

# Generative AI Apps for Automatic Petrophysical Data Analysis and Machine Learning Modelling

Marcelo Guarido\*, Kristopher Innanen\*

\*CREWES – University of Calgary

## Summary

AI assistants powered by large-language models have been applied to different fields for quick analyses, ~~data~~ summarization, and advice chatbots. We proposed an ensemble of AI assistants for an app for Well-log analysis and data modelling. It works as a chat assistant that can access SQL databases to extract data and process it to return a table, a chart, or a machine-learning model based on the user's request. Our tests showed that the app could yield satisfactory results for the three output requests, but it required trial and error in asking the question or request. This application can create complex modelling and analyses for wireline logs without programming, making it accessible to a more significant number of geoscientists.

## Method

Since the release of the paper “Attention in All You Need” (Vaswani, 2017), transformer models have been widely used for natural language processing to create AI chat assistants. This deep learning structure is the foundation for the Large-Language Models (LLM) such as the GPT family, from OpenAI, or open-source models like Llama (Touvron et al., 2023), from Meta, which are pre-trained models trained on extensive datasets, equivalent to internet-scale size, and they can be used as APIs for different applications. Those are generalized models that can create content on different subjects. Although, some applications require the use of IP data that were not used to train the model. To create the apps, there are two solutions: fine-tuning the model with the IP data (Peng et al., 2023), which can be expensive, or using Retrieval-Augmented Generation, aka RAG (Lewis et al., 2020), which enhances response accuracy by biasing outputs based on a designated set of documents, relying solely on prompt-engineering.

We are proposing an app that performs automatic petrophysical analysis and machine learning modelling powered by GPT-4o-mini. It consists of an ensemble of AI agents (small AI assistants that work on specific tasks) that access a SQL database containing wireline log data from multiple wells, and can generate processed tables, charts, and/or machine learning predictions based on a user typing request, like a chat assistant. Internally, the app generates SQL and Python codes that are executed, and the results are printed to the user.

## Results

Tests of the CREWES Well-Log AI Assistant were done using wireline log data from the 2016 SEG ML Contest (Hall, 2016; Hall and Hall, 2017), that are well-logs from the Hugoton and Panoma gas fields (Dubois et al., 2007). It contains measurements from 10 well locations, summing five wireline logs (GR, ILD\_log10, DeltaPHI, PHIND, and PE), two indicators (NM\_M, RELPOS), the depth, formation, and the facies from core analysis (Figure 2). The data was

originally a CSV file, but we converted it to an SQL database in an SQL table called curves for the tests.

The first test was to access the table capabilities of the process. A request that "activates" this path should make the method return a table or statistical data description. Internally, it will generate an SQL code to access the data. We made the following request:

*"What are the columns and their data types in the database?"*

Figure 1 is the output generated by the app: data info of all the columns presented in the data, pointing the columns names, data types, and missing values.

```
Data:
  cid  name      type  notnull  dflt_value  pk
0    0  Facies  INTEGER    0         None      0
1    1  Formation TEXT        0         None      0
2    2  Well_Name TEXT        0         None      0
3    3   Depth   REAL        0         None      0
4    4    GR     REAL        0         None      0
5    5  ILD_log10 REAL        0         None      0
6    6  DeltaPHI  REAL        0         None      0
7    7   PHIND  REAL        0         None      0
8    8    PE     REAL        0         None      0
9    9   NM_M  INTEGER    0         None      0
10   10  RELPOS  REAL        0         None      0
```

Figure 1: Output presented to the user when asked what column names and data types are in the database.

Testing the chart capabilities of the application requires a request that will turn the "chart" path on (the app decides to make a plot and generates a Python code for it). For that, we made the following request:

*"Print a static heatmap, using seaborn, with the Spearman correlation between all the numerical variables. Use cmap = 'Spectral'."*

When we asked for a chat using Seaborn, it meant that it was a static plot. We can also specify some plot arguments, as the selection Spectral for the plot colormap and use the Spearman correlation for non-linear correlations instead of the default Pearson. Our tests show some trials and errors in asking questions to get the desired output. Figure 2 is the correlation plot requested, using Spearman correlation. The LLM also returned a plot including a title for the plot and axis labels.

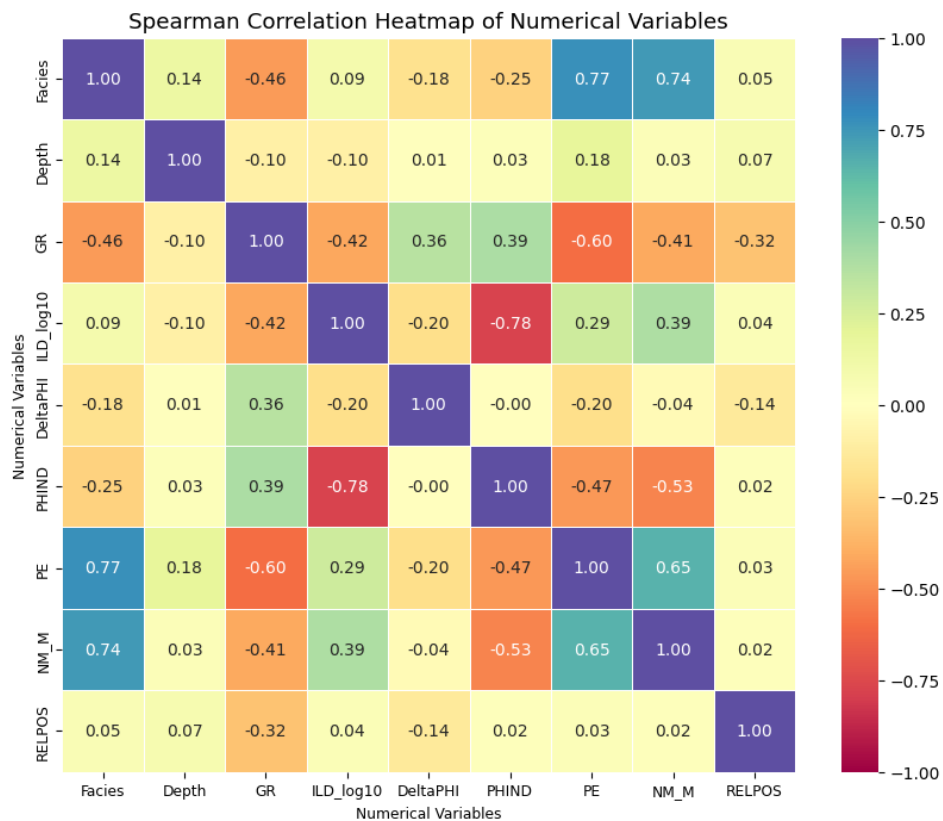


Figure 2: Chart generated by the Chart AI Assistant to plot the Spearman correlation between all the numerical columns in the data.

Finally, the final test is to test the application’s *machine learning* capabilities. For that, the ML AI Assistant is set to (using prompt engineering) access the data automatically, split it into train (80% of the rows) and validation (20% of the rows), apply the one-hot-encoding on the categorical columns, train the model on the training set, make predictions on the train/validation data, and return a table with the original data with two additional columns: the predicted values and a tag of train/validation. For the test, we sent the following request:

*“From the database, use the XGBClassifier to predict the Facies column. Remember that XGBClassifier needs the target classes starting from 0. Do the conversion for the predictions and then return to the original classes.”*

Note that in the request we specified which model to use, some considerations of the model (classes numbering), and the target column. As no hyperparameters were defined, the XGBClassifier is trained using the default values. The app will extract the data from the SQL database and then generate and execute Python code to do the machine learning modelling requested. Figure 3 is the output table with the original data plus the prediction columns.

	Facies	Formation	Well_Name	Depth	GR	ILD_log10	DeltaPHI	PHIND	PE	NM_M	RELPOS	Set	Predicted_Facies
0	2	A1 SH	churchman bible	2917.5	63.500	0.705	4.916	12.174	3.162	1	1.000	validation	2
1	2	A1 SH	churchman bible	2918.0	80.875	0.709	7.514	13.286	2.932	1	0.950	train	2
2	2	A1 SH	churchman bible	2918.5	87.813	0.700	7.785	13.212	2.953	1	0.900	train	2
3	2	A1 SH	churchman bible	2919.0	86.750	0.685	7.547	13.547	3.420	1	0.850	train	2
4	2	A1 SH	churchman bible	2919.5	79.438	0.669	5.912	12.252	3.324	1	0.800	train	2
...	...	...	...	...	...	...	...	...	...	...	...	...	...
399	4	C LM	churchman bible	3120.5	46.719	0.947	1.828	7.254	3.617	2	0.685	validation	4
400	4	C LM	churchman bible	3121.0	44.563	0.953	2.241	8.013	3.344	2	0.677	validation	4
401	4	C LM	churchman bible	3121.5	49.719	0.964	2.925	8.013	3.190	2	0.669	train	4
402	4	C LM	churchman bible	3122.0	51.469	0.965	3.083	7.708	3.152	2	0.661	train	4
403	4	C LM	churchman bible	3122.5	50.031	0.970	2.609	6.668	3.295	2	0.653	train	4

Figure 3: Results generated by the ML AI Assistant. The original data merged with the predicted facies and a tag for train and validation.

This application's potential is to access large SQL (or other types of) databases containing multiple well locations. The user could request wells by location or name and create different analyses and modelling by asking the chat assistant, reducing the execution time without the need for programming. This could be done with other data types, such as seismic shots, gravimetric surveys, GPR, etc.

## Conclusion

AI assistants powered by large-language models have been applied to different fields for quick analyses, data summarization, and advice chatbots. We proposed an ensemble of AI assistants for an app for Well-log analysis and data modelling. It works as a chat assistant that can access SQL databases to extract data and process it to return a table, a chart, or a machine-learning model based on the user's request. Our tests showed that the app could yield satisfactory results for the three output requests, but it required trial and error in asking the question or request. This application can create complex modelling and analyses for wireline logs without programming, making it accessible to a more significant number of geoscientists.

## Acknowledgements

We extend our heartfelt thanks to the sponsors of CREWES for their invaluable support. This work was made possible by generous funding from CREWES industrial sponsors and NSERC (Natural Science and Engineering Research Council of Canada) through the grant CRDPJ 543578-19. We also acknowledge the additional support provided by Emissions Reduction Alberta through the ACT4-SPARSE project.

## References

- Dubois, M. K., Bohling, G. C., and Chakrabarti, S., 2007, Comparison of four approaches to a rock facies classification problem: *Computers Geosciences*, 33, No. 5, 599–617.
- Hall, B., 2016, Facies classification using machine learning: *The Leading Edge*, 35, No. 10, 906–909.
- Hall, M. and Hall, B., 2017. Distributed collaborative prediction: Results of the machine learning contest. *The Leading Edge*, 36(3), pp.267-269.

Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., Küttler, H., Lewis, M., Yih, W.-t., Rocktäschel, T. et al., 2020, Retrieval-augmented generation for knowledge-intensive nlp tasks: Advances in Neural Information Processing Systems, 33, 9459–9474.

Peng, B., Li, C., He, P., Galley, M., & Gao, J. (2023). Instruction tuning with gpt-4. arXiv preprint arXiv:2304.03277

Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M.-A., Lacroix, T., Rozière, B., Goyal, N., Hambro, E., Azhar, F. et al., 2023, Llama: Open and efficient foundation language models: arXiv preprint arXiv:2302.13971.

Vaswani, A. (2017). Attention is all you need. Advances in Neural Information Processing Systems.