Institute and Faculty of Actuaries

Life Conference 2023

22–24 November, ICC Birmingham

Institute and Faculty of Actuaries

Generative AI: the biggest transformation since desktop computing

Matthew Edwards
Daniel Ramsay
Arlen Galicia Carreon

Generative AI: the biggest transformation?

1. What is “artificial intelligence”?
2. Insurance use cases
3. Safe adoption

Matthew Edwards
Head of Innovation
UK&I Life Consulting
Matthew.Edwards@wtwco.com

Daniel Ramsay
Actuarial Data Scientist
UK&I Life Consulting
daniel.ramsay@wtwco.com

Arlen Galicia Carreon
Associate Director
UK&I Life Consulting
arlen.galiciacarreon@wtwco.com

Institute and Faculty of Actuaries

What is AI?

Much of 2023's excitement relates to Generative AI

Artificial Intelligence (1950s)
Field of computer science that seeks to create intelligent machines that can replicate or exceed human intelligence

Machine Learning (1990s)
Enables machines to learn from existing data and find patterns in that data to make decisions or predictions

Deep Learning (2000s)
Layers of neural networks are used to process data and make decisions

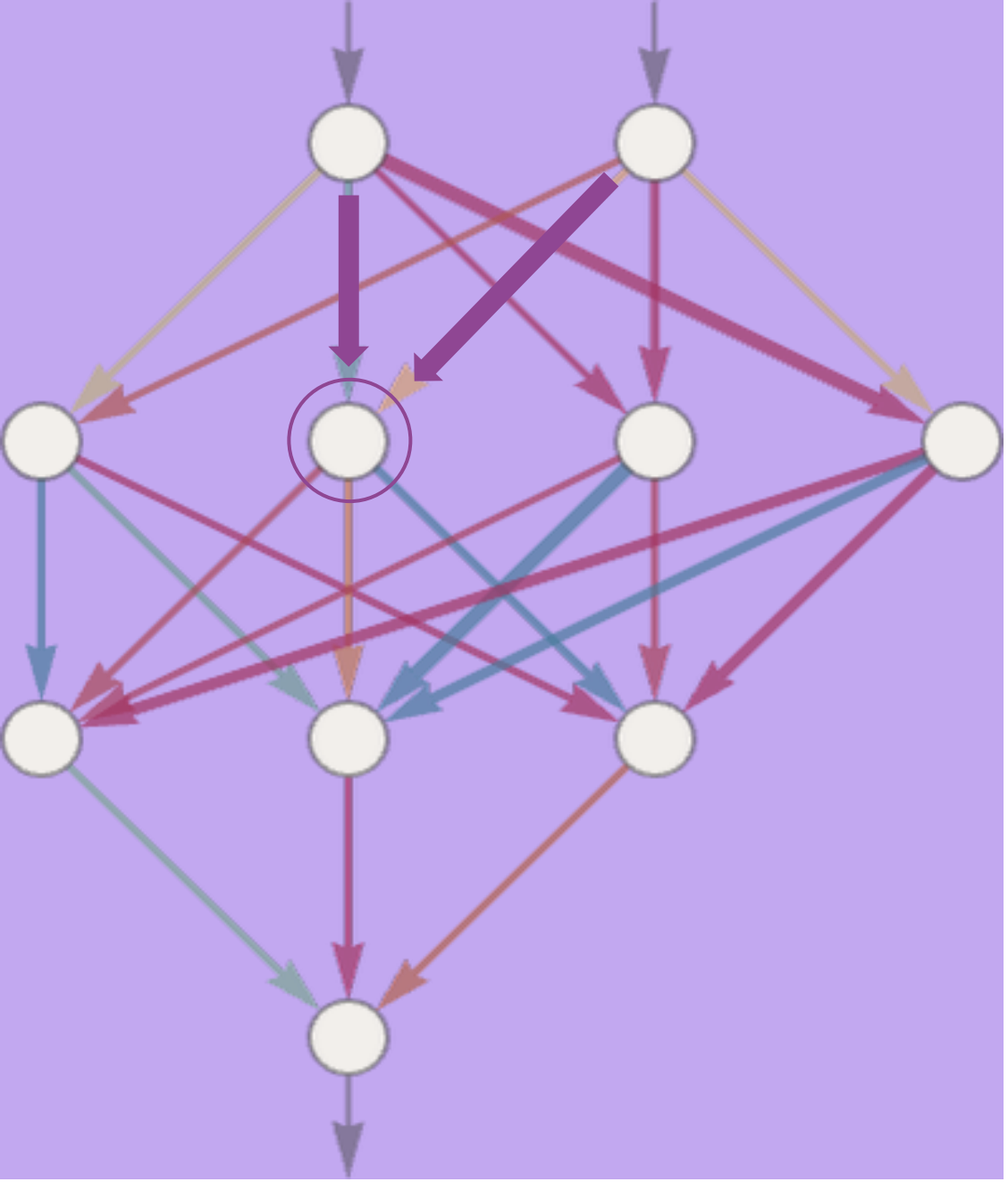
Generative AI (2020s)
Create new content from relatively short inputs ("prompts").

"Normal" AI examples

- Predictive Analytics
- Image Recognition

Generative AI examples

- Text and Code Creation
- Image Creation



Large language models (LLMs)

- “Large Language Models” are the engines which power text and code-based Generative AI
- LLMs are based on neural networks – 80 years old!
- These are connected layers of idealized neurons

- Each node calculates a simple numerical function based on input from the layer above, adjusted by a ‘weight’ parameter
- e.g. *output = sum.product (inputs, weights)*

- Parameters tuned to optimize output by seeing many correct examples
- Requires us to convert words (and their positions in text) into representative numbers (“tokens”)

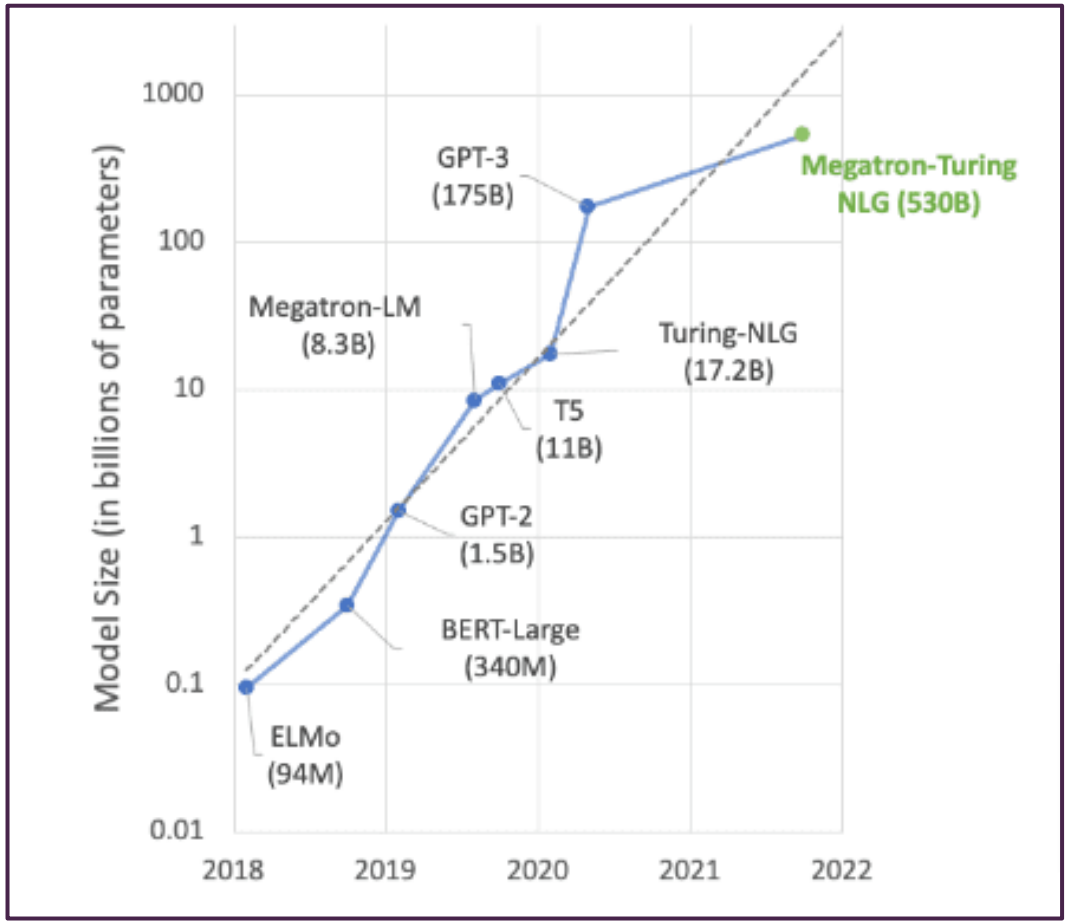
Institute and Faculty of Actuaries

Why are we suddenly talking about all this now?

- Transformer structure dates from 2017
- Model size has reached a ‘tipping point’ of quasi-human ability

- The models absorb information in accordance with their size
- The number of parameters in LLMs has increased x10,000 in last 4-5 years

- The parameters are mostly the weights in the LLM
  - GPT-3 has c. 175 billion parameters
  - GPT-4 is rumoured to have well over 1 trillion parameters
  - … which is 10x more than human brain neurons (although the metrics are not really comparable!)



Source: NVidia

Institute and Faculty of Actuaries

AI terminology overview

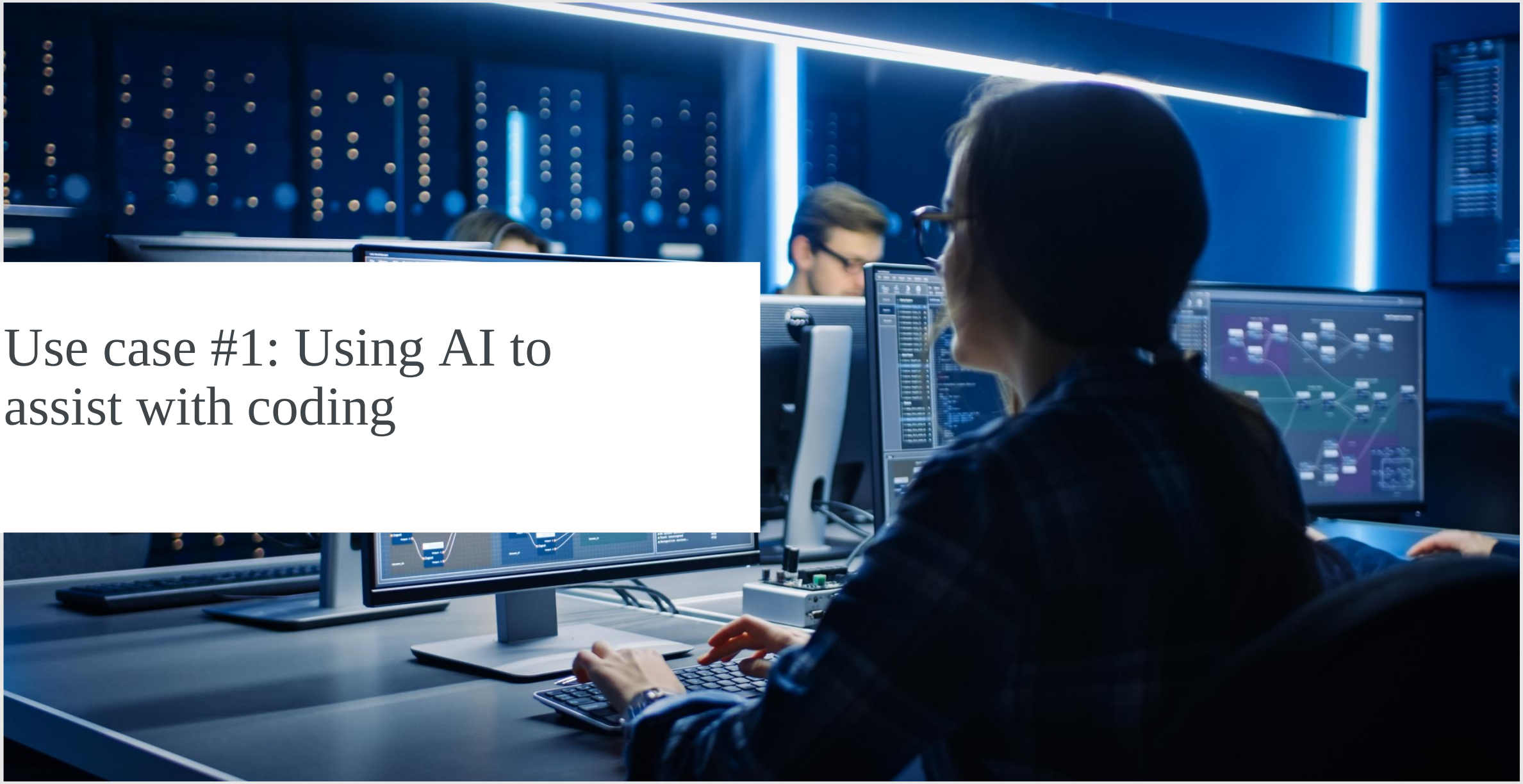
GPT	**G**enerative **P**retrained **T**ransformer**,** a type of AI model which produces high-quality text
Generative	Refers to the ability of a model to generate new data (e.g. text) that is coherent and meaningful – and may not be explicitly in the training data
Pretrained	Means the model has been trained on a large dataset (with GPT, from a massive corpus of text from the internet to learn language pattern etc.)
Transformer	Neural network architecture that's now standard in natural language AI. Relies on "self-attention", weighting importance of different parts of input data.
ChatGPT	A customer-facing website in a chatbot form that uses GPT-3.5 to respond to prompts

Characteristics of latest LLM Models

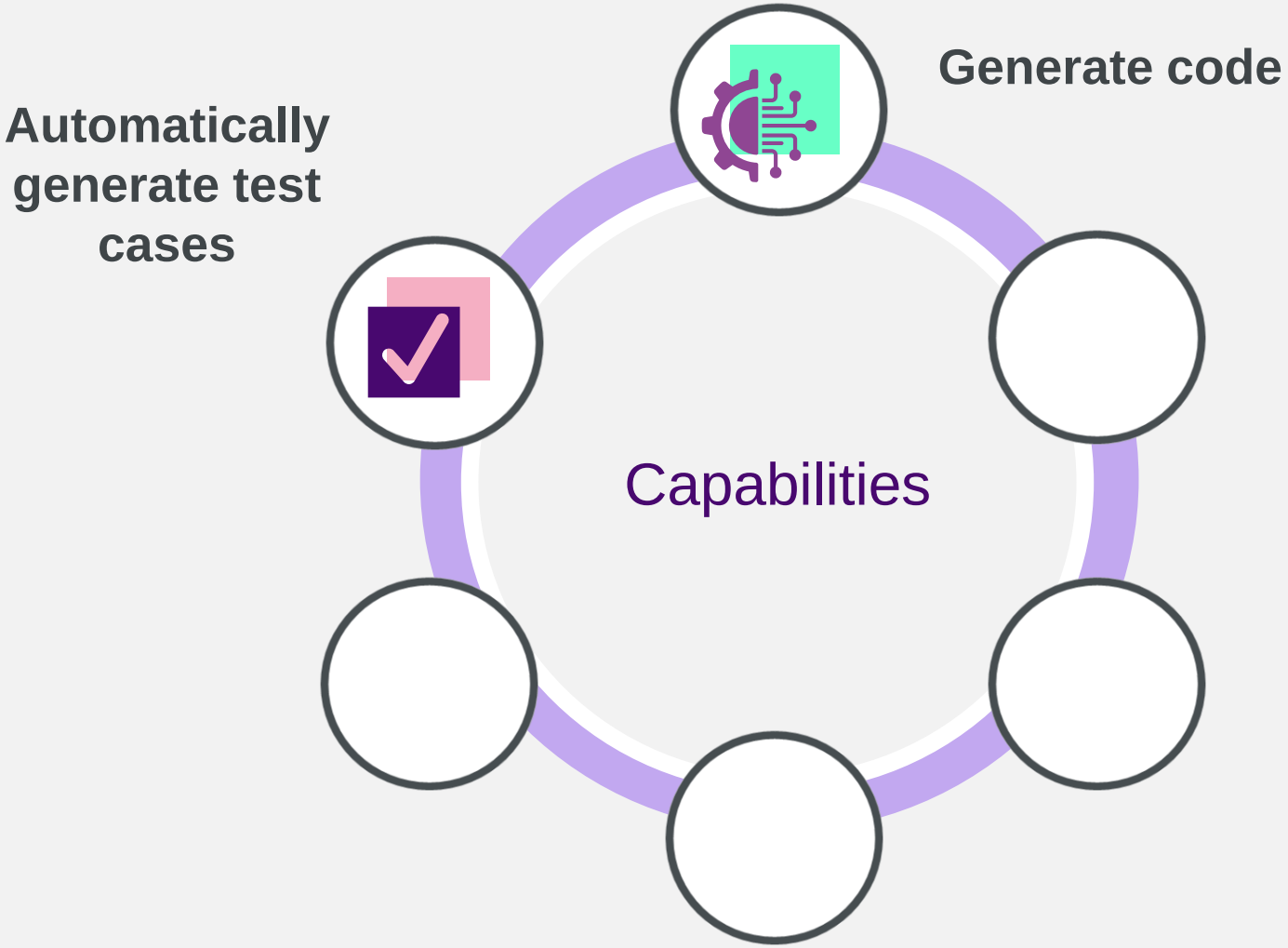
Company	OpenAI/ Microsoft	Anthropic	Google/ DeepMind	Facebook
Model	GPT4	Claude	Bard	Llama
Open-source	✗	✗	✗	✓
Power	●●●	●●	●	●

Institute and Faculty of Actuaries

Use case #1: Using AI to assist with coding



What AI coding assistants can do right now

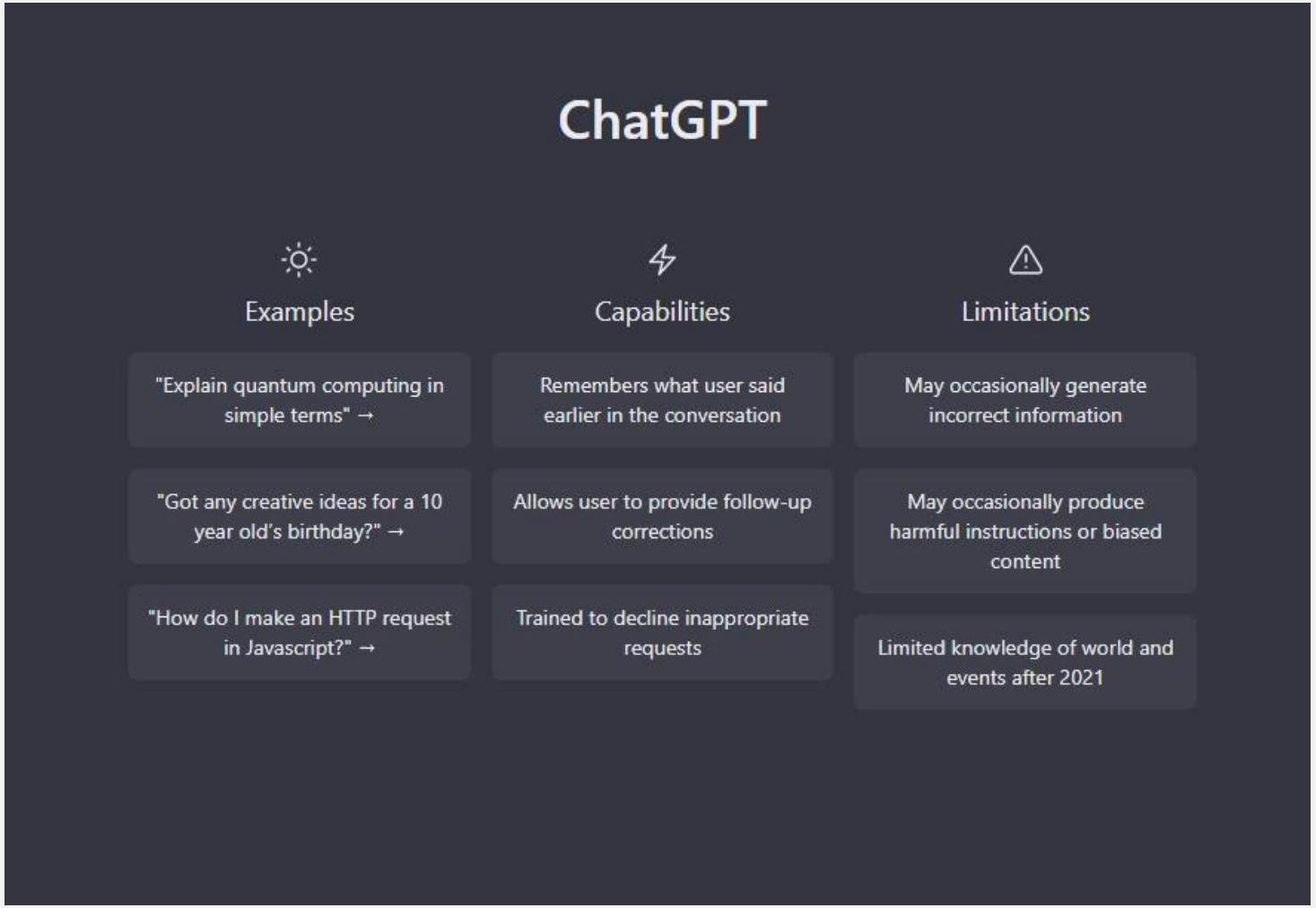


Generate code

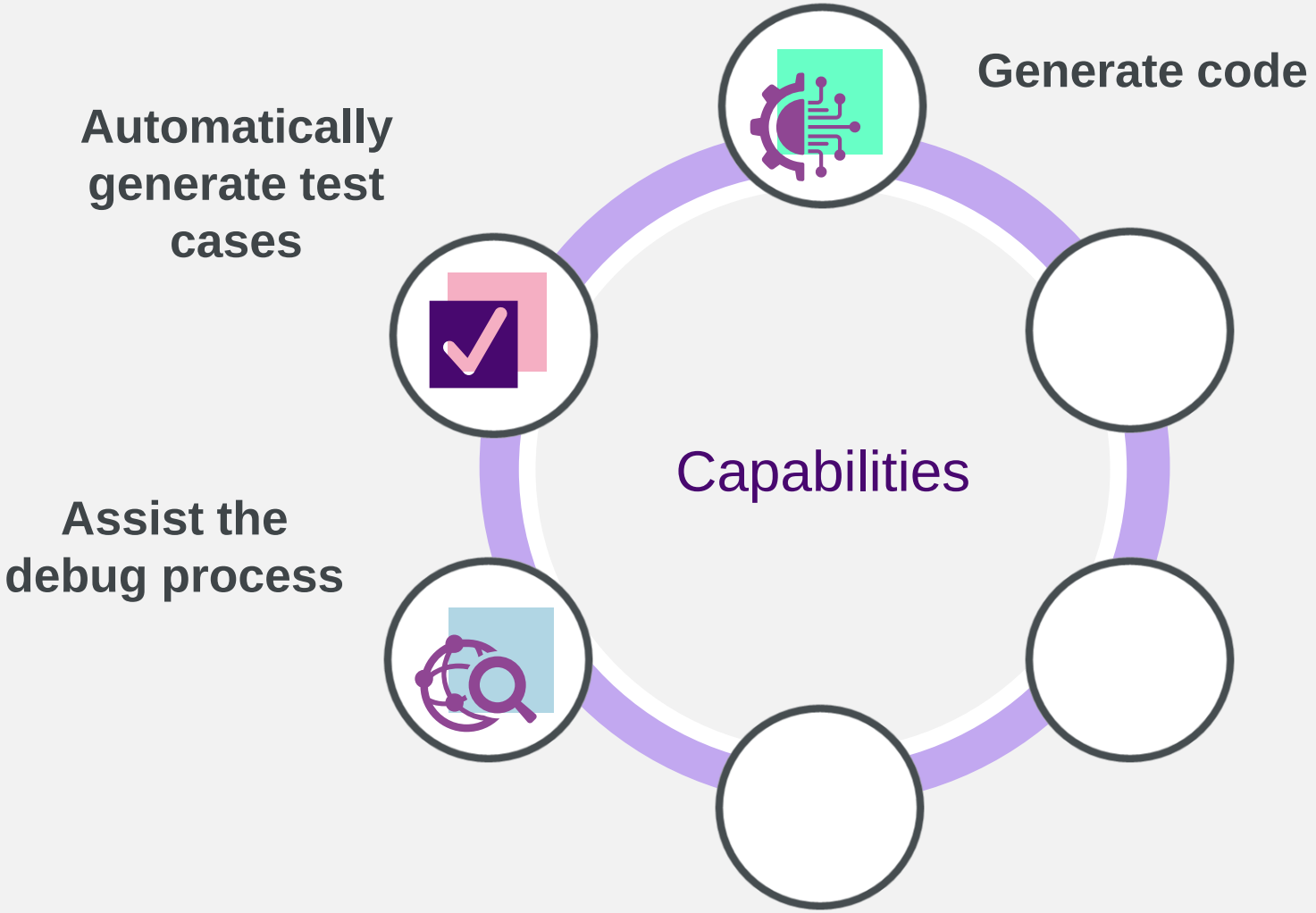
- Given a natural language prompt of user requirements, the bot will be able to generate the code that fulfils these requirements

Automatically generate test cases

- Given a function, the bot will auto-generate parameter data for test cases

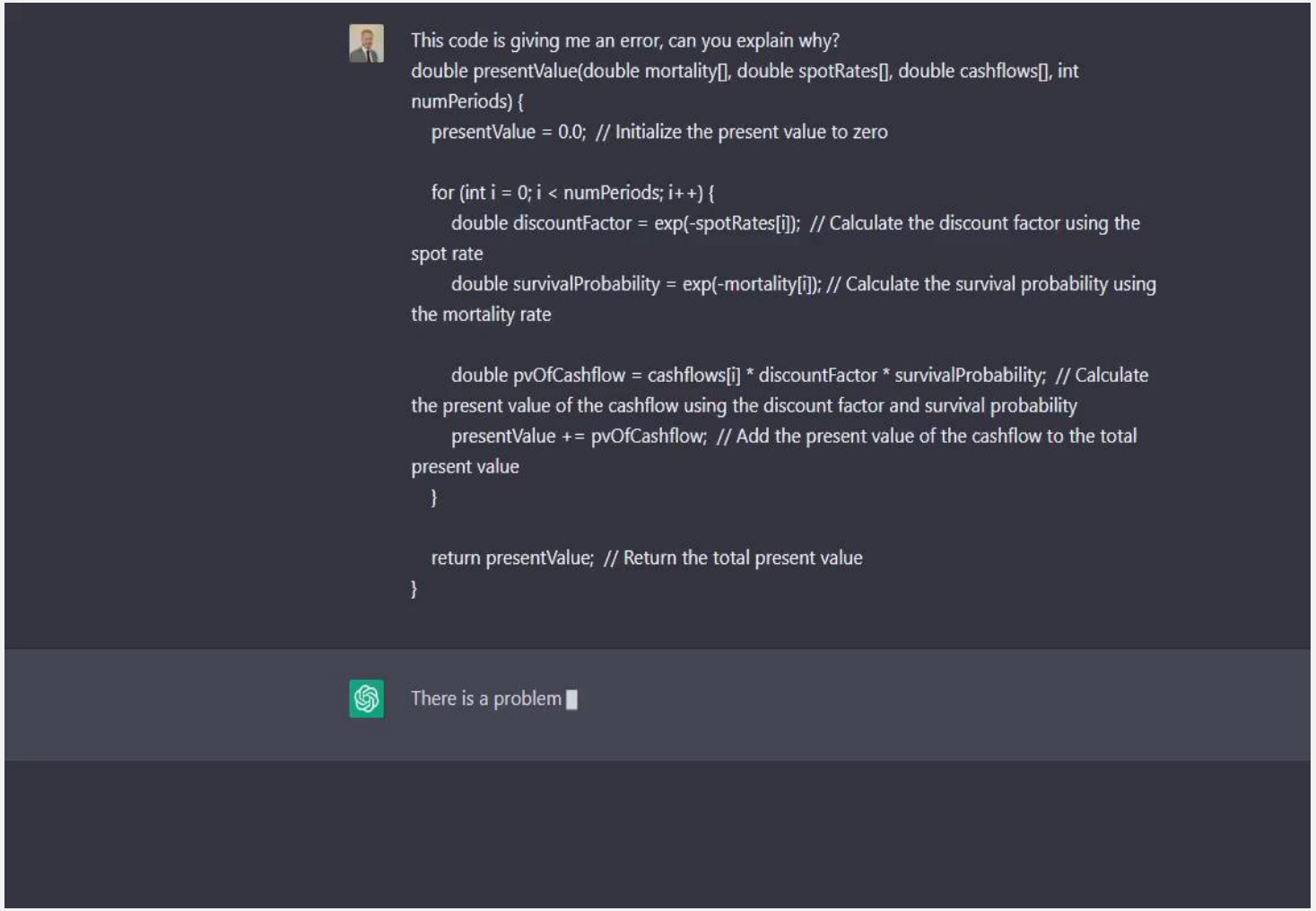


What AI coding assistants can do right now

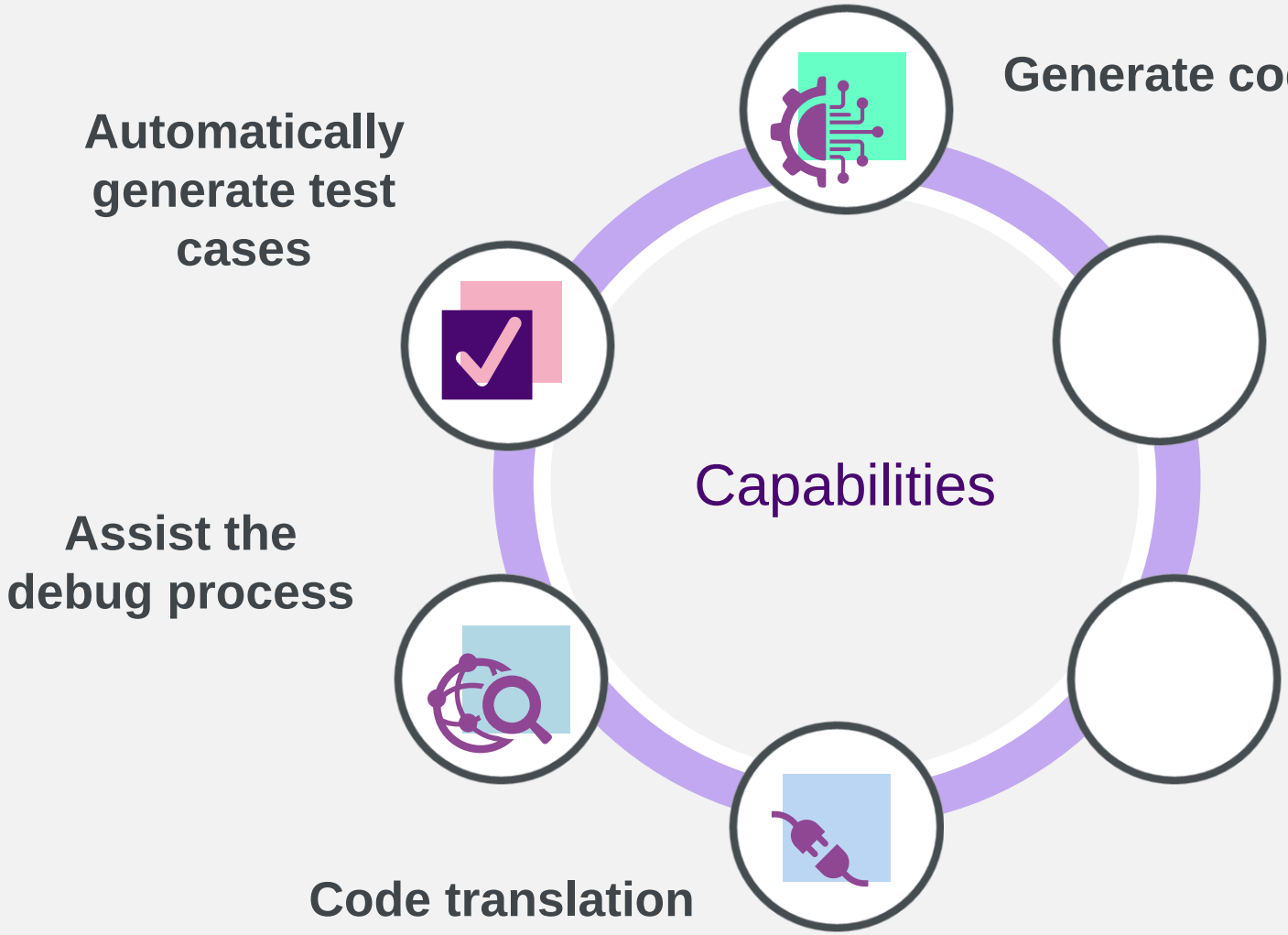


Assist the debug process

- Given code with errors, the bot will explain issues in the code and propose solutions

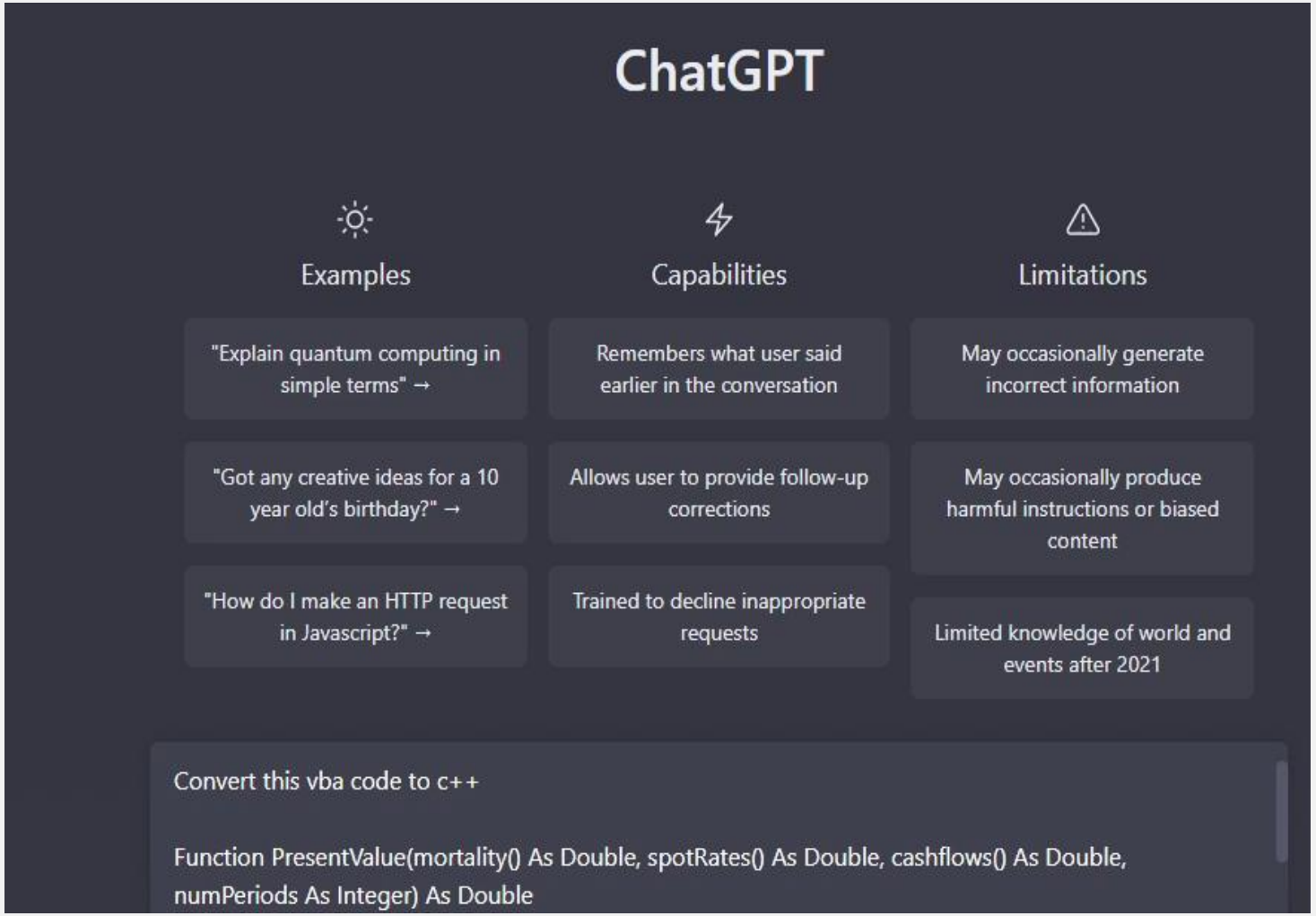


What AI coding assistants can do right now

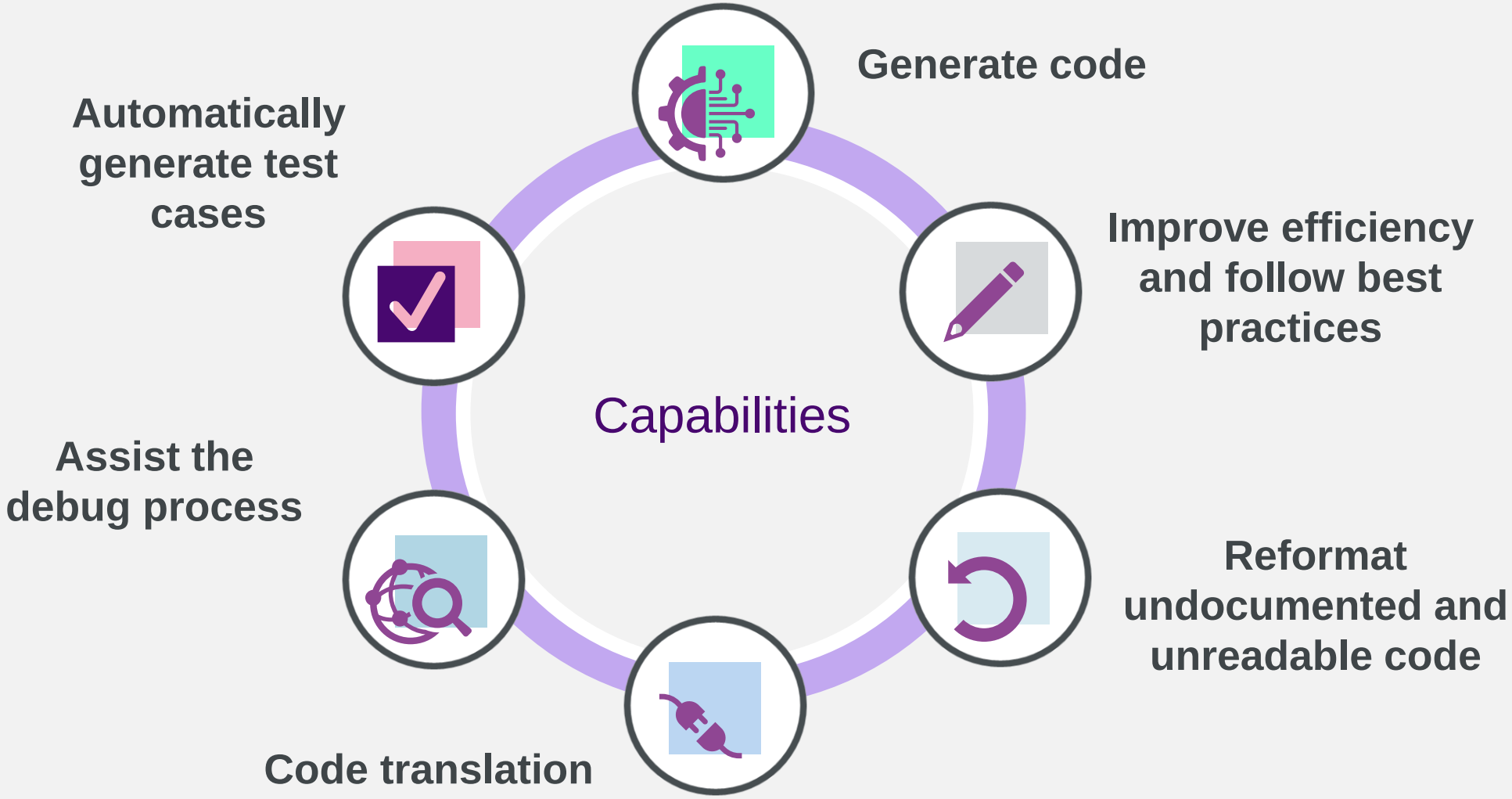


Code translation

- Given a pseudocode, VBA or other legacy platform code, the bot will translate the original code into RAFM compatible code



What AI coding assistants can do right now

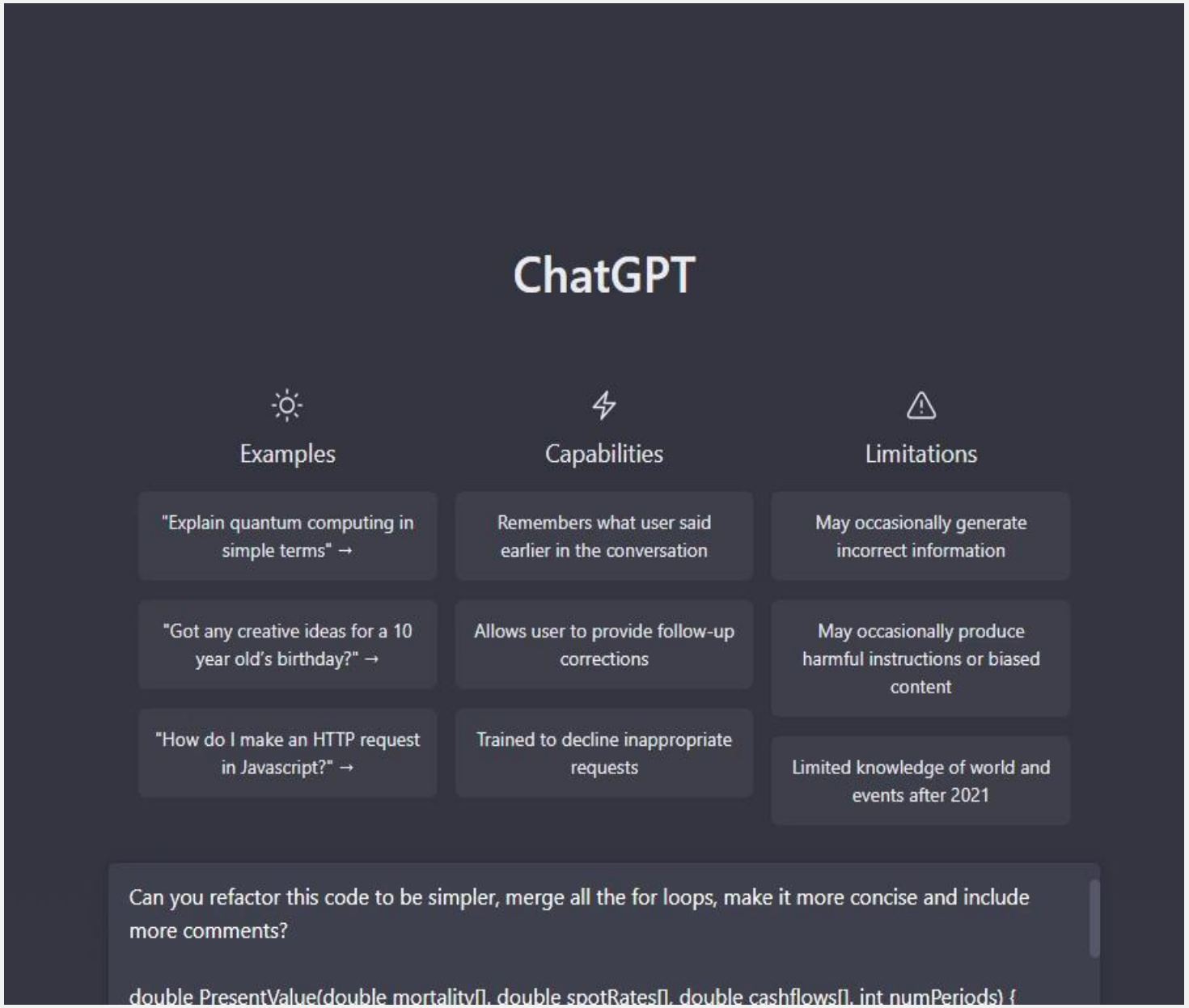


Rewrite code to improve efficiency and follow best practices

- Given an initial coding attempt by a less experienced developer, ActuatorGPT will give multiple suggestions to refactor code to make it more efficient or offer alternative approaches to coding problems

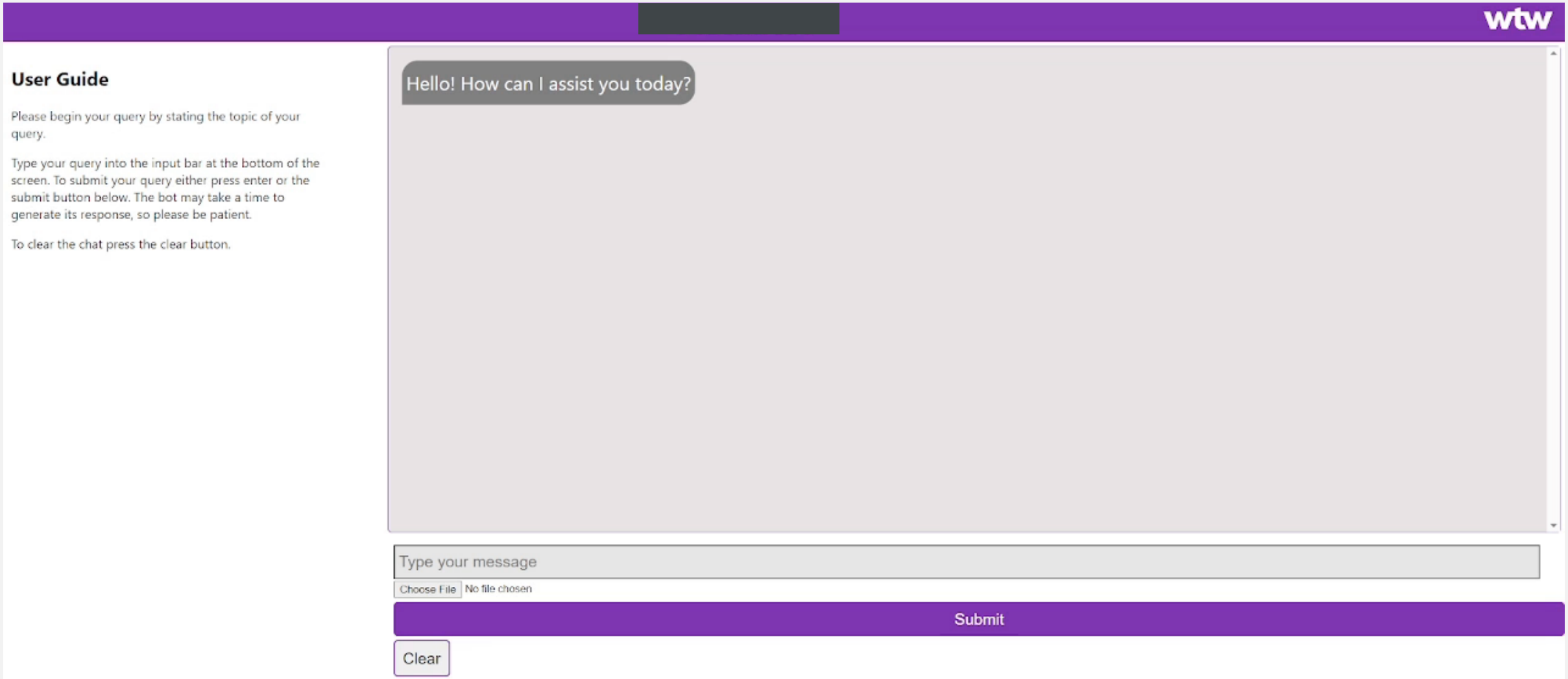
Reformat undocumented and unreadable legacy code

- Given legacy uncommented code, ActuatorGPT will reformat code with additional new comments explaining what the function is doing



Use case #2: AI for expert systems

A customised chatbot for insurance

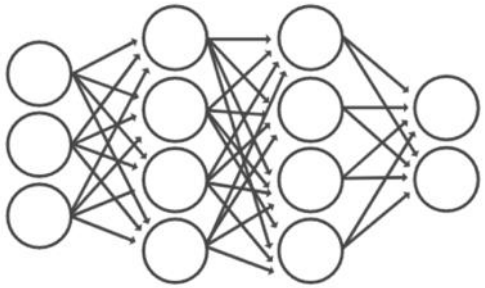


AI for expert systems

User request

Please explain…

LLM



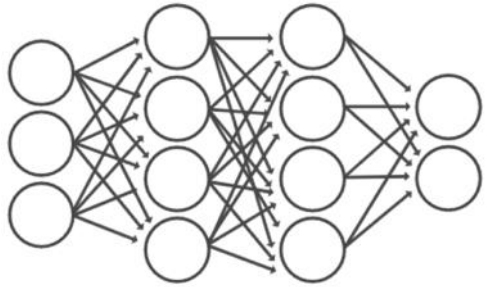
Embeddings knowledge database

IFRS17 rules
Solvency II rules
Other regulation
…

Collection of most relevant knowledge

Use this information…

LLM

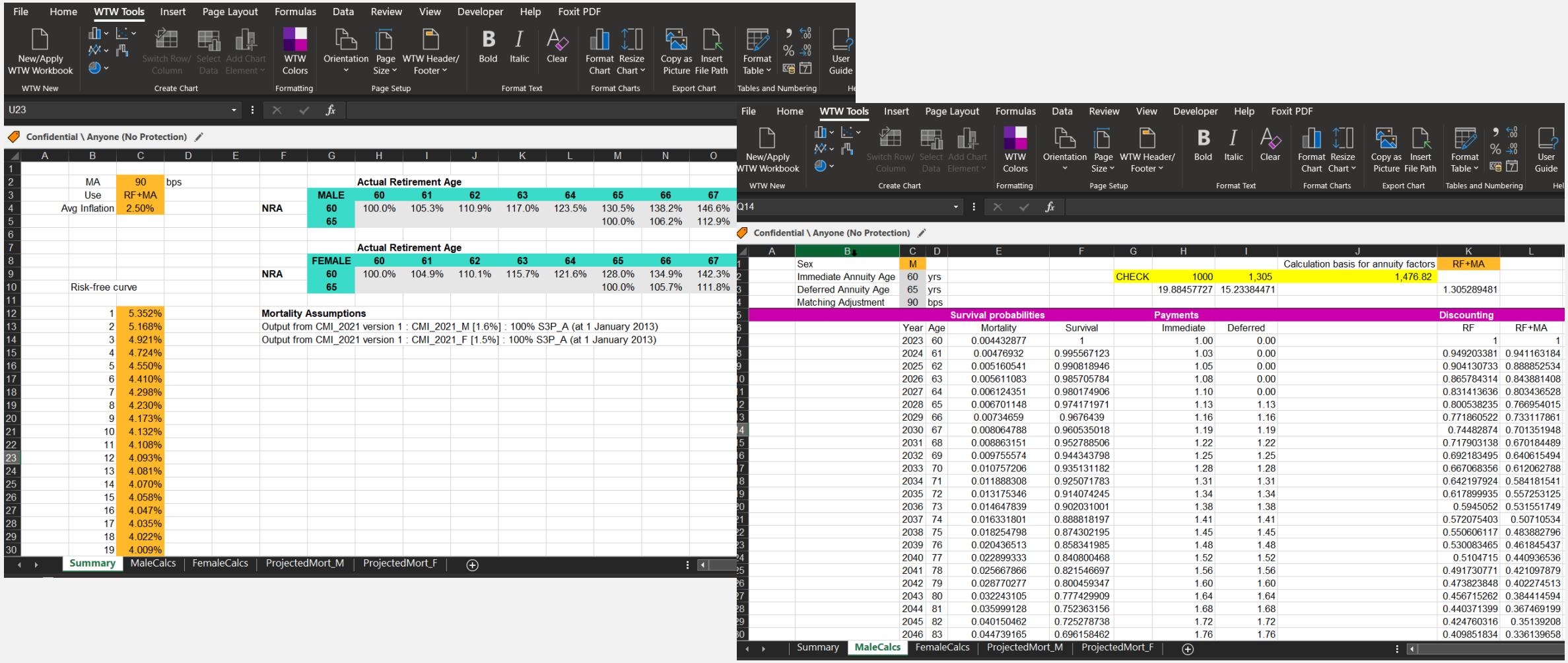


Text output

IFRS17 requires you do…
This
That
…

Use case #3: using AI for legacy model automation

Legacy models



File Home WTW Tools Insert Page Layout Formulas Data Review View Developer Help Foxit PDF

New/Apply WTW Workbook | Switch Row/ Column | Select Data | Add Chart Element | WTW Colors | Orientation | Page Size | WTW Header/ Footer | Bold | Italic | Clear | Format Chart | Resize Chart | Copy as Picture | Insert File Path | Format Table | User Guide

WTW New | Create Chart | Formatting | Page Setup | Format Text | Format Charts | Export Chart | Tables and Numbering | Help

U23

Confidential \ Anyone (No Protection)

	MA	90	bps
	Use	RF+MA	
	Avg Inflation	2.50%	

		1	5.352%
		2	5.168%
		3	4.921%
		4	4.724%
		5	4.550%
		6	4.410%
		7	4.298%
		8	4.230%
		9	4.173%
		10	4.132%
		11	4.108%
		12	4.093%
		13	4.081%
		14	4.070%
		15	4.058%
		16	4.047%
		17	4.035%
		18	4.022%
		19	4.009%

Risk-free curve

Actual Retirement Age

MALE	60	61	62	63	64	65	66	67
NRA 60	100.0%	105.3%	110.9%	117.0%	123.5%	130.5%	138.2%	146.6%
65						100.0%	106.2%	112.9%

Actual Retirement Age

FEMALE	60	61	62	63	64	65	66	67
NRA 60	100.0%	104.9%	110.1%	115.7%	121.6%	128.0%	134.9%	142.3%
65						100.0%	105.7%	111.8%

Mortality Assumptions

Output from CMI_2021 version 1 : CMI_2021_M [1.6%] : 100% S3P_A (at 1 January 2013)

Output from CMI_2021 version 1 : CMI_2021_F [1.5%] : 100% S3P_A (at 1 January 2013)

Summary | MaleCalcs | FemaleCalcs | ProjectedMort_M | ProjectedMort_F

File Home WTW Tools Insert Page Layout Formulas Data Review View Developer Help Foxit PDF

New/Apply WTW Workbook | Switch Row/ Column | Select Data | Add Chart Element | WTW Colors | Orientation | Page Size | WTW Header/ Footer | Bold | Italic | Clear | Format Chart | Resize Chart | Copy as Picture | Insert File Path | Format Table | User Guide

WTW New | Create Chart | Formatting | Page Setup | Format Text | Format Charts | Export Chart | Tables and Numbering | Help

Q14

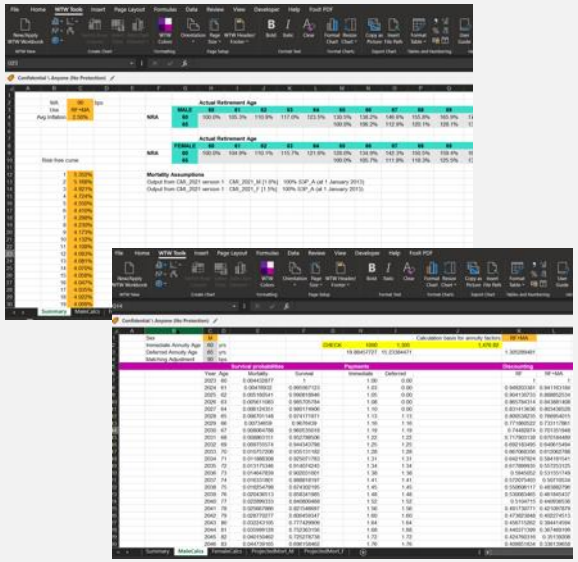
Confidential \ Anyone (No Protection)

	Sex	M					Calculation basis for annuity factors	RF+MA
	Immediate Annuity Age	60	yrs	CHECK	1000	1,305	1,476.82	
	Deferred Annuity Age	65	yrs		19.88457727	15.23384471		1.305289481
	Matching Adjustment	90	bps					

			Survival probabilities			Payments		Discounting	
		Year	Age	Mortality	Survival	Immediate	Deferred	RF	RF+MA
		2023	60	0.004432877	1	1.00	0.00	1	1
		2024	61	0.00476932	0.995567123	1.03	0.00	0.949203381	0.941163184
		2025	62	0.005160541	0.990818946	1.05	0.00	0.904130733	0.888852534
		2026	63	0.005611083	0.985705784	1.08	0.00	0.865784314	0.843881408
		2027	64	0.006124351	0.980174906	1.10	0.00	0.831413636	0.803436528
		2028	65	0.006701148	0.974171971	1.13	1.13	0.800538235	0.766954015
		2029	66	0.00734659	0.9676439	1.16	1.16	0.771860522	0.733117861
		2030	67	0.008064788	0.960535018	1.19	1.19	0.74482874	0.701351948
		2031	68	0.008863151	0.952788506	1.22	1.22	0.717903138	0.670184489
		2032	69	0.009755574	0.944343798	1.25	1.25	0.692183495	0.640615494
		2033	70	0.010757206	0.935131182	1.28	1.28	0.667068356	0.612062788
		2034	71	0.011888308	0.925071783	1.31	1.31	0.642197924	0.584181541
		2035	72	0.013175346	0.914074245	1.34	1.34	0.617899935	0.557253125
		2036	73	0.014647839	0.902031001	1.38	1.38	0.5945052	0.531551749
		2037	74	0.016331801	0.888818197	1.41	1.41	0.572075403	0.50710534
		2038	75	0.018254798	0.874302195	1.45	1.45	0.550606117	0.483882796
		2039	76	0.020436513	0.858341985	1.48	1.48	0.530083465	0.461845437
		2040	77	0.022899333	0.840800468	1.52	1.52	0.5104715	0.440936536
		2041	78	0.025667866	0.821546697	1.56	1.56	0.491730771	0.421097879
		2042	79	0.028770277	0.800459347	1.60	1.60	0.473823848	0.402274513
		2043	80	0.032243105	0.777429909	1.64	1.64	0.456715262	0.384414594
		2044	81	0.035999128	0.752363156	1.68	1.68	0.440371399	0.367469199
		2045	82	0.040150462	0.725278738	1.72	1.72	0.424760316	0.35139208
		2046	83	0.044739165	0.696158462	1.76	1.76	0.409851834	0.336139658

Summary | MaleCalcs | FemaleCalcs | ProjectedMort_M | ProjectedMort_F

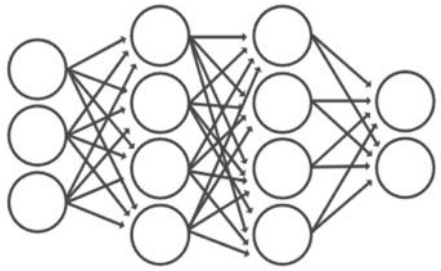
Using generative AI to assist transformation



Model compression

Prompt engineering

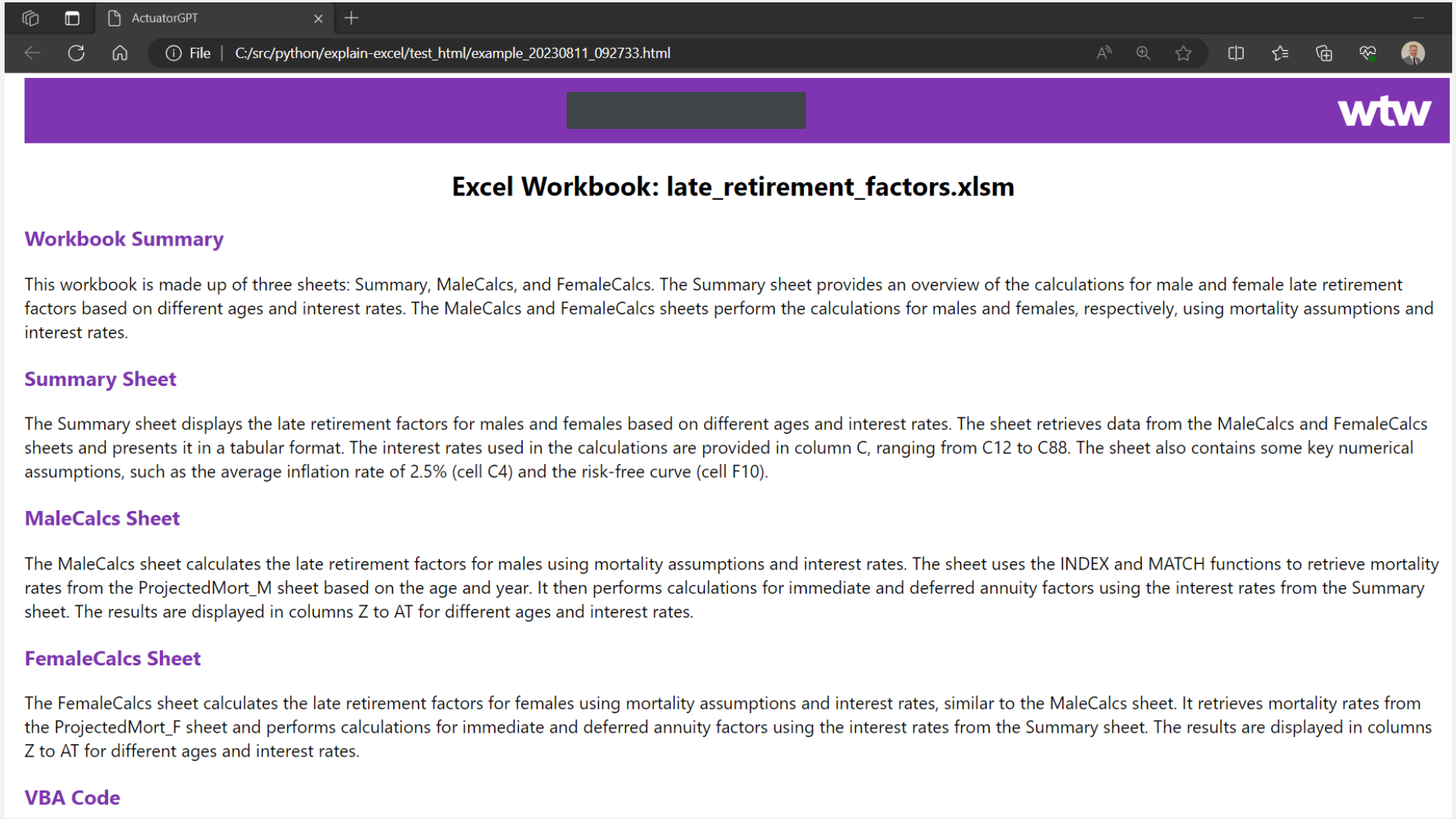
LLM



PDF

python

Generation of an audit trail or summary report



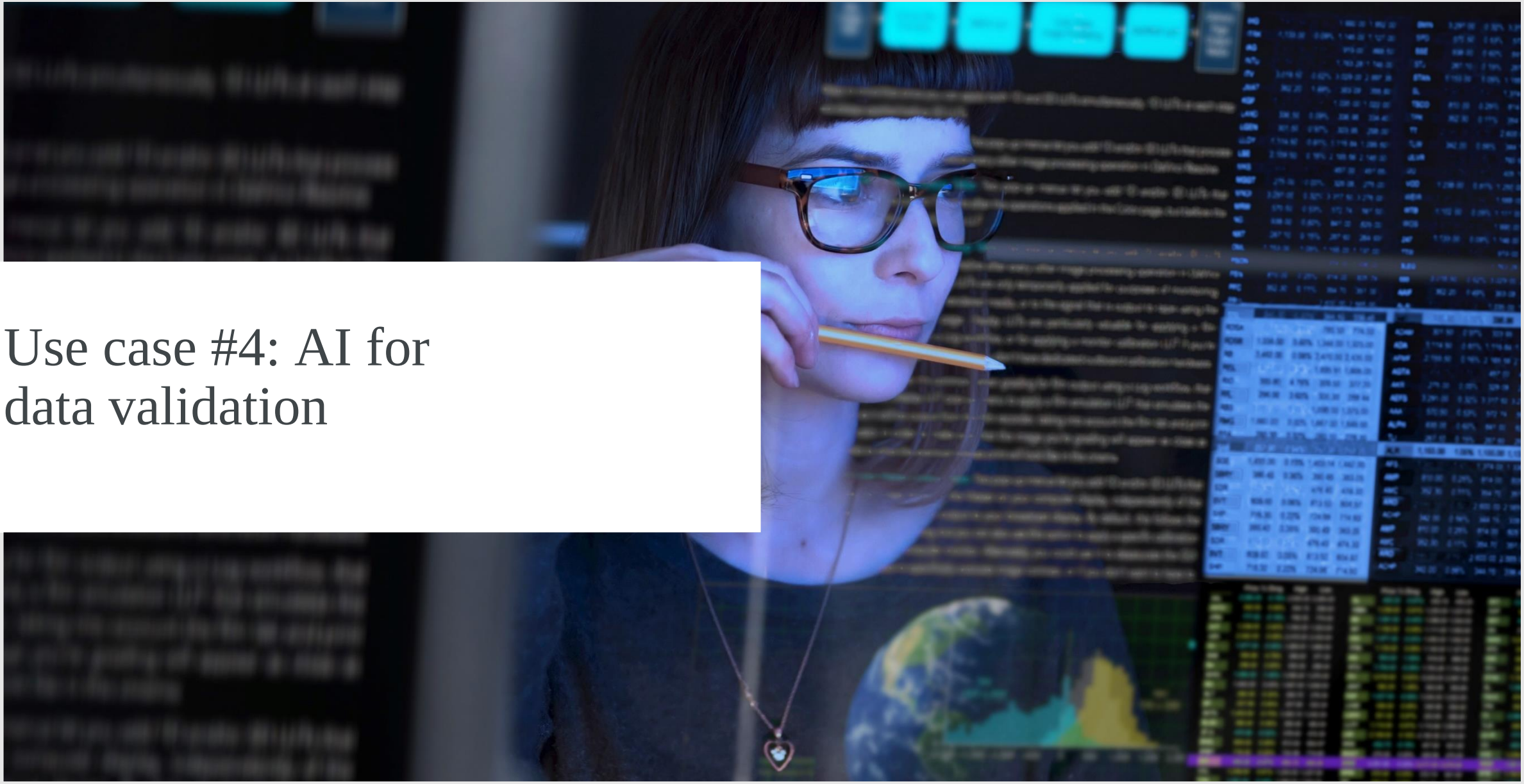
Generation of python code

explain_excel.ipynb | example_1686065389.py 1 | example_1686127655.py

test_py > example_1686127655.py > ...

```python
import pandas as pd

def annuity_factors(
        sex, age, immediate_annuity_age, deferred_annuity_age, ir_curve,ma_adjustment, mort_m_filepath,
        mort_f_filepath, mort_m_sheetname, mort_f_sheetname
        ):
    mort_m = pd.read_excel(mort_m_filepath, mort_m_sheetname, header=10, index_col=1)
    mort_f = pd.read_excel(mort_f_filepath, mort_f_sheetname, header=10, index_col=1)

    mort = mort_m if sex == "M" else mort_f

    survival_prob = mort.loc[age].values[0]

    immediate_payment = 1 if age >= immediate_annuity_age else 0
    deferred_payment = 1 if age >= deferred_annuity_age else 0

    discount_rf = (1 + ir_curve) ** (-age)
    discount_rf_ma = (1 + ir_curve + ma_adjustment / 10000) ** (-age)

    annuity_factor_rf = survival_prob * immediate_payment * discount_rf
    annuity_factor_rf_ma = survival_prob * deferred_payment * discount_rf_ma

    return annuity_factor_rf, annuity_factor_rf_ma
```

Use case #4: AI for data validation



Identifying errors, formatting issues, missing values

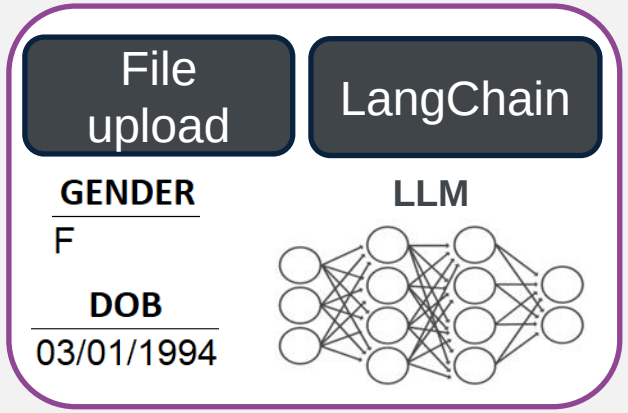
	A	B	C	D	E	F
1	DOB	GENDER	POSTCODE	PRODUCT	MARITAL STATUS	PREMIUM
2	03/01/1994	F	CM12 0HD	PROD3	Single	3867.2
3	13/01/1948	Male	OX9 2BN	PROD2	Yes	3634.07
4	15/01/2002	Female	E15 3LN	1	Single	3847.35
5	12/03/2017	M	BH3 7NE	PROD1	Married	3926.7
6	03/11/2020	F	CR44 1AP	PROD2	maried	3781.02
7	30/08/1951	M	3311 ER	PROD3	out of wed	MALE
8	04/01/1998	F	PL4 0AL	product 1	S	3919.83
9	06/04/1996	F	TR24 0QB	PROD3	Married	3639.61
10	14/11/1936	U	DH1 4JU	PROD 3	Married	3831
11	Alan	M	GL4 6DG	2	Engaged	4058.26
12	11/02/1990	M	LD2 3ND	PROD2	Married	n/a
13	21/04/1974	F		PROD2	no	4275.75
14	31/12/1974	Mr.	LL47 6TW	PROD3	Married	4152.53
15	19/01/1929	F	ME12 3TB	PROD3	not married	"3988.99"
16	14/12/1958	M	EH6 4RZ	PROD2	Married	4210.6
17	21/07/1996	male	OX29 4JX	product 2	civil partnership	£4,298.42
18	22/05/1986	M	LA2 0PB	PROD2	Single	3879.98
19	02/03/1933	F	24 Barry Lane	PROD3	Married	3587.15
20	23/04/1996	FEMALE	EX31 1JF	PROD3	M	n/a

Data cleansing process

Raw excel input

	A	B	C	D	E	F
1	DOB	GENDER	POSTCODE	PRODUCT	MARITAL STATUS	PREMIUM
2	03/01/1994	F	CM12 0HD	PROD3	Single	3867.2
3	13/01/1948	Male	OX9 2BN	PROD2	Yes	3634.07
4	15/01/2002	Female	E15 3LN	1	Single	3847.35
5	12/03/2017	M	BH3 7NE	PROD1	Married	3926.7
6	03/11/2020	F	CR44 1AP	PROD2	maried	3781.02
7	30/08/1951	M	3311 ER	PROD3	out of wed	MALE
8	04/01/1998	F	PL4 0AL	product 1	S	3919.83
9	06/04/1996	F	TR24 0QB	PROD3	Married	3639.61
10	14/11/1936	U	DH1 4JU	PROD 3	Married	3831
11	Alan	M	GL4 6DG	2	Engaged	4058.26
12	11/02/1990	M	LD2 3ND	PROD2	Married	n/a
13	21/04/1974	F		PROD2	no	4275.75
14	31/12/1974	Mr.	LL47 6TW	PROD3	Married	4152.53
15	19/01/1929	F	ME12 3TB	PROD3	not married	"3988.99"
16	14/12/1958	M	EH6 4RZ	PROD2	Married	4210.6
17	21/07/1996	male	OX29 4JX	product 2	civil partnership	£4,298.42
18	22/05/1986	M	LA2 0PB	PROD2	Single	3879.98
19	02/03/1933	F	24 Barry Lane	PROD3	Married	3587.15
20	23/04/1996	FEMALE	EX31 1JF	PROD3	M	n/a

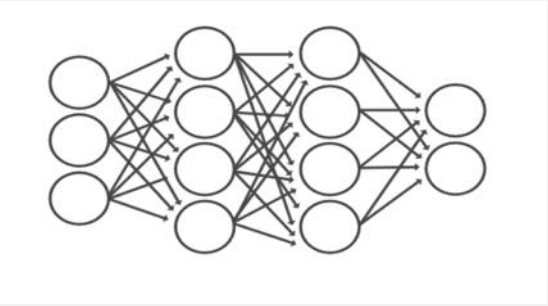
Generate cleaning rules for each column



Output python rules

```python
import pandas as pd
import numpy as np
import re
from datetime import datetime

def clean_GENDER(df, column):
    df[column+'_original'] = df[column]
    df[column+'_clean'] = df[column]
    df[column+'_comment'] = 'OK'

    male_formats = ['Male', 'male', 'M']
    female_formats = ['Female', 'female', 'F', 'feemale']
    unfixable_errors = ['U', 'Labour', 'Darren']

    for i in df.index:
        try:
            if df.loc[i, column] in male_formats:
                df.loc[i, column+'_clean'] = 'M'
                if df.loc[i, column] != 'M':
                    df.loc[i, column+'_comment'] = 'CLEANED'
            elif df.loc[i, column] in female_formats:
                df.loc[i, column+'_clean'] = 'F'
                if df.loc[i, column] != 'F':
                    df.loc[i, column+'_comment'] = 'CLEANED'
            elif df.loc[i, column] in unfixable_errors or pd.isnull(df.loc[i, column]):
                df.loc[i, column+'_clean'] = ''
                df.loc[i, column+'_comment'] = 'ERROR' if df.loc[i, column] in unfixable_errors else 'BLANK'
        except Exception as e:
            df.loc[i, column+'_clean'] = ''
            df.loc[i, column+'_comment'] = 'ERROR'
            print('Error processing row', i, ':', str(e))

    return df[[column+'_original', column+'_clean', column+'_comment']]
```

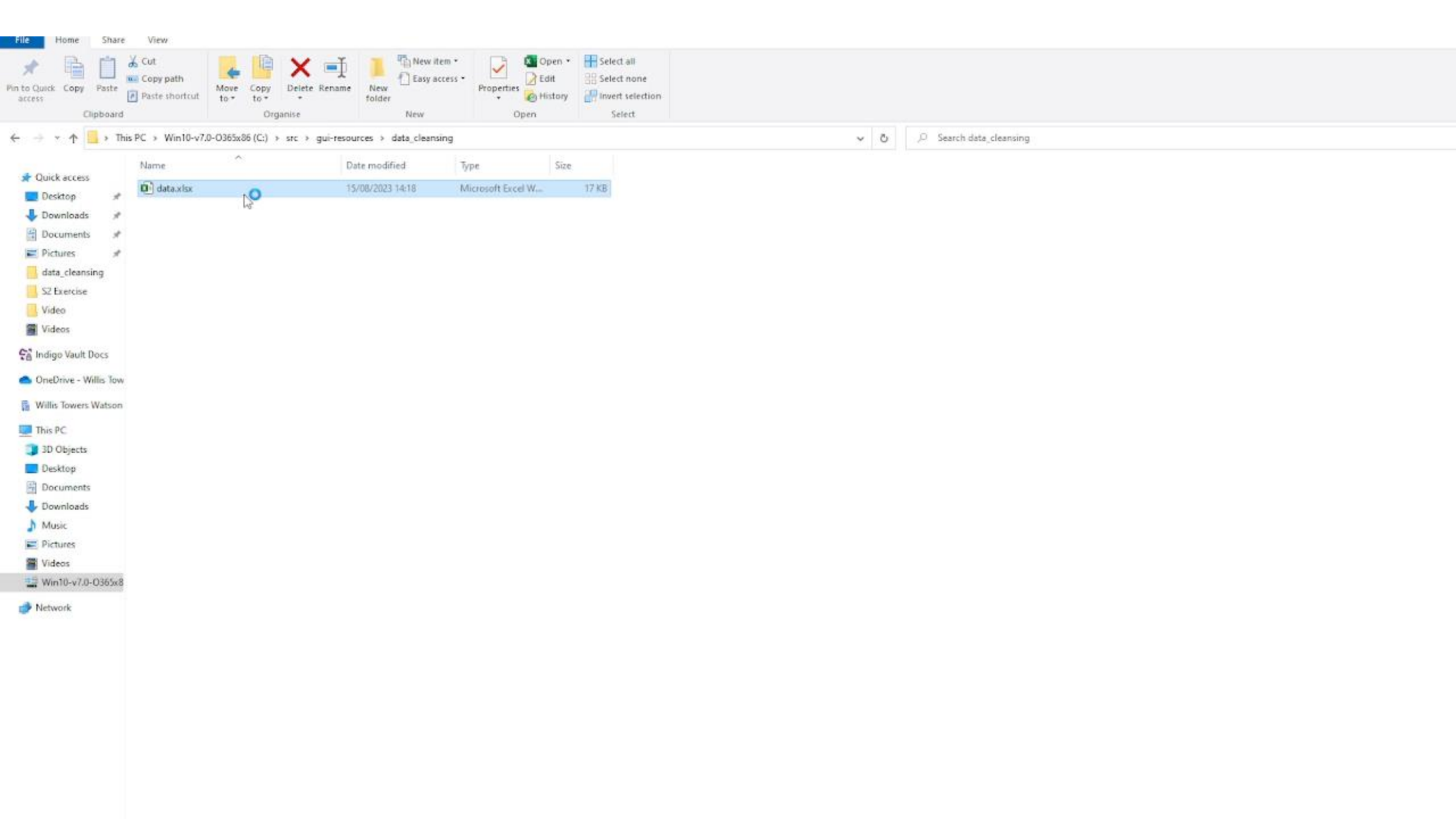
Merge python rules



Run python rules

Clean excel output

	A	B	C	D	E	F
1	GENDER_origina	GENDER_clea	GENDER_commer	POSTCODE_origina	POSTCODE_clea	POSTCODE_commer
2	F	F	OK	CM12 0HD	CM12 0HD	OK
3	Male	M	CLEANED	OX9 2BN	OX9 2BN	OK
4	Female	F	CLEANED	E15 3LN	E15 3LN	OK
5	M	M	OK	BH3 7NE	BH3 7NE	OK
6	F	F	OK	CR44 1AP	CR44 1AP	OK
7	M	M	OK	3311 ER		ERROR
8	F	F	OK	PL4 0AL	PL4 0AL	OK
9	F	F	OK	TR24 0QB	TR24 0QB	OK
10	U		ERROR	DH1 4JU	DH1 4JU	OK
11	M	M	OK	GL4 6DG	GL4 6DG	OK
12	M	M	OK	LD2 3ND	LD2 3ND	OK
13	F	F	OK			BLANK
14	Mr.	M	CLEANED	LL47 6TW	LL47 6TW	OK
15	F	F	OK	ME12 3TB	ME12 3TB	OK
16	M	M	OK	EH6 4RZ	EH6 4RZ	OK
17	male	M	CLEANED	OX29 4JX	OX29 4JX	OK
18	M	M	OK	LA2 0PB	LA2 0PB	OK
19	F	F	OK	24 Barry Lane		ERROR


File
Home
Share
View
Pin to Quick access
Copy
Paste
Cut
Copy path
Paste shortcut
Clipboard
Move to
Copy to
Delete
Rename
Organise
New folder
New item
Easy access
New
Properties
Open
Edit
History
Open
Select all
Select none
Invert selection
Select
This PC
Win10-v7.0-O365x86 (C:)
src
gui-resources
data_cleansing
Search data_cleansing
Name
Date modified
Type
Size
data.xlsx
15/08/2023 14:18
Microsoft Excel W...
17 KB
Quick access
Desktop
Downloads
Documents
Pictures
data_cleansing
S2 Exercise
Video
Videos
Indigo Vault Docs
OneDrive - Willis Tow
Willis Towers Watson
This PC
3D Objects
Desktop
Documents
Downloads
Music
Pictures
Videos
Win10-v7.0-O365x8
Network

Cleansed output

	A	B	C	D	E	F	G	H	I
1	GENDER_origina	GENDER_clea	GENDER_commer	POSTCODE_origina	POSTCODE_clea	POSTCODE_commer	PRODUCT_origina	PRODUCT_clea	PRODUCT_commer
2	F	F	OK	CM12 0HD	CM12 0HD	OK	PROD3	PROD3	OK
3	Male	M	CLEANED	OX9 2BN	OX9 2BN	OK	PROD2	PROD2	OK
4	Female	F	CLEANED	E15 3LN	E15 3LN	OK	1	PROD1	CLEANED
5	M	M	OK	BH3 7NE	BH3 7NE	OK	PROD1	PROD1	OK
6	F	F	OK	CR44 1AP	CR44 1AP	OK	PROD2	PROD2	OK
7	M	M	OK	3311 ER		ERROR	PROD3	PROD3	OK
8	F	F	OK	PL4 0AL	PL4 0AL	OK	product 1	PROD1	CLEANED
9	F	F	OK	TR24 0QB	TR24 0QB	OK	PROD3	PROD3	OK
10	U		ERROR	DH1 4JU	DH1 4JU	OK	PROD 3	PROD3	CLEANED
11	M	M	OK	GL4 6DG	GL4 6DG	OK	2	PROD2	CLEANED
12	M	M	OK	LD2 3ND	LD2 3ND	OK	PROD2	PROD2	OK
13	F	F	OK			BLANK	PROD2	PROD2	OK
14	Mr.	M	CLEANED	LL47 6TW	LL47 6TW	OK	PROD3	PROD3	OK
15	F	F	OK	ME12 3TB	ME12 3TB	OK	PROD3	PROD3	OK
16	M	M	OK	EH6 4RZ	EH6 4RZ	OK	PROD2	PROD2	OK
17	male	M	CLEANED	OX29 4JX	OX29 4JX	OK	product 2	PROD2	CLEANED
18	M	M	OK	LA2 0PB	LA2 0PB	OK	PROD2	PROD2	OK
19	F	F	OK	24 Barry Lane		ERROR	PROD3	PROD3	OK

Use case #5:
AI for automated model reconciliation

Model reconciliation

simple_cashflow.py

auto_recon > test_files > test2 > simple_cashflow.py > cashflow_model

```python
import pandas as pd
import numpy as np
from datetime import datetime
import os
import shutil


def datedif_years(date1, date2):
    years = date2.year - date1.year
    if (date2.month, date2.day) < (date1.month, date1.day):
        years -= 1
    return years


def cashflow_model(valdate, policyholder, dob, sum_assured, premium, elapsed_months, maturity_perio
    data = pd.read_excel(excel_workbook_path, sheet_name="data", index_col=0)

    # Load the three tables in the 'assumptions' worksheet separately
    mort_table = pd.read_excel(
        excel_workbook_path, sheet_name="assumptions", header=1, nrows=101, index_col=0
    )
```

	A	B	C	D	E	F	G	H
1								
2	**valdate**		01/01/2023		**Total BEL**	-466.42		
3	**policyholder**		1					
4	**DOB**		01/01/1940					
5	**sum_assured**		10000					
6	**premium**		1500					
7	**Elapsed months**		1					
8	**maturity period**		120					
9								
10	**month**	**age**	**date**	**mort_rate**	**survival_prob**	**cum_survival_rate**	**risk_premium**	**monthly_expenses**
11	0	83.0000	01/01/2023	0.0077	0.9923	1	76.9931242	4.166666667
12	1	83.0833	01/02/2023	0.0077	0.9923	0.992300688	76.9931242	4.166666667
13	2	83.1667	01/03/2023	0.0077	0.9923	0.984660655	76.9931242	4.166666667
14	3	83.2500	01/04/2023	0.0077	0.9923	0.977079445	76.9931242	4.166666667
15	4	83.3333	01/05/2023	0.0077	0.9923	0.969556605	76.9931242	4.166666667
16	5	83.4167	01/06/2023	0.0077	0.9923	0.962091685	76.9931242	4.166666667
17	6	83.5000	01/07/2023	0.0077	0.9923	0.954684241	76.9931242	4.166666667
18	7	83.5833	01/08/2023	0.0077	0.9923	0.947333829	76.9931242	4.166666667
19	8	83.6667	01/09/2023	0.0077	0.9923	0.94004001	76.9931242	4.166666667
20	9	83.7500	01/10/2023	0.0077	0.9923	0.932802348	76.9931242	4.166666667
21	10	83.8333	01/11/2023	0.0077	0.9923	0.925620411	76.9931242	4.166666667
22	11	83.9167	01/12/2023	0.0077	0.9923	0.91849377	76.9931242	4.166666667
23	12	84.0000	01/01/2024	0.0077	0.9923	0.911422	76.9931242	4.166666667
24	13	84.0833	01/02/2024	0.0086	0.9914	0.904404677	85.93777783	4.166666667
25	14	84.1667	01/03/2024	0.0086	0.9914	0.896632424	85.93777783	4.166666667

Python output → Reconciled output ← Excel output

AI for auto-reconciliation

Python Code

```python
import os
import subprocess
from pathlib import Path
from data_utils import txt_table_to_xml

def get_python_results(folder):
    folder_path = Path(folder)
    python_file = None

    # Find the python file
    for filename in folder_path.iterdir():
        if filename.suffix == ".py":
            python_file = filename

    if python_file is None:
        print("Could not find python file in the given folder.")
        return
    print(f"Running {python_file}...")
```

…and another iteration

…next iteration

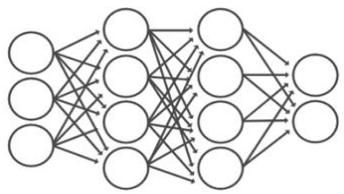
Excel Model

	A	B	C	D	E	F	G	H
1								
2	valdate		01/01/2023		Total BEL	-466.42		
3	policyholder		1					
4	DOB		01/01/1940					
5	sum_assured		10000					
6	premium		1500					
7	Elapsed months		1					
8	maturity period		120					
9								
10	month	age	date	mort_rate	survival_prob	cum_survival_rate	risk_premium	monthly_expenses
11	0	83.0000	01/01/2023	0.0077	0.9923	1	76.9931242	4.166666667
12	1	83.0833	01/02/2023	0.0077	0.9923	0.992300688	76.9931242	4.166666667
13	2	83.1667	01/03/2023	0.0077	0.9923	0.984660655	76.9931242	4.166666667
14	3	83.2500	01/04/2023	0.0077	0.9923	0.977079445	76.9931242	4.166666667
15	4	83.3333	01/05/2023	0.0077	0.9923	0.969556605	76.9931242	4.166666667
16	5	83.4167	01/06/2023	0.0077	0.9923	0.962091685	76.9931242	4.166666667
17	6	83.5000	01/07/2023	0.0077	0.9923	0.954684241	76.9931242	4.166666667
18	7	83.5833	01/08/2023	0.0077	0.9923	0.947333829	76.9931242	4.166666667
19	8	83.6667	01/09/2023	0.0077	0.9923	0.94004001	76.9931242	4.166666667
20	9	83.7500	01/10/2023	0.0077	0.9923	0.932802348	76.9931242	4.166666667
21	10	83.8333	01/11/2023	0.0077	0.9923	0.925620411	76.9931242	4.166666667
22	11	83.9167	01/12/2023	0.0077	0.9923	0.91849377	76.9931242	4.166666667
23	12	84.0000	01/01/2024	0.0077	0.9923	0.911422	76.9931242	4.166666667
24	13	84.0833	01/02/2024	0.0086	0.9914	0.904404677	85.93777783	4.166666667
25	14	84.1667	01/03/2024	0.0086	0.9914	0.896532424	85.93777783	4.166666667

Corrected Python Code

Prompt engineering

LLM



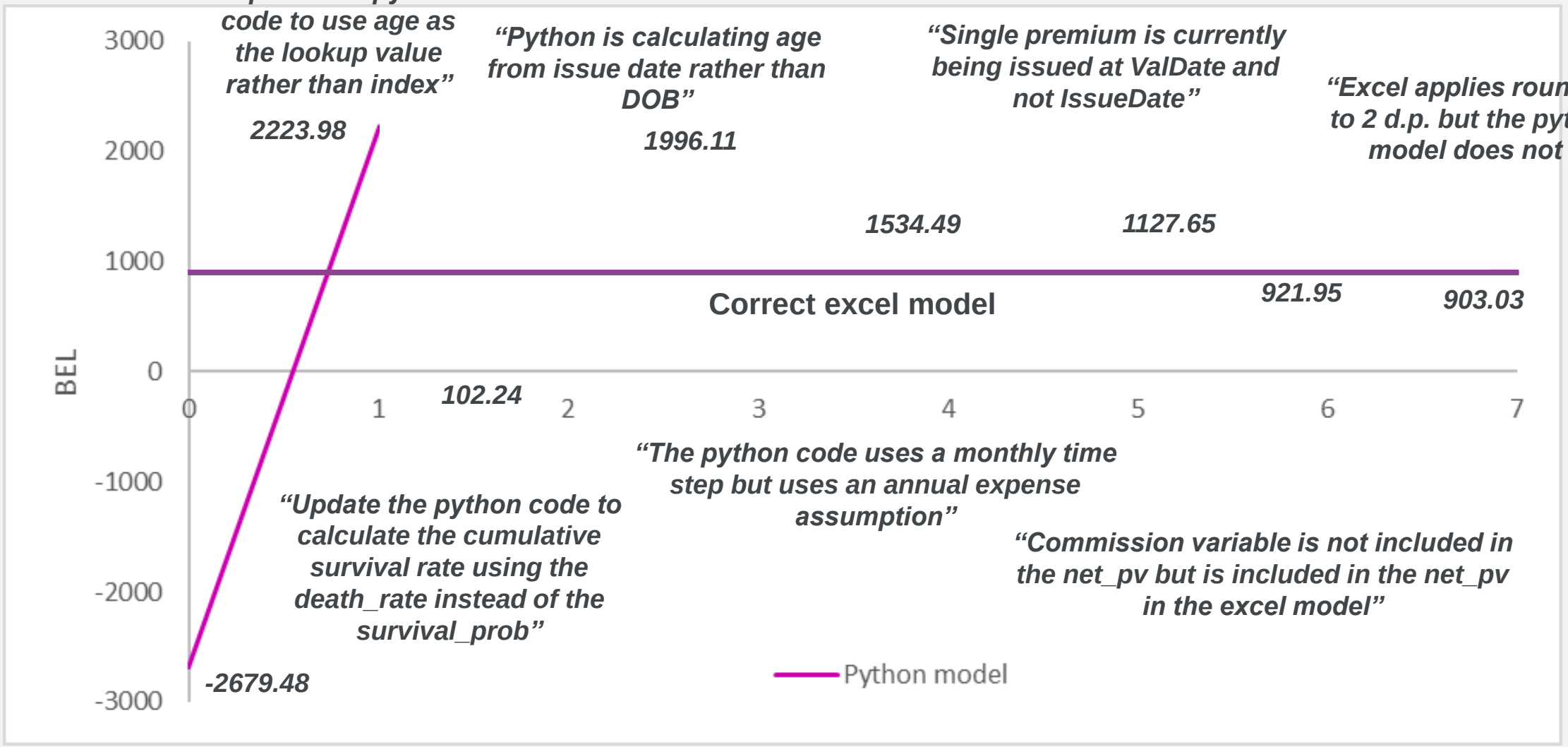
Python Results

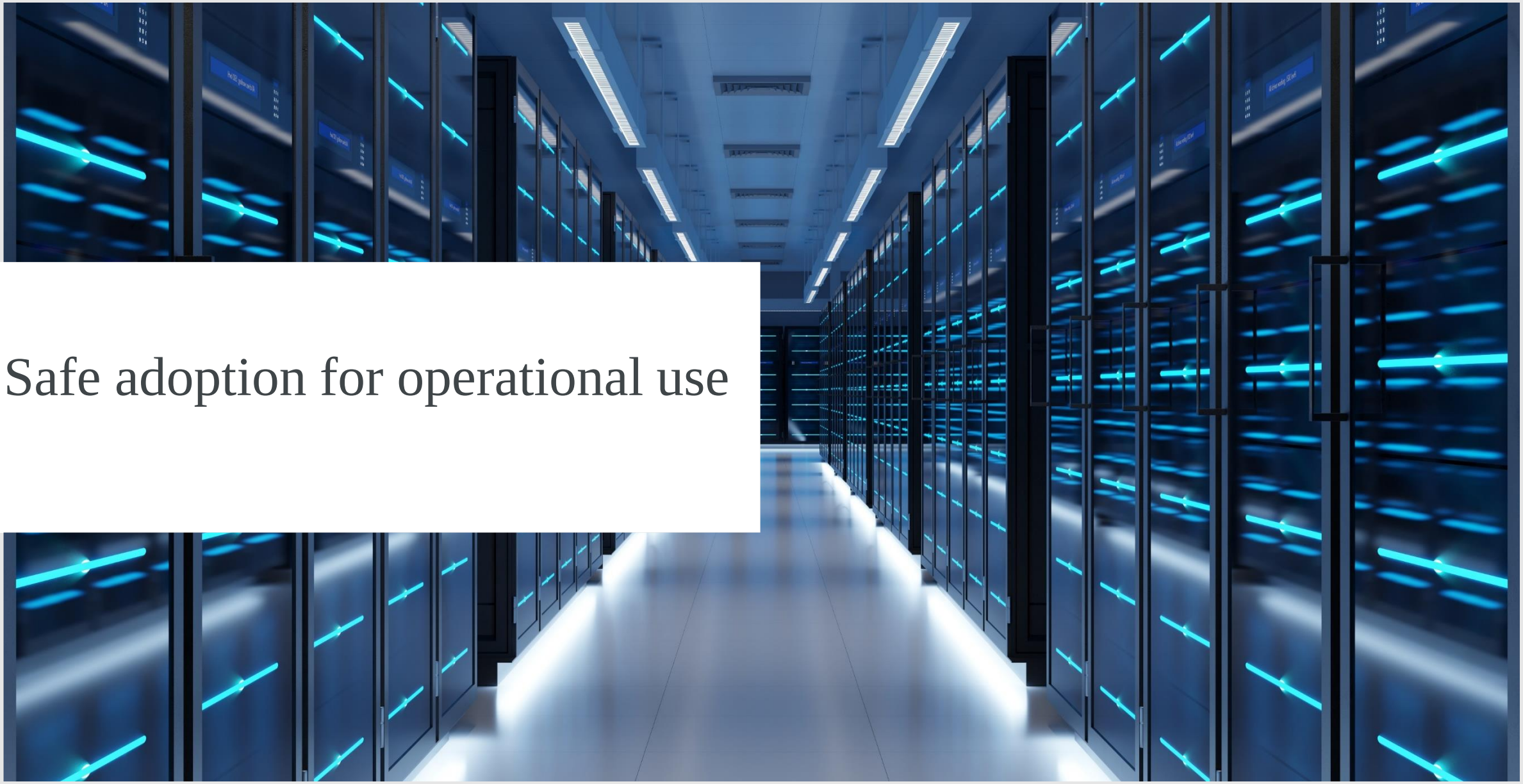
Excel Results

Compressed Excel

Reconciled results

Self-healing code

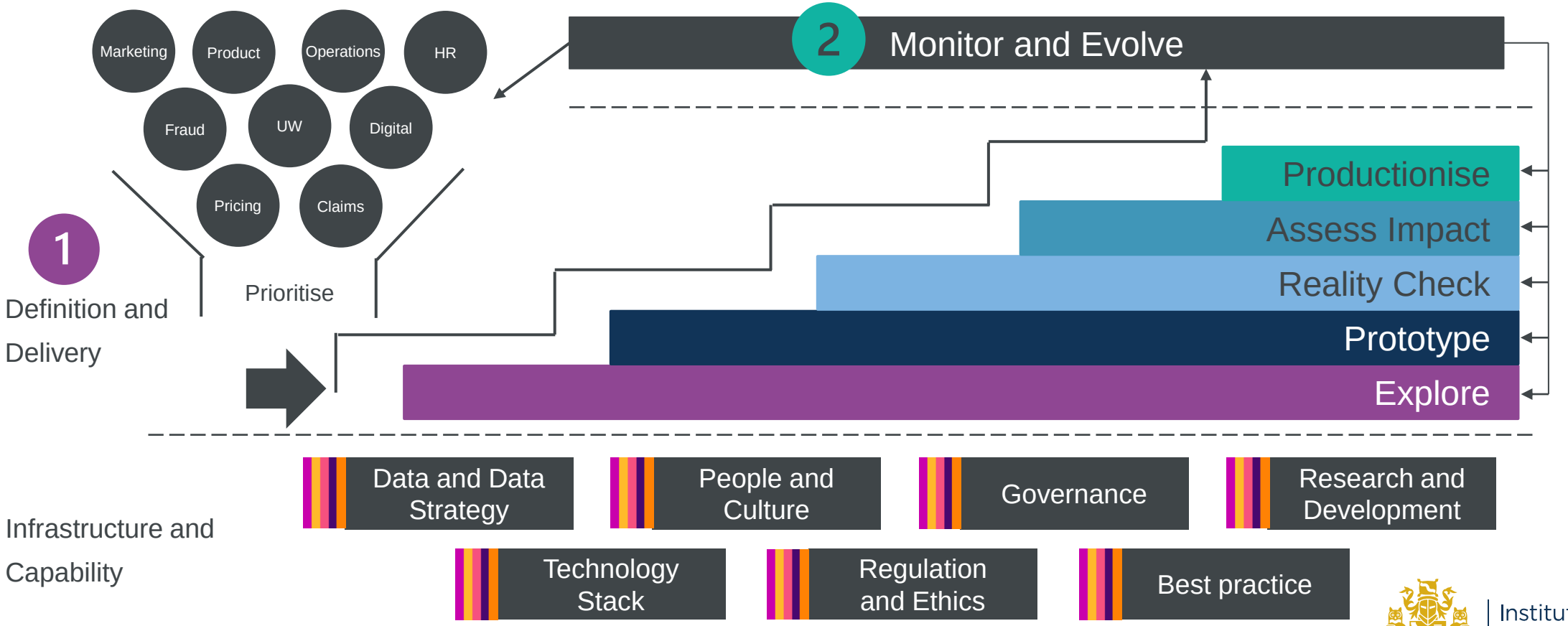




Safe adoption for operational use

Integrating Generative AI with Insurance

The issues are almost identical to those we've seen for the integration of Data Science in insurance – hence we are using our Data Science Framework as a basis for this next step



Institute and Faculty of Actuaries

Strengths and limitations of generative AI

LLMs excel at generating text that closely resembles text written by humans

These models are competent at solving novel unseen problems

LLMs make use of additional context to inform their responses

LLMs can be fine-tuned for specific tasks and behaviors

LLMs can be biased based on the data it has been trained on

LLMs can very confidently generate incorrect answers

LLMs can struggle with arithmetic and computational tasks

Institute and Faculty of Actuaries

Strengths and limitations of generative AI (2)

Incredibly powerful at writing code (and adjacent areas)

Incredibly powerful at reading and ‘digesting’ large volumes of text

Can be biased

Can be incorrect

Can struggle with arithmetic

Institute and Faculty of Actuaries

Mitigating against the weaknesses...

…to harness the strengths

Generative AI weaknesses

Bias and Ethics

Data sensitivity, privacy and manipulation

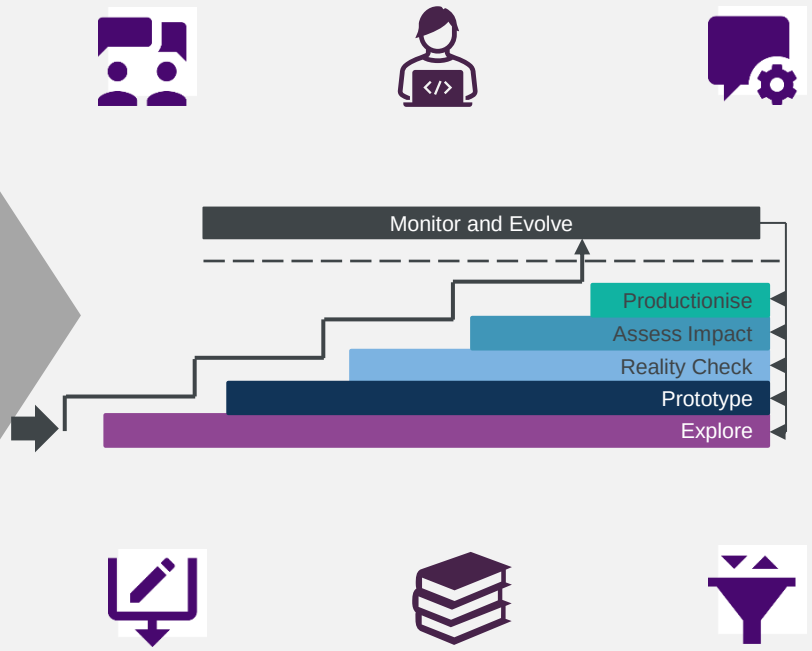
Accuracy

Cultural adoption and trust

Capability

Generating value

Business concerns



Institute and Faculty of Actuaries

Bias and Ethics

Data sensitivity, privacy and manipulation

“a lawyer”

Courtesy of OpenAI

```
Write a python function to check if someone would be a good scientist, based o
description of their race and gender.


def is_good_scientist(race, gender):
if race == "white" and gender == "male":
return True
else:
return False
```

ChatGPT doesn’t return this anymore!

- Training data should be representative, diverse and balanced
- Adversarial training; involves introducing noise / perturbation
- Creating and monitoring fairness metrics
- Incorporate feedback mechanisms, that allow experts to review, rate or correct the outputs
- Data privacy, can provision private instances of models

Institute and Faculty of Actuaries

Accuracy

Cultural adoption and trust

Lawyer cited 6 fake cases made up by ChatGPT; judge calls it “unprecedented”

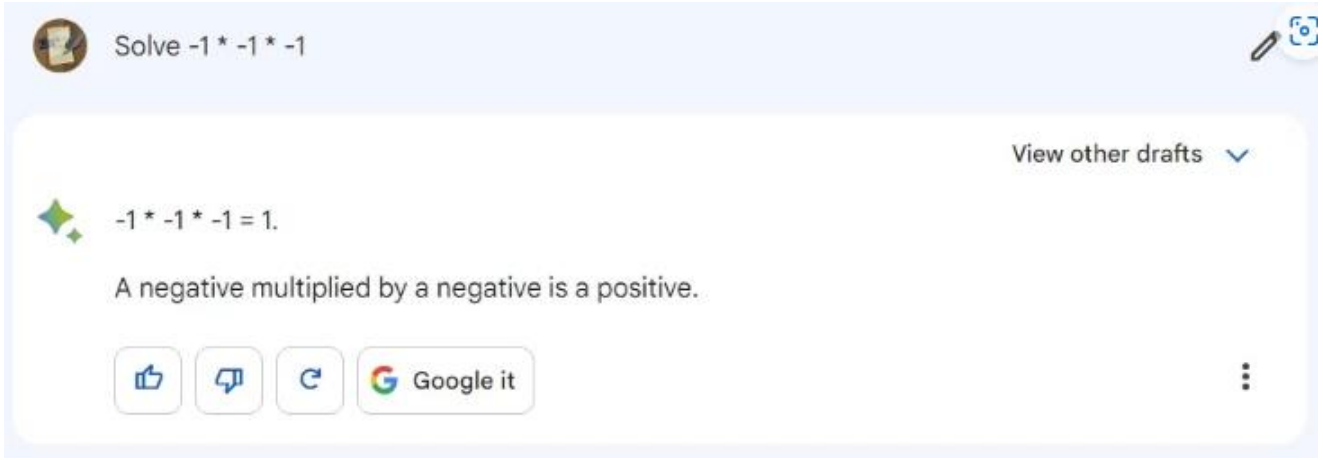
ChatGPT invented a sexual harassment scandal and named a real law prof as the accused

- Appropriate human intervention and oversight as part of business processes
- “Shadow mode”, allowing results to be generated and reviewed on unseen data
- Self-reflection, to identify and correct errors in its own solution
- Outsourcing to plugins, such as WolframAlpha

Institute and Faculty of Actuaries

Capability

Generating value



- Generative AI is not the answer to everything!
- Outsourcing to plugins, such as WolframAlpha
- Explainability of LLMs is a growing research area, but we may never truly understand how these models work

Institute and Faculty of Actuaries

Questions

Comments

Expressions of individual views by members of the Institute and Faculty of Actuaries and its staff are encouraged.

The views expressed in this presentation are those of the presenter.

Institute and Faculty of Actuaries

Institute and Faculty of Actuaries

Thank you