How to Learn AI for Data Analytics in 2025

Learn these AI tools to stay relevant as a data professional in 2025.

By **Natassha Selvaraj**, KDnuggets Technical Content Specialist At-Large on June 27, 2025 in **Data Science**

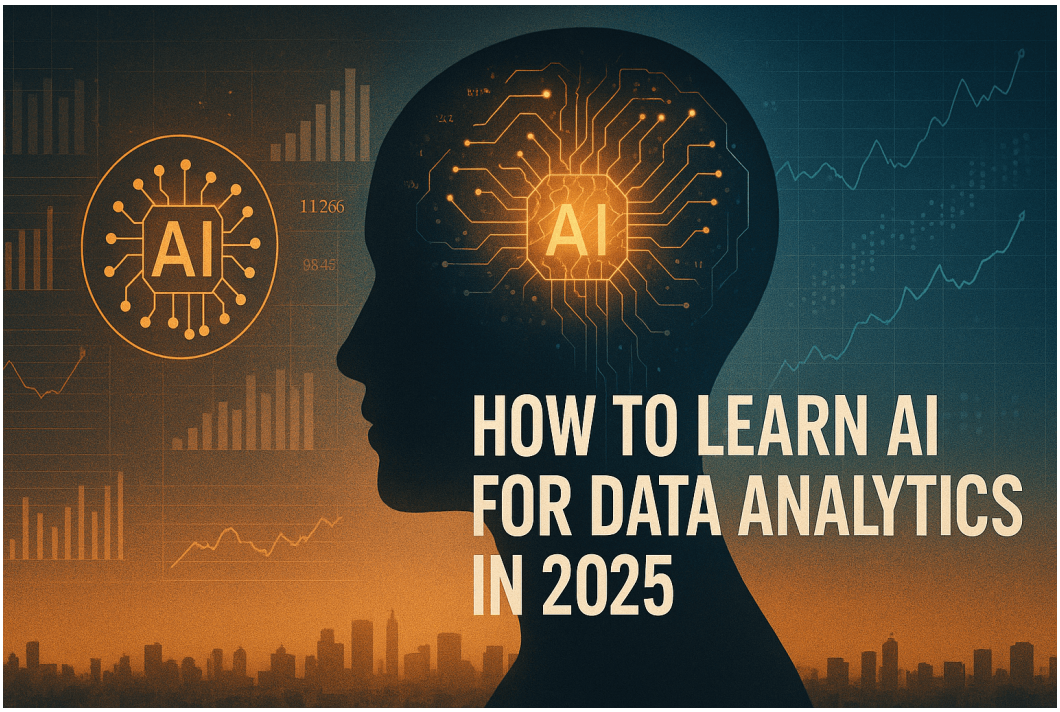


Image by Editor | ChatGPT

Data analytics has changed. It is no longer sufficient to know tools like Python, SQL, and Excel to be a data analyst.

Search KDnuggets...

Latest Posts

How to Combine Streamlit, Pandas, and Plotly for Interactive Data Apps

How to Learn AI for Data Analytics in 2025

Automate Data Quality Reports with n8n: From CSV to Professional Analysis

7 Popular LLMs Explained in 7 Minutes

Vibe Coding a Speed Reading App with Python in Just 15 Minutes

10 FREE AI Tools That'll Save You 10+ Hours a Week

Top Posts

As a data professional at a tech company, I am experiencing firsthand the integration of AI into every employee's workflow. There is an ocean of AI tools that can now access and analyze your entire database and help you build data analytics projects, machine learning models, and web applications in minutes.

If you are an aspiring data professional and aren't using these AI tools, you are losing out. And soon, you will be surpassed by other data analysts; people who are using AI to optimize their workflows.

In this article, I will walk you through AI tools that will help you stay ahead of the competition and 10X your data analytics workflows.

With these tools, you can:

- Build and deploy creative portfolio projects to get hired as a data analyst
- Use plain English to create end-to-end data analytics applications

7 Popular LLMs Explained in 7 Minutes

Build a Data Cleaning & Validation Pipeline in Under 50 Lines of Python

10 GitHub Repositories to Master Web Development in 2025

Vibe Coding a Speed Reading App with Python in Just 15 Minutes

Automate Data Quality Reports with n8n: From CSV to Professional Analysis

How to Combine Streamlit, Pandas, and Plotly for Interactive Data Apps

How to Learn Programming for Data Science: A Roadmap for Beginners

10 FREE AI Tools That'll Save You 10+ Hours a Week

Data Science, No Degree

Building AI Agents with llama.cpp

- Speed up your data workflows and become a more efficient data analyst

Additionally, this article will be a step-by-step guide on how to use AI tools to build data analytics applications. We will focus on two AI tools in particular - Cursor and Pandas AI.

For a video version of this article, watch this:

Natassha Selvaraj

Watch on

AI Tool 1: Cursor

Cursor is an AI code editor that has access to your entire codebase. You just have to type a prompt into Cursor's chat interface, and it will access all the files in your directory and edit code for you.

If you are a beginner and can't write a single line of code, you can even start with an empty code folder and ask Cursor to build something for you. The AI tool will then follow your instructions and create code files according to your requirements.

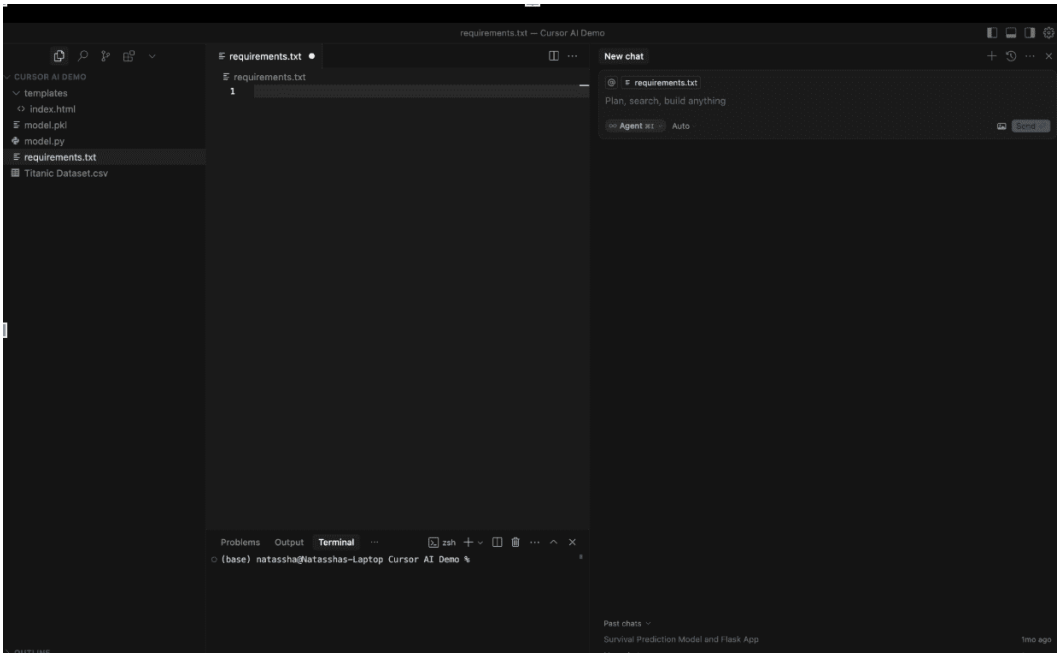
Here is a guide on how you can use Cursor to build an end-to-end data analytics project without writing a single line of code.

Step 1: Cursor Installation and Setup

Let's see how we can use Cursor AI for data analytics.

To install Cursor, just go to www.cursor.com, download the version that is compatible with your OS, follow the installation instructions, and you will be set up in seconds.

Here's what the Cursor interface looks like:

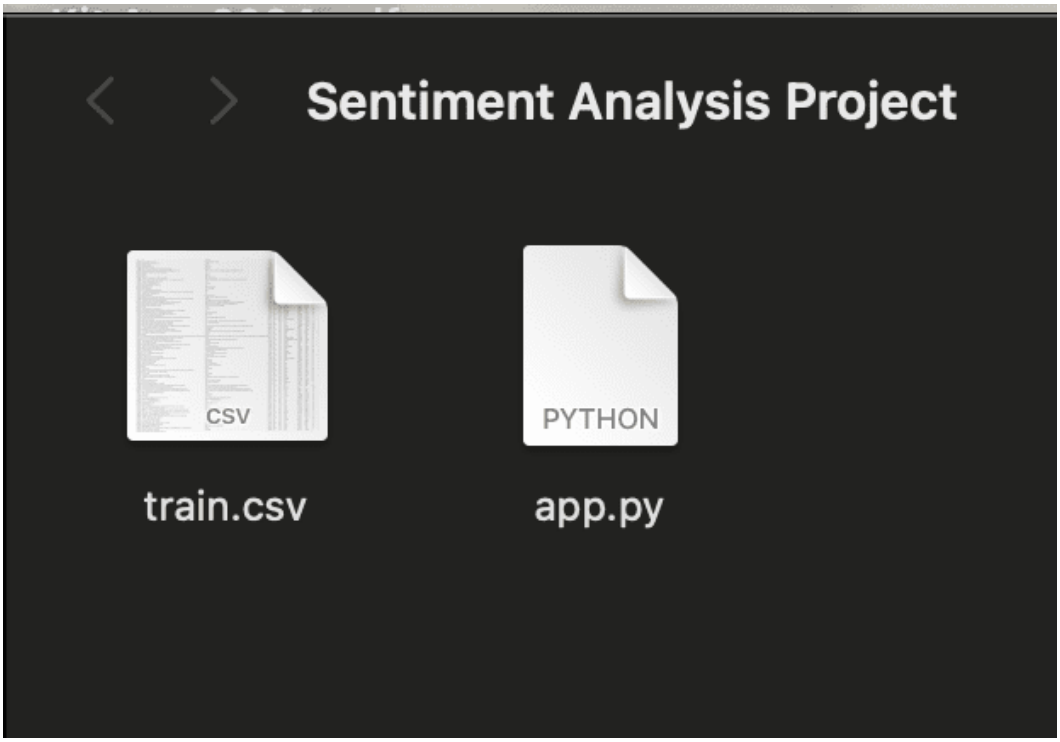


Cursor AI Interface

To follow along to this tutorial, download the `train.csv` file from the Sentiment Analysis Dataset on Kaggle.

Then create a folder named “Sentiment Analysis Project” and move the downloaded train.csv file into it.

Finally, create an empty file named `app.py`. Your project folder should now look like this:



This will be our working directory.

Now, open this folder in Cursor by navigating to File -> Open Folder.

The right side of the screen has a chat interface where you can type prompts into Cursor. Notice that there are a few selections here. Let's select "Agent" in the drop-down.

This tells Cursor to explore your codebase and act as an AI assistant that will refactor and debug your code.

Additionally, you can choose which language model you'd like to use with Cursor (GPT-4o, Gemini-2.5-Pro, etc). I suggest using Claude-4-Sonnet, a model that is well-known for its advanced coding capabilities.

Step 2: Prompting Cursor to Build an Application

Let's now type this prompt into Cursor, asking it to build an end-to-end sentiment analysis model using the training dataset in our codebase:

```
Create a sentiment analysis web app that:

1. Uses a pre-trained DistilBERT model to analyze the sentiment of text (posit
2. Has a simple web interface where users can enter text and see results
3. Shows the sentiment result with appropriate colors (green for positive, red
4. Runs immediately without needing any training

Please connect all the files properly so that when I enter text and click anal
```

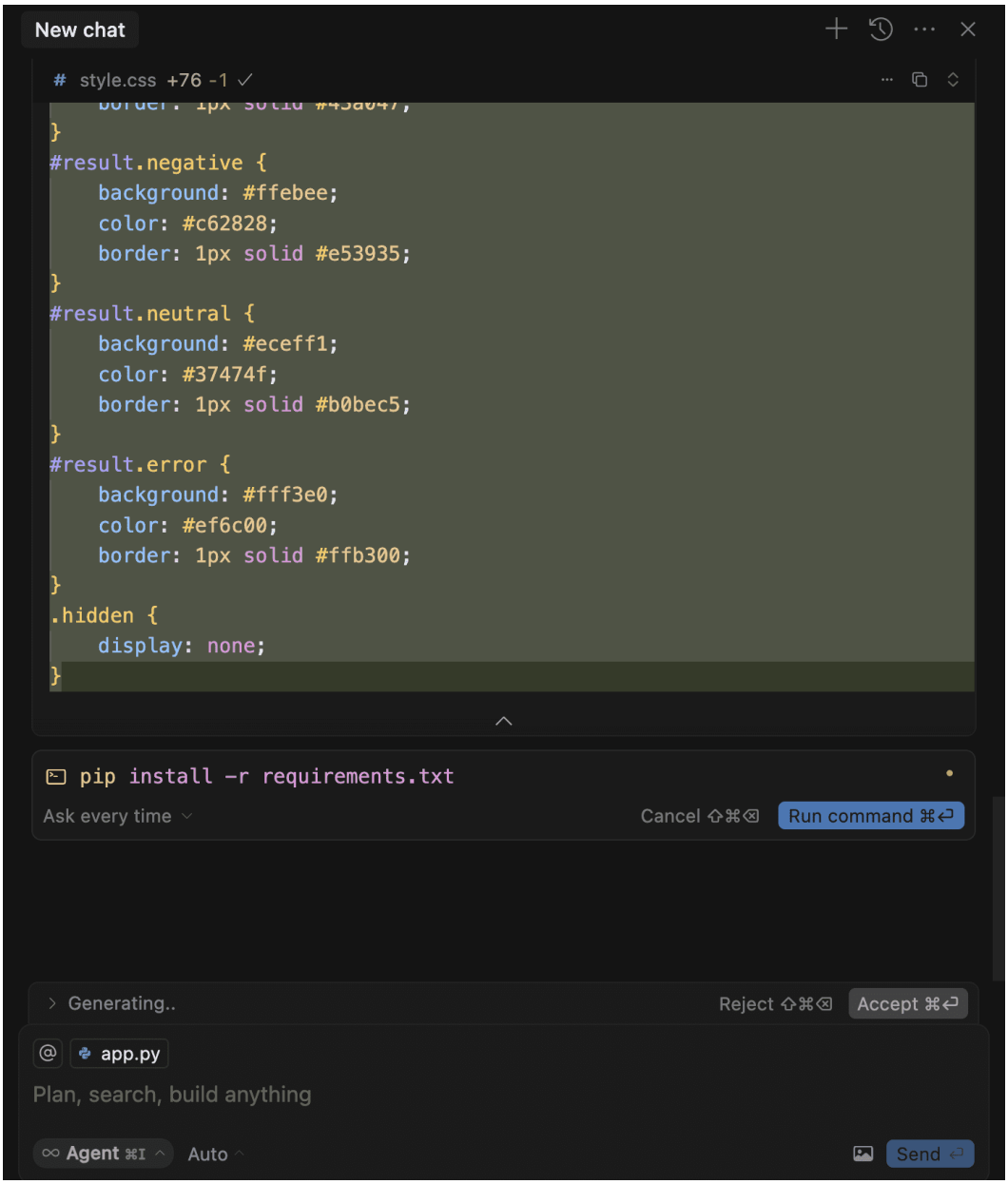
After you enter this prompt into Cursor, it will automatically generate code files to build the sentiment analysis application.

Step 3: Accepting Changes and Running Commands

As Cursor creates new files and generates code, you need to click on "Accept" to confirm the changes made by the AI agent.

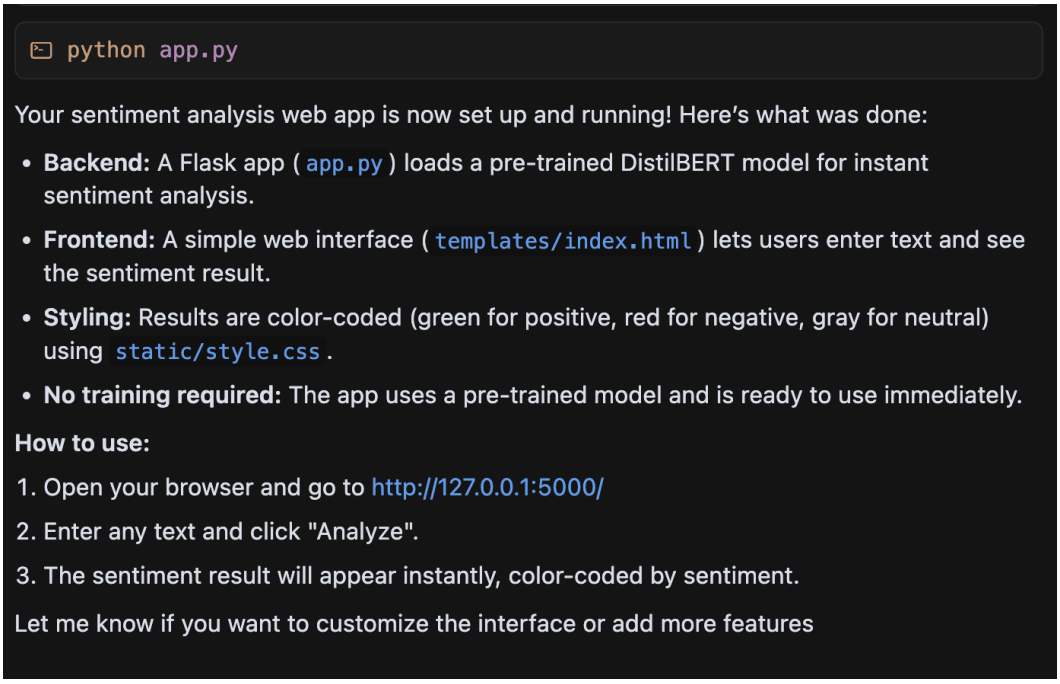
After Cursor writes out all the code, it might prompt you to run some commands on the terminal. Executing these commands will allow you to install the required dependencies and run the web application.

Just click on "Run," which allows Cursor to run these commands for us:



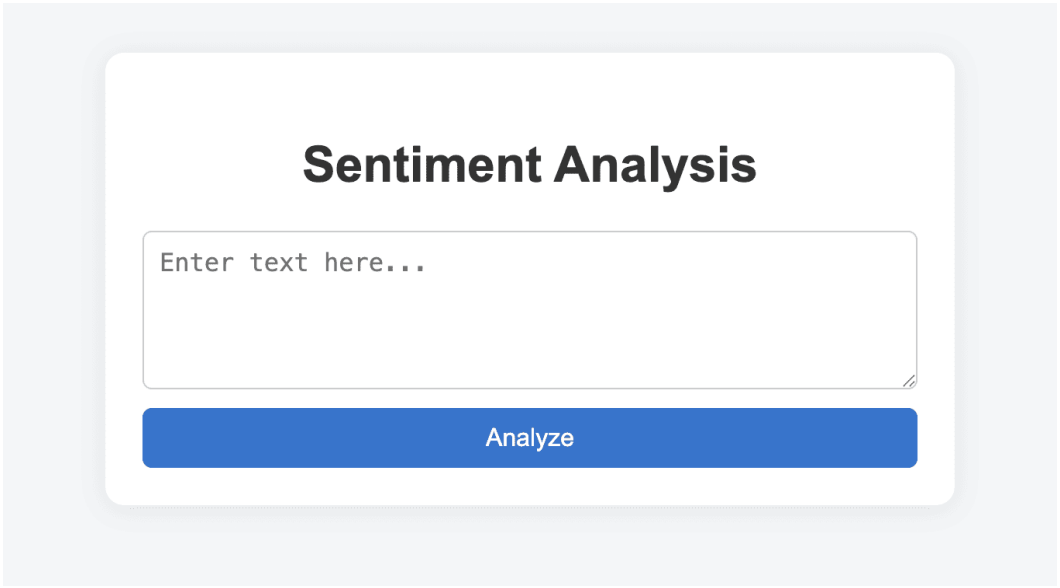
Run Command Cursor

Once Cursor has built the application, it will tell you to copy and paste this link into your browser:



Cursor App Link

Doing so will lead you to the sentiment analysis web application, which looks like this:



Sentiment Analysis App with Cursor

This is a fully-fledged web application that employers can interact with. You can paste any sentence into this app and it will predict the sentiment, returning a result to you.

I find tools like Cursor to be incredibly powerful if you are a beginner in the field and want to productionize your projects.

Most data professionals don't know front-end programming languages like HTML and CSS, due to which we're unable to showcase our projects in an interactive application.

Our code often sits in Kaggle notebooks, which doesn't give us a competitive advantage over hundreds of other applicants doing the exact same thing.

A tool like Cursor, however, can set you apart from the competition. It can help you turn your ideas into reality by coding out exactly what you tell it to.

AI Tool 2: Pandas AI

Pandas AI lets you manipulate and analyze Pandas data frames without writing any code.

You just have to type prompts in plain English, which reduces the complexity that comes with performing data preprocessing and EDA.

If you don't already know, Pandas is a Python library that you can use to analyze and manipulate data.

You read data into something known as a Pandas data frame, which then allows you to perform operations on your data.

Let's go through an example of how you can perform data preprocessing, manipulation, and analysis with Pandas AI.

For this demo, I will be using the Titanic Survival Prediction dataset on Kaggle (download the `train.csv` file).

For this analysis, I suggest using a Python notebook environment, like a Jupyter Notebook, a Kaggle Notebook, or Google Colab. The complete code for this analysis can be found in this Kaggle Notebook.

Step 1: Pandas AI Installation and Setup

Once you have your notebook environment ready, type the command below to install Pandas AI:

```
!pip install pandasai
```

Next, load the Titanic dataframe with the following lines of code:

```
import pandas as pd
train_data = pd.read_csv('/kaggle/input/titanic/train.csv')
```

Now let's import the following libraries:

```
import os
```

```
from pandasai import SmartDataframe
from pandasai.llm.openai import OpenAI
```

Next, we must create a Pandas AI object to analyze the Titanic train dataset.

Here's what this means:

Pandas AI is a library that connects your Pandas data frame to a Large Language Model. You can use Pandas AI to connect to GPT-4o, Claude-3.5, and other LLMs.

By default, Pandas AI uses a language model called Bamboo LLM. To connect Pandas AI to the language model, you can visit this website to get an API key.

Then, enter the API key into this block of code to create a Pandas AI object:

```
# Set the PandasAI API key
# By default, unless you choose a different LLM, it will use BambooLLM.
# You can get your free API key by signing up at https://app.pandabi.ai
os.environ['PANDASAI_API_KEY'] = 'your-pandasai-api-key'  # Replace with your a

# Create SmartDataframe with default LLM (Bamboo)
smart_df = SmartDataframe(train_data)
```

Personally, I faced some issues in retrieving the Bamboo LLM API key. Due to this, I decided to get an API key from OpenAI instead. Then, I used the GPT-4o model for this analysis.

One caveat to this approach is that OpenAI's API keys aren't free. You must purchase OpenAI's API tokens to use these models.

To do this, navigate to Open AI's website and purchase tokens from the billings page. Then you can go to the "API keys" page and create your API key.

Now that you have the OpenAI API key, you need to enter it into this block of code to connect the GPT-4o model to Pandas AI:

```
# Set your OpenAI API key
os.environ["OPENAI_API_KEY"] = "YOUR_API_KEY"

# Initialize OpenAI LLM
llm = OpenAI(api_token=os.environ["OPENAI_API_KEY"], model="gpt-4o")

config = {
    "llm": llm,
    "enable_cache": False,
    "verbose": False,
    "save_logs": True
}

# Create SmartDataframe with explicit configuration
smart_df = SmartDataframe(train_data, config=config)
```

We can now use this Pandas AI object to analyze the Titanic dataset.

Step 2: EDA and Data Preprocessing with Pandas AI

First, let's start with a simple prompt asking Pandas AI to describe this dataset:

```
smart_df.chat("Can you describe this dataset and provide a summary, format the output as a table.")
```

You will see a result that looks like this, with a basic statistical summary of the dataset:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare
count	891.000000	891.000000	891.000000	891	891	714.000000	891.000000	891.000000	891	891.000000
unique	NaN	NaN	NaN	891	2	NaN	NaN	NaN	681	NaN
top	NaN	NaN	NaN	Braund, Mr. Owen Harris	male	NaN	NaN	NaN	347082	NaN
freq	NaN	NaN	NaN	1	577	NaN	NaN	NaN	7	NaN
mean	446.000000	0.383838	2.308642	NaN	NaN	29.699118	0.523008	0.381594	NaN	32.204208
std	257.353842	0.486592	0.836071	NaN	NaN	14.526497	1.102743	0.806057	NaN	49.693429
min	1.000000	0.000000	1.000000	NaN	NaN	0.420000	0.000000	0.000000	NaN	0.000000
25%	223.500000	0.000000	2.000000	NaN	NaN	20.125000	0.000000	0.000000	NaN	7.910400
50%	446.000000	0.000000	3.000000	NaN	NaN	28.000000	0.000000	0.000000	NaN	14.454200
75%	668.500000	1.000000	3.000000	NaN	NaN	38.000000	1.000000	0.000000	NaN	31.000000
max	891.000000	1.000000	3.000000	NaN	NaN	80.000000	8.000000	6.000000	NaN	512.329200

Titanic Dataset Description

Typically we'd write some code to get a summary like this. With Pandas AI, however, we just need to write a prompt.

This will save you a ton of time if you're a beginner who wants to analyze some data but don't know how to write Python code.

Next, let's perform some exploratory data analysis with Pandas AI:

I'm asking it to give me the relationship between the "Survived" variable in the Titanic dataset, along with some other variables in the dataset:

```
smart_df.chat("Are there correlations between Survived and the following variables: Age, Sex, Ticket Fare. Format this output as a table.")
```

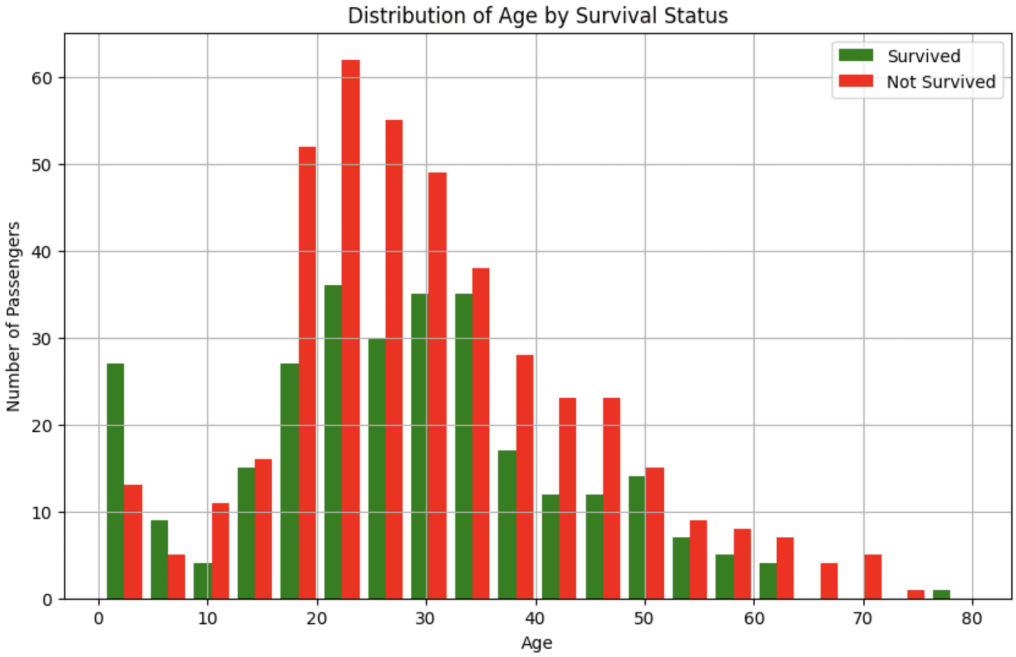
The above prompt should provide you with a correlation coefficient between "Survived" and the other variables in the dataset.

Next, let's ask Pandas AI to help us visualize the relationship between these variables:

1. Survived and Age

```
smart_df.chat("Can you visualize the relationship between the Survived and Age columns?")
```

The above prompt should give you a histogram that looks like this:



Titanic Dataset Age Distribution

This visual tells us that younger passengers were more likely to survive the crash.

2. Survived and Gender

```
smart_df.chat("Can you visualize the relationship between the Survived and Sex")
```

You should get a bar chart showcasing the relationship between “Survived” and “Gender.”

3. Survived and Fare

```
smart_df.chat("Can you visualize the relationship between the Survived and Fare")
```

The above prompt rendered a box plot, telling me that passengers who paid higher fare prices were more likely to survive the Titanic crash.

Note that LLMs are non-deterministic, which means that the output you’ll get might differ from mine. However, you will still get a response that will help you better understand the dataset.

Next, we can perform some data preprocessing with prompts like these:

Prompt Example 1

```
smart_df.chat("Analyze the quality of this dataset. Identify missing
```

```
values, outliers, and potential data issues that would need to be
addressed before we build a model to predict survival.")
```

Prompt Example 2

```
smart_df.chat("Let's drop the cabin column from the dataframe as it
has too many missing values.")
```

Prompt Example 3

```
smart_df.chat("Let's impute the Age column with the median value.")
```

If you'd like to go through all the preprocessing steps I used to clean this dataset with Pandas AI, you can find the complete prompts and code in my Kaggle notebook.

In less than 5 minutes, I was able to preprocess this dataset by handling missing values, encoding categorical variables, and creating new features. This was done without writing much Python code, which is especially helpful if you are new to programming.

How to Learn AI for Data Analytics: Next Steps

In my opinion, the main selling point of tools like Cursor and Pandas AI is that they allow you to analyze data and make code edits within your programming interface.

This is far better than having to copy and paste code out of your programming IDE into an interface like ChatGPT.

Additionally, as your codebase grows (i.e. if you have thousands of lines of code and over 10 datasets), it is incredibly useful to have an integrated AI tool that has all the context and can understand the connection between these code files.

If you're looking to learn AI for data analytics, here are some more tools that I've found helpful:

- GitHub Copilot: This tool is similar to Cursor. You can use it within your programming IDE to generate code suggestions, and it even has a chat interface you can interact with.
- Microsoft Copilot in Excel: This AI tool helps you automatically analyze data in your spreadsheets.
- Python in Excel: This is an extension that allows you to run Python code within Excel. While this isn't an AI tool, I've found it incredibly useful as it allows you to centralize your data analysis without having to switch between different applications.

Natassha Selvaraj is a self-taught data scientist with a passion for writing. Natassha writes

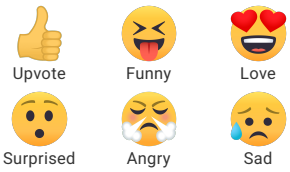
on everything data science-related, a true master of all data topics. You can connect with her on LinkedIn or check out her YouTube channel.

More On This Topic

- How I Would Learn Data Science in 2025 (If I Could Start Over)
- Top 10 High-Paying AI Skills to Learn in 2025
- How I Would Learn Python in 2025 (If I Could Start Over)
- Discover What's Ahead: Gartner Data & Analytics Summit 2025
- Data Science Salaries & Job Market Analysis: From 2024 to 2025
- These Are The Soft Skills You Will Need As A Data Scientist in 2025

What do you think?

4 Responses



<= Previous post Next post =>