BeautifulSoup library for parsing HTML - Handle cases where price is not found - Prices should be converted to float (remove currency symbols) - Include error handling for invalid HTML - Provide at least 2 test cases - Add comments explaining the logic

Certainly! I'll create a Python function based on your requirements to extract product prices from an HTML string. Here's the implementation:



```
</> Web Scraper Price Extractor  
Click to open code
```

This Python script implements the `extract_prices` function according to your requirements. Here's a breakdown of the implementation:

1. We import the necessary libraries: `BeautifulSoup` for HTML parsing and `re` for regular expressions.
2. The `extract_prices` function takes an HTML string as input and returns a list of floats representing prices.


```
from bs4 import BeautifulSoup  
import re  
  
def extract_prices(html_string):  
    """  
    Extracts product prices from a given HTML string.  
  
    Args:  
    html_string (str): HTML string containing product information  
  
    Returns:  
    list: List of floats representing prices  
  
    Raises:  
    ValueError: If the input HTML is invalid  
    """  
    try:  
        # Parse the HTML string  
        soup = BeautifulSoup(html_string, 'html.parser')  
  
        # Find all elements with price information  
        # This example assumes prices are in elements with class 'price'  
        # Adjust this selector based on the actual HTML structure  
        price_elements = soup.select('.price')  
  
        prices = []
```



Subscribe to Pro for 5x more usage with Claude 3.5 Sonnet. [Subscribe to Pro](#)

Reply to Claude...  

Claude 3.5 Sonnet

Last edited just now

 **UMCS**
CENTRUM SZTUCZNEJ INTELIGENCJI
I MODELWANIA KOMPUTEROWEGO

 linkedin.com/in/kamil-filipek-a220741 

Specific Tips

Real tests

ChatGPT 4o Share

Here's a Python function using BeautifulSoup to extract product prices from an HTML string. The function will remove currency symbols and convert prices to floats. It also includes error handling for invalid HTML and cases where prices aren't found.

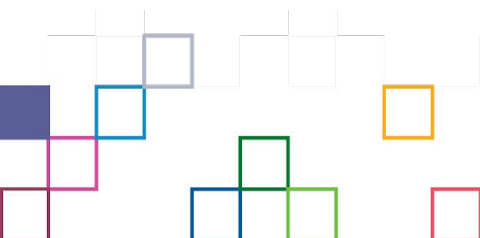
```
python Copy code

from bs4 import BeautifulSoup
import re

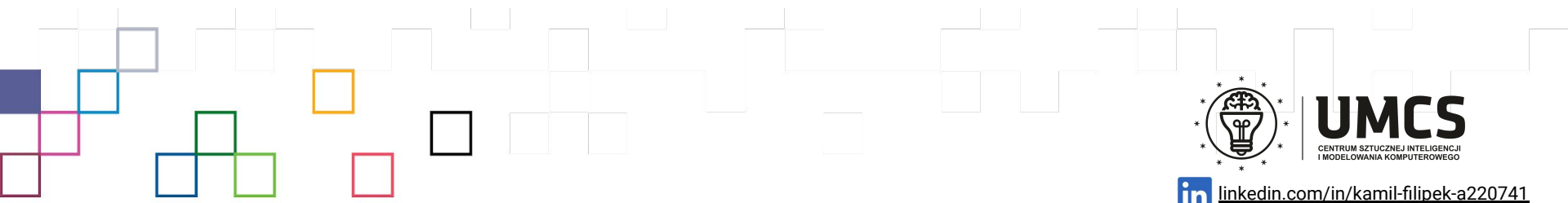
def extract_prices(html_string):
    """
    Extract product prices from a given HTML string.

    Args:
        html_string (str): The input HTML containing product information.

    Returns:
        List[float]: A list of floats representing the prices.
    """
    prices = []
    try:
        # Parse the HTML string using BeautifulSoup
```



Quantitative Analysis



UMCS
CENTRUM SZTUCZNEJ INTELIGENCJI
I MODELOWANIA KOMPUTEROWEGO

Open Source Data

USGS
science for a changing world

EarthExplorer

System Notification (1) Help Feedback Login

Important Notice: All NASA LP DAAC data products will be removed from EarthExplorer and M2M on August 30, 2024. Users are encouraged to visit the LP DAAC website to familiarize themselves with alternative search and download options. A list of available tools for each dataset are included under the "Access Data" button on each dataset's DOI landing page, these landing pages can be found using the Search Data Catalog. Please see the following news announcement to learn more: <https://lpdaac.usgs.gov/news/removal-of-nasa-lp-daac-products-from-usgs-earthexplorer-and-machine-to-machine-api-on-august-30-2024/> Please email lpdaac@usgs.gov with any questions.

Search Criteria Data Sets Additional Criteria Results

2. Select Your Data Set(s)

Check the boxes for the data set(s) you want to search. When done selecting data set(s), click the **Additional Criteria** or **Results** buttons below. Click the plus sign next to the category name to show a list of data sets.

Use Data Set Prefilter (What's This?)

Data Set Search:

This data set list is cached for performance. If your user permissions have changed or you are not seeing an expected dataset, [click here to refresh your list.](#)

- Aerial Imagery
- AVHRR
- CEOS Legacy
- Commercial Satellites
- Declassified Data
- Digital Elevation
- Digital Line Graphs
- Digital Maps
- EO-1
- Global Fiducials
- HCMM
- IBERV
- Land Cover
- Landcat
- LCMAP
- Radar

Search Criteria Summary (Show) Clear Search Criteria

68° 54' 39" N, 24° 12' 11" E Options + -

Norwegian Sea

Norwegian Sea

SWEDEN

NORWAY

FINLAND

ESTONIA

Murmansk

Arkhangel'sk

Oslo

Stockholm

Tallinn

Heiljnu

Reykjavik

Saint-Petersburg

Leen Loojoga

<https://earthexplorer.usgs.gov/>

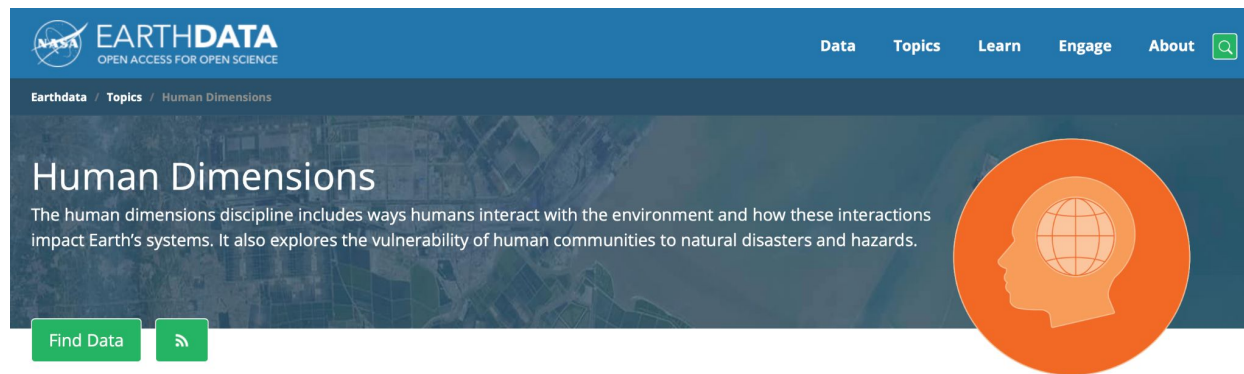


UMCS
CENTRUM SZTUCZNEJ INTELIGENCJI
I MODELWANIA KOMPUTEROWEGO



[linkedin.com/in/kamil-filipek-a220741](https://www.linkedin.com/in/kamil-filipek-a220741)

Open Source Data



The screenshot shows the NASA EarthData website. At the top left is the NASA logo and the text "EARTHDATA OPEN ACCESS FOR OPEN SCIENCE". To the right are navigation links: "Data", "Topics", "Learn", "Engage", and "About", followed by a search icon. Below the navigation is a breadcrumb trail: "Earthdata / Topics / Human Dimensions". The main heading is "Human Dimensions" in large white text. Below it is a paragraph: "The human dimensions discipline includes ways humans interact with the environment and how these interactions impact Earth's systems. It also explores the vulnerability of human communities to natural disasters and hazards." To the right of the text is a circular orange graphic containing a white silhouette of a human head in profile, with a globe inside. At the bottom left of the main content area are two green buttons: "Find Data" and a social media icon.

Human activities, such as the conversion of natural land to built-up areas, the cultivation of agricultural land, and the extent and type of agricultural production, movement and consumption of natural resources, can be tracked by sensors aboard Earth observing satellites. The combination of remotely sensed data with data collected by national and sub-national government agencies (such as census, disease, species diversity, and similar data) enables investigations into the impact of human activities on Earth. Additionally, these data are a vital resource for managing disaster response, such as using nighttime lights data to monitor power outages following storms or thermal anomaly data to track the movement of wildfires near populated areas.

<https://www.earthdata.nasa.gov/topics/human-dimensions>

Open Source Data

	site	project_name	id	project_id
1	AK_Rogers_C001	AK_Rogers	ps.3568	BR_AK1.ps.3
2	AK_Rogers_C001	AK_Rogers	ps.3569	BR_AK1.ps.4
3	AK_Rogers_C001	AK_Rogers	ps.3566	BR_AK1.ps.1
4	AK_Rogers_C001	AK_Rogers	ps.3567	BR_AK1.ps.2
5	AK_Rogers_C001	AK_Rogers	ps.3572	BR_AK1.ps.7
6	AK_Rogers_C001	AK_Rogers	ps.3573	BR_AK1.ps.8

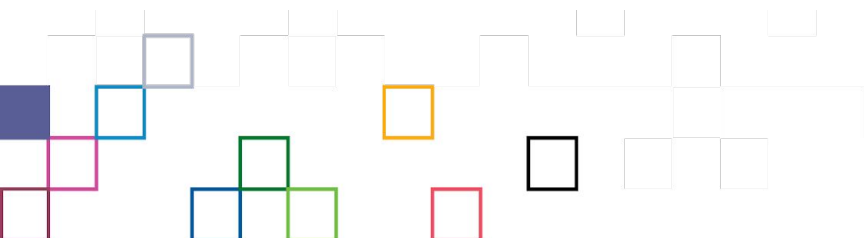
What can you do with this data? What is this data about?

```
python

# Re-importing pandas due to session reset
import pandas as pd

# Load the second dataset to inspect it and understand its structure
file_path_2 = '/mnt/data/AK_CA_Soil_Profile_Synthesis.csv'
data_soil_profile = pd.read_csv(file_path_2)

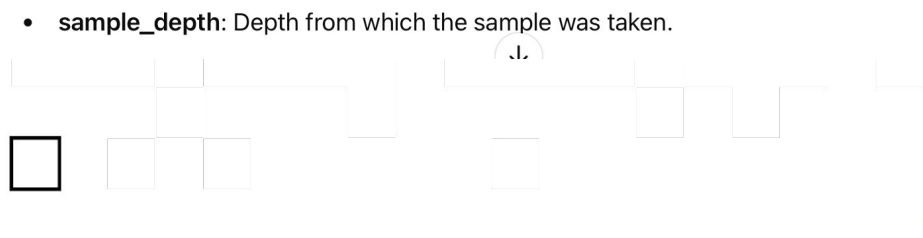
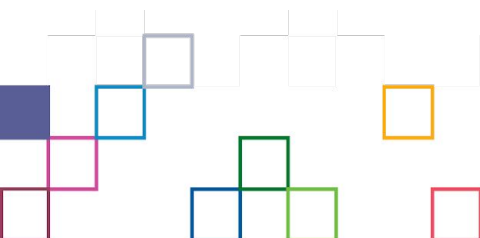
# Display the first few rows of the second dataset to understand its structure
data_soil_profile.head()
```



Open Source Data

The second dataset seems to focus on **soil profile synthesis** from different plots in Alaska and California. Here's a breakdown of the data:

- **site**: Location of the plot where the data was collected.
- **project_name**: Name of the project under which the data was collected.
- **id**: Identifier for the individual sample or measurement.
- **project_id**: Identifier for the project.
- **plot**: Identifier for the specific plot within the site.
- **latitude & longitude**: Geographic coordinates of the plot.
- **treatment**: Indicates whether the plot was part of a control or treatment group.
- **profile, min_org**: Refers to soil profile and organic layer categorization.
- **sample_depth**: Depth from which the sample was taken.



Open Source Data

Potential analyses with this data:

1. **Carbon and Nitrogen Content Analysis:** Understand the distribution of carbon and nitrogen across various plots and treatments.
2. **Soil Bulk Density Comparisons:** Explore how bulk density varies across different depths, treatments, and sites.
3. **Geographic Analysis:** Examine how soil properties vary based on location (latitude/longitude).
4. **pH Analysis:** Investigate the variation of soil pH in different plots and how it relates to carbon and nitrogen content.
5. **Impact of Fire on Soil Properties:** If linked with fire or other environmental data, explore how events like forest fires impact soil composition over time.

Open Source Data

python

```
# Cleaning the dataset by removing NaN values and
cleaned_data = data_soil_profile[(data_soil_profile['fine_carbon'] > 0) & (data

# Visualizing average fine carbon content by plot after removing NaN and invalid
plt.figure(figsize=(10, 6))
cleaned_data.groupby('plot')['fine_carbon'].mean().sort_values().plot(kind='bar')
plt.title('Average Fine Carbon Content by Plot (Cleaned)')
plt.xlabel('Plot')
plt.ylabel('Fine Carbon Content (%)')
plt.xticks(rotation=90, fontsize=6) # Keeping the labels small for readability
plt.tight_layout()
plt.show()

# Visualizing average fine nitrogen content by plot after cleaning
plt.figure(figsize=(10, 6))
cleaned_data.groupby('plot')['fine_nitrogen'].mean().sort_values().plot(kind='bar')
plt.title('Average Fine Nitrogen Content by Plot (Cleaned)')
plt.xlabel('Plot')
plt.ylabel('Fine Nitrogen Content (%)')
plt.xticks(rotation=90, fontsize=6)
plt.tight_layout()
plt.show()

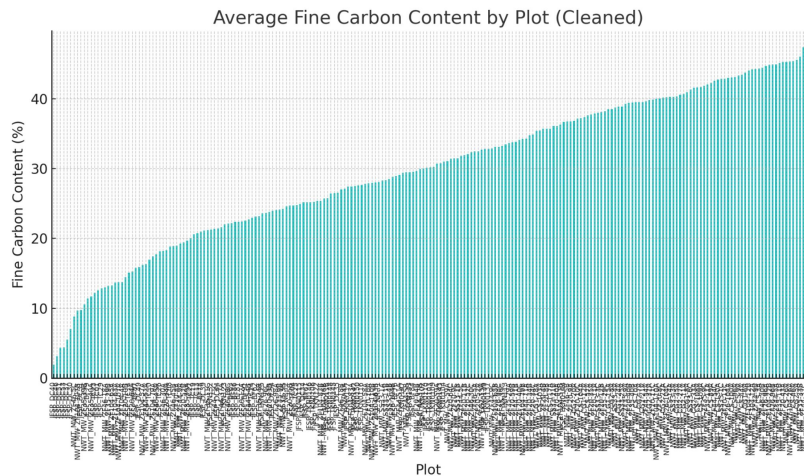
# Summary statistics after cleaning
carbon_cleaned_stats = cleaned_data['fine_carbon'].describe()
nitrogen_cleaned_stats = cleaned_data['fine_nitrogen'].describe()

carbon_cleaned_stats, nitrogen_cleaned_stats
```

Open Source Data

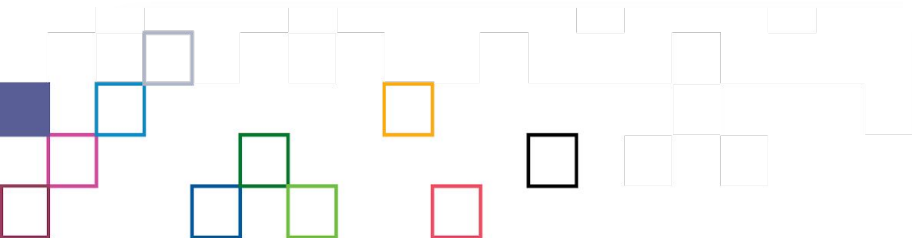
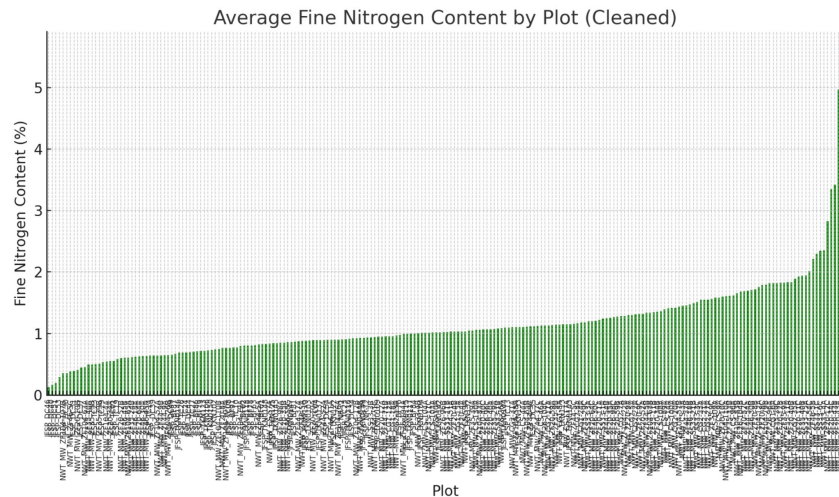
Average Fine Carbon Content By Plot (Cleaned)

🔍 📄 📊 📄 📄



Average Fine Nitrogen Content By Plot (Cleaned)

🔍 📄 📊 📄 📄



UMCS
CENTRUM SZTUCZNEJ INTELIGENCJI
I MODELOWANIA KOMPUTEROWEGO

Open Source Data

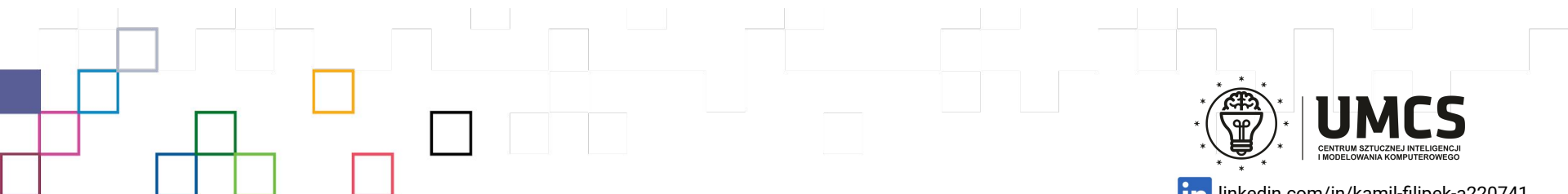
Charts:

1. Fine Carbon Content by Plot (Cleaned):

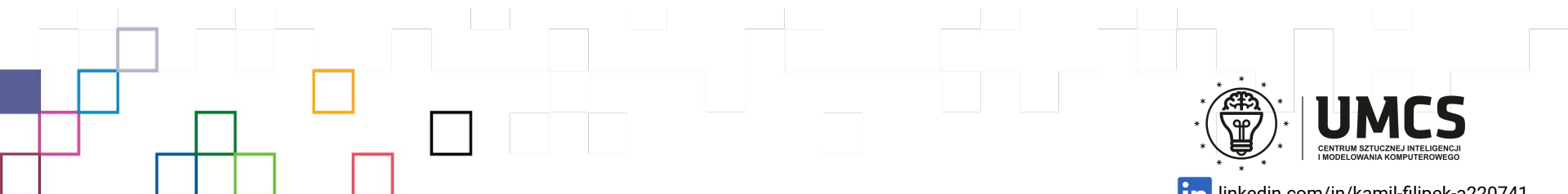
- The average carbon content is now visualized without the influence of negative or invalid values.
- The values range from about 0.093% to 66.38%, with most plots showing average carbon contents between 17% and 44%.

2. Fine Nitrogen Content by Plot (Cleaned):

- The nitrogen content ranges from 0.02% to 11.93%, with most plots having average nitrogen contents between 0.64% and 1.5%.

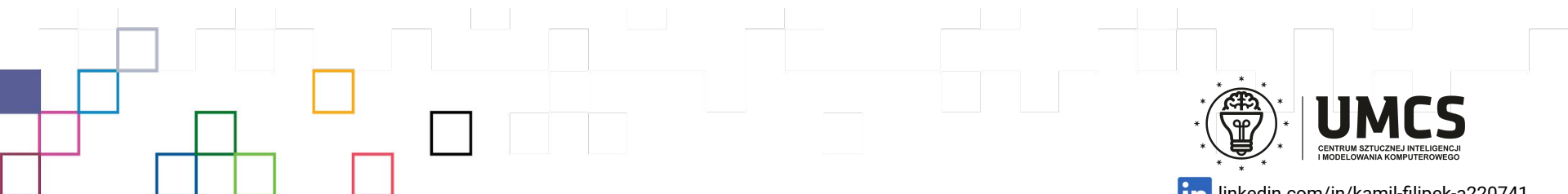


Another Example



Example

Imagine you are a climate scientist preparing a report on the impacts of rapid climate change on Earth's ecosystems. Your goal is to communicate the severity of these changes to both policymakers and the general public in a clear, scientific manner. For this, you will create a section of your report that focuses on a specific example of ecosystem disruption due to rising global temperatures.



Example

1. Define the Project Objectives and Requirements

- **Objective Clarification:**
 - Accurately identify and delineate glacier boundaries in satellite images.
 - Handle variability in image conditions (seasons, lighting, angles).
 - Differentiate between glacier ice and sea ice.
- **Performance Metrics:**
 - Accuracy, Precision, Recall, Intersection over Union (IoU).
 - Computational efficiency for processing large datasets.

2. Data Collection

- **Satellite Image Sources:**
 - **Landsat Program:** Provides multispectral images with different resolutions.
 - **Sentinel-2:** Offers high-resolution optical imagery.
 - **MODIS (Moderate Resolution Imaging Spectroradiometer):** Useful for large-scale monitoring.
- **Additional Data Sources:**
 - **Digital Elevation Models (DEMs):** To incorporate topographical information.
 - **Climate Data:** For contextual analysis (temperature, precipitation).
- **Data Acquisition Considerations:**
 - Obtain images from different seasons and times of day to capture variability.
 - Ensure coverage of diverse glacier regions to enhance model generalization.

3. Data Preprocessing

- **Image Correction:**
 - **Radiometric Correction:** Adjust for sensor noise and atmospheric distortions.
 - **Geometric Correction:** Align images to a common coordinate system.
- **Normalization:**
 - Scale pixel values to a standard range (e.g., 0-1) to facilitate model training.
- **Data Augmentation:**
 - Apply transformations such as rotation, flipping, scaling, and brightness adjustments to increase dataset diversity and robustness.
- **Handling Different Angles:**
 - Use techniques like image stitching or perspective transformation to mitigate angle-induced distortions.

4. Data Annotation

- **Creating Ground Truth Labels:**
 - **Manual Annotation:** Use GIS software (e.g., QGIS, ArcGIS) or specialized annotation tools (e.g., Labelbox, CVAT) to manually delineate glacier borders.
 - **Semi-Automated Annotation:** Utilize existing GIS layers or preliminary models to assist in labeling.
- **Annotation Guidelines:**
 - Establish consistent criteria for what constitutes glacier boundaries to ensure label uniformity.
 - Include annotations for both glacier ice and sea ice where applicable.
- **Quality Assurance:**
 - Perform inter-annotator agreement checks to maintain annotation quality.
 - Review and refine annotations iteratively.

Example

5. Data Splitting

- **Training, Validation, and Testing Sets:**
 - Split the dataset (e.g., 70% training, 15% validation, 15% testing) ensuring diverse representation in each set.
 - Maintain spatial and temporal diversity to prevent data leakage and enhance model generalization.

6. Model Selection and Architecture Design

- **Technology and Programming Language:**
 - **Programming Language:** Python is recommended due to its extensive libraries and community support.
 - **Deep Learning Frameworks:** TensorFlow or PyTorch for model development.
 - **Geospatial Libraries:** GDAL, Rasterio for handling geospatial data; OpenCV for image processing.
- **Model Architecture:**
 - **Convolutional Neural Networks (CNNs):** Suitable for image feature extraction.
 - **U-Net Architecture:** Particularly effective for semantic segmentation tasks like delineating boundaries.
 - **Transfer Learning:** Utilize pre-trained models (e.g., ResNet, VGG) as backbone networks to leverage existing feature representations.
- **Advanced Techniques:**
 - **Attention Mechanisms:** To focus on relevant parts of the image.
 - **Multi-Scale Feature Extraction:** To capture both fine and coarse boundary details.

7. Model Training

- **Environment Setup:**
 - Use GPU-enabled hardware or cloud-based platforms (e.g., AWS, Google Cloud, Azure) to accelerate training.
- **Training Process:**
 - **Loss Function:** Use appropriate loss functions such as Binary Cross-Entropy, Dice Loss, or Intersection over Union (IoU) Loss for segmentation tasks.
 - **Optimizer:** Adam or SGD optimizers with suitable learning rates.
 - **Batch Size and Epochs:** Determine based on dataset size and computational resources.
- **Hyperparameter Tuning:**
 - Experiment with different learning rates, batch sizes, and network depths.
 - Utilize techniques like Grid Search or Bayesian Optimization for optimal parameter selection.

8. Model Evaluation

- **Performance Metrics:**
 - **Accuracy:** Overall correctness of the model.
 - **Precision and Recall:** Especially important for imbalanced classes.
 - **Intersection over Union (IoU):** Measures overlap between predicted and true boundaries.
 - **F1 Score:** Harmonic mean of precision and recall.
- **Validation Techniques:**
 - **Cross-Validation:** To assess model performance across different data subsets.
 - **Confusion Matrix:** To visualize true vs. predicted classifications.
- **Error Analysis:**
 - Identify common misclassification patterns.
 - Analyze cases with poor performance to understand underlying issues (e.g., specific seasons, angles).



Example

9. Model Optimization and Refinement

- **Addressing Overfitting/Underfitting:**
 - Implement regularization techniques such as Dropout, L2 regularization.
 - Adjust model complexity based on performance trends.
- **Improving Generalization:**
 - Incorporate more diverse data.
 - Enhance data augmentation strategies.
- **Ensemble Methods:**
 - Combine multiple models to improve robustness and accuracy.

10. Deployment and Integration

- **Model Deployment:**
 - **APIs:** Use frameworks like Flask or FastAPI to create RESTful APIs for model inference.
 - **Containerization:** Deploy using Docker for scalability and portability.
- **Integration with GIS Systems:**
 - Integrate model outputs with GIS platforms for visualization and further analysis.
- **User Interface:**
 - Develop dashboards or visualization tools (e.g., using Dash, Streamlit) for end-users to interact with model predictions.

11. Monitoring and Maintenance

- **Performance Monitoring:**
 - Continuously track model performance using real-time data.
 - Implement alerts for performance degradation.
- **Periodic Retraining:**
 - Update the model with new data to maintain accuracy over time.
 - Adapt to changes in glacier dynamics and satellite imaging technologies.
- **Feedback Loop:**
 - Incorporate user feedback to identify areas for improvement.
 - Refine annotations and retrain models accordingly.

12. Documentation and Reporting

- **Comprehensive Documentation:**
 - Detail data sources, preprocessing steps, model architectures, training procedures, and evaluation metrics.
 - Include instructions for deployment and usage.
- **Reporting:**
 - Prepare reports highlighting model performance, limitations, and potential areas for future research.

13. Ethical and Environmental Considerations

- **Data Privacy and Compliance:**
 - Ensure compliance with data usage policies of satellite image providers.
- **Environmental Impact:**
 - Optimize model efficiency to minimize computational resource consumption.

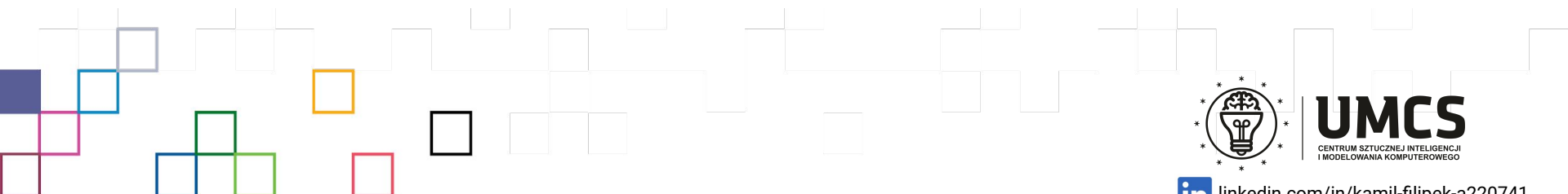
Example

Proposed Technology Stack

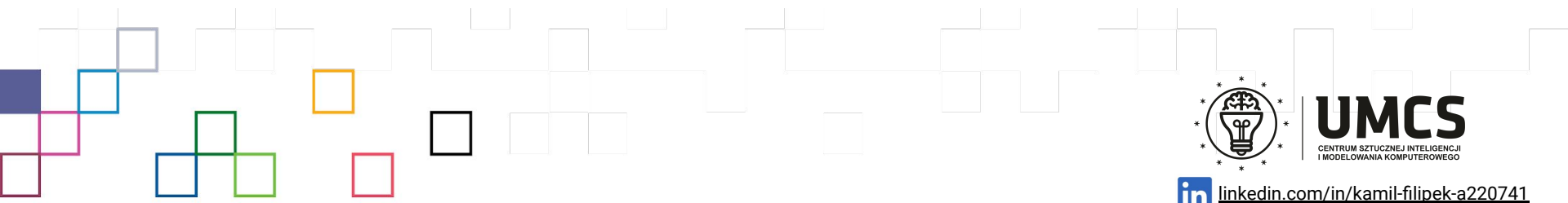
- **Programming Language:** Python
- **Deep Learning Frameworks:** TensorFlow or PyTorch
- **Geospatial Libraries:** GDAL, Rasterio
- **Image Processing Libraries:** OpenCV, scikit-image
- **Data Annotation Tools:** Labelbox, CVAT, QGIS
- **Development Environment:** Jupyter Notebooks, VS Code
- **Deployment Tools:** Docker, Flask/FastAPI, Kubernetes (for scaling)
- **Visualization Tools:** Dash, Streamlit, GIS Platforms (e.g., QGIS)

Potential Novel Adaptation Strategies

- **Few-Shot Learning:**
 - Implement models that can generalize from a limited number of annotated examples, reducing the annotation burden.
- **Self-Supervised Learning:**
 - Utilize unlabeled satellite images to pre-train models, enhancing feature extraction capabilities.
- **Domain Adaptation:**
 - Adapt models trained on specific regions or conditions to perform well across diverse geographic and environmental settings.
- **Active Learning:**
 - Develop systems where the model actively selects the most informative samples for annotation, improving training efficiency.
- **Integration of Temporal Data:**
 - Incorporate time-series satellite data to capture glacier dynamics, enhancing boundary delineation accuracy.



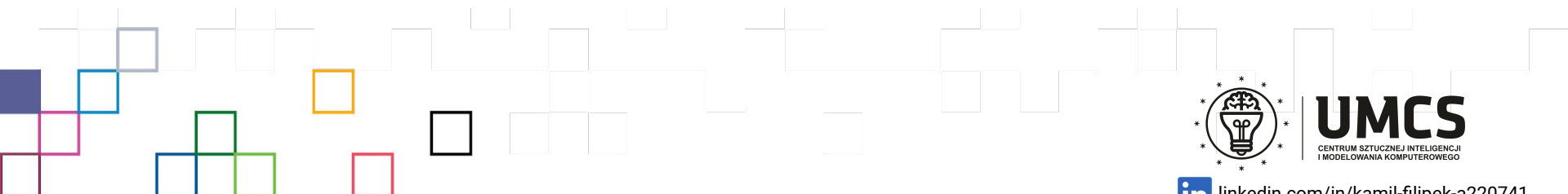
Claude



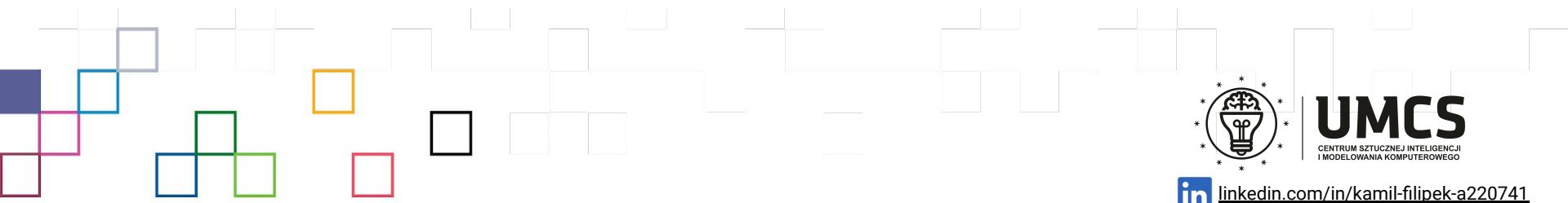
UMCS
CENTRUM SZTUCZNEJ INTELIGENCJI
I MODELOWANIA KOMPUTEROWEGO

Example

Training an AI model to accurately delineate glacier borders on satellite images involves a comprehensive workflow that addresses data variability, annotation precision, model selection, and deployment considerations. Below is a structured step-by-step guide to performing such an analysis, along with recommended technologies and programming languages.



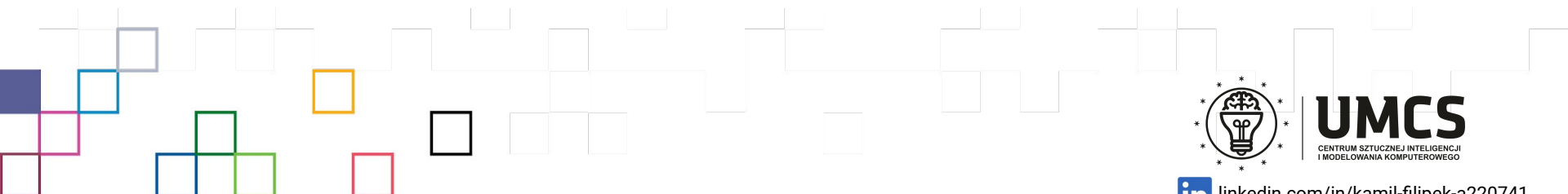
Code Assessment and Interpretation



UMCS
CENTRUM SZTUCZNEJ INTELIGENCJI
I MODELOWANIA KOMPUTEROWEGO

Code Assessment and Interpretation

- **My tips:**
 - Review each line of the code
 - Check partial results
 - Ask ChatGPT for alternative code
 - Compare results for two scripts

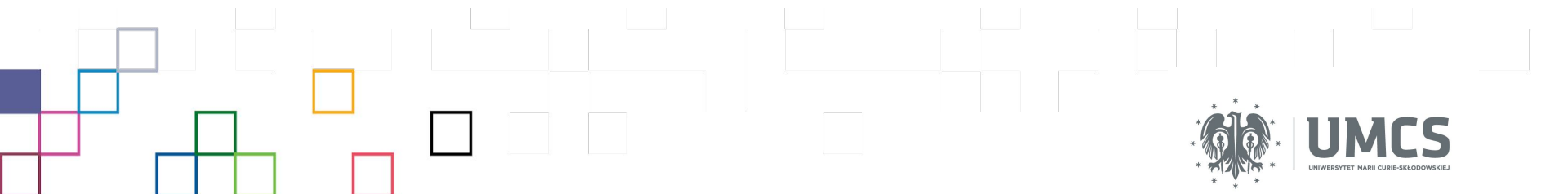


Thank you!

kfilipek@umcs.pl

@Kamil_Filipek

[linkedin.com/in/kamil-filipek-a220741](https://www.linkedin.com/in/kamil-filipek-a220741)



UMCS
UNIWERSYTET MARII CURIE-SKŁODOWSKIEJ