GENERATIVE AI FOR

Biomechanists

Sinan Koparan

Sinan Koparan

PhD Candidate

Sports Data Science & AI

CSIRO Data61

Rugby Australia

NSWIS

PhD Research

Investigating how community club characteristics influence participation growth and retention, using ML to identify where sporting organisations can best allocate resources.

Applied Generative AI with NSWIS

Exploring how Large Language Models can automate content analysis in coach and athlete environments to support faster insight generation.

LLM Enthusiast

Passionate about developing deep technical understanding of LLMs and exploring how these models can drive real-world impact in sports and data science.

Agenda

01 **Models, Providers & Products**
Understanding the key distinctions

02 **Model Capabilities**
Text, multimodal, code, tool use & RAG

03 **Agentic Coding Tools**
AI-powered autonomous coding agents

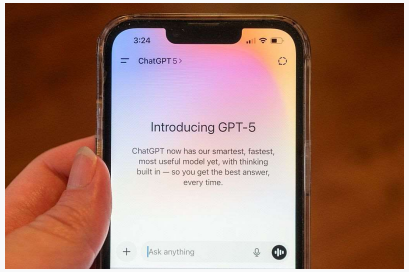
SECTION ONE

Models, Providers & Products

Models, Providers & Products

Model

The underlying system trained on data



Examples:
GPT-4, Claude Opus 4.5, Gemini 3 Pro

Provider

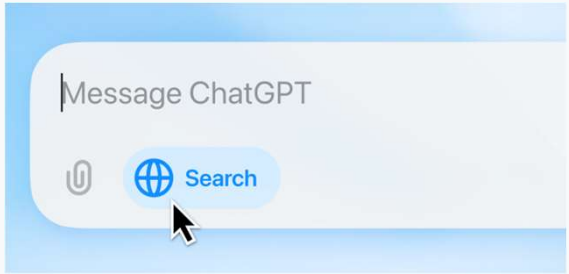
The company that builds and operates the model

OpenAI

Examples:
OpenAI, Anthropic, Google, Meta

Product

The user-facing application built on top of a model



Examples:
ChatGPT, Claude.ai, Microsoft Copilot

The Generative AI Landscape

Provider		Model		Product
OpenAI	→	**GPT-4o**	→	ChatGPT
Anthropic	→	**Claude Opus 4.5**	→	Claude.ai
Google	→	**Gemini 3 Pro**	→	Gemini App
Meta	→	**Llama 4**	→	Open Source (LMStudio)
OpenAI	→	**GPT Models**	→	Microsoft Copilot

Why the Distinction Matters

- The **same model** can behave differently depending on the product you access it through
- The **product** determines the interface, features, and often adds its own limitations
- Each model has different **strengths, weaknesses, and pricing**

> **Key Question:** When evaluating AI tools, always ask: which model powers this product?

SECTION TWO

02

Model Capabilities

Text Generation & Structured Outputs

Text Generation

The most familiar AI capability. Models can:

- Draft communications
- Summarise reports
- Translate content
- Answer questions from context

Structured Outputs

Instead of free-form text, models can respond with JSON.

You can define a schema of the fields you want the model to return.

```
{
  "name": "John Smith",
  "age": 34,
  "eye_colour": "blue"
}
```

Multimodal Understanding

Documents

Process PDFs up to 1,000 pages with native vision. It can read text, charts, tables, and diagrams.

Images

Analyse photos, screenshots, and diagrams; describe content and extract information

Video

Describe video content, identify moments, answer questions using visual and audio

Audio

Transcribe speech, understand spoken content, and identify individual speakers

Code Generation

Models can write, explain, and debug code across virtually any language. You can:

- Describe what you want in plain language and receive working code
- Paste existing code to explain, debug, or refactor it
- Convert code between languages
- Generate tests and documentation

Tasks that took hours (boilerplate, debugging, scaffolding) can be completed in minutes.

```
# Prompt: "Write a function that
# calculates compound interest"

def compound_interest(p, r, n, t):
    """Calculate compound
    interest."""
    amount = p * (1 + r/n) ** (n*t)
    return round(amount, 2)
```

Tool Use & Function Calling

Models can call external tools — databases, APIs, and software — and use results in their response.

Information Tools

Retrieve data without changing anything

- Web searches
- Database queries
- Document lookups

Low risk — read-only operations

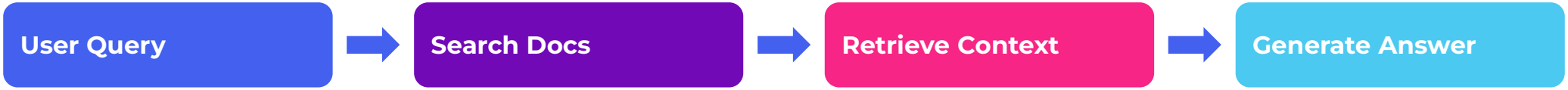
Action Tools

Do something in the real world

- Send emails
- Create calendar events
- Update database records
- Trigger workflows

Higher risk — requires safeguards

Retrieval-Augmented Generation (RAG)



- RAG allows models to search your actual documents and use relevant information in responses
- Grounds answers in your data rather than relying solely on training knowledge
- Users ask questions in plain language and receive answers with source references

Note: RAG relies on retrieving the right chunks — important details can be missed if not well-matched to the query.

SECTION THREE

03

Agentic Coding Tools

Agentic Coding Tools

Claude Code

by Anthropic

An agentic coding tool that reads your codebase, edits files, and runs commands autonomously.

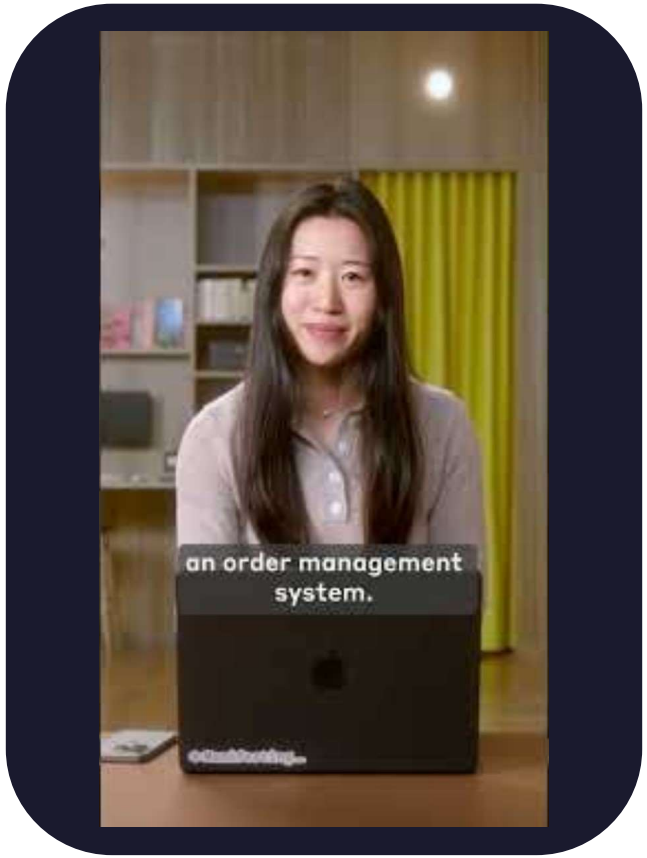
Available in your terminal, IDE, desktop app, and browser.

Terminal · VS Code · Desktop · Web

What Can It Do?

- Build **features** and fix **bugs**
- Automate **development tasks**
- Understand your **entire codebase**
- Work across **multiple files** and **tools**
- Run **commands** and integrate with **dev tools**
- Self-correct when something **fails**

Claude Code in Action

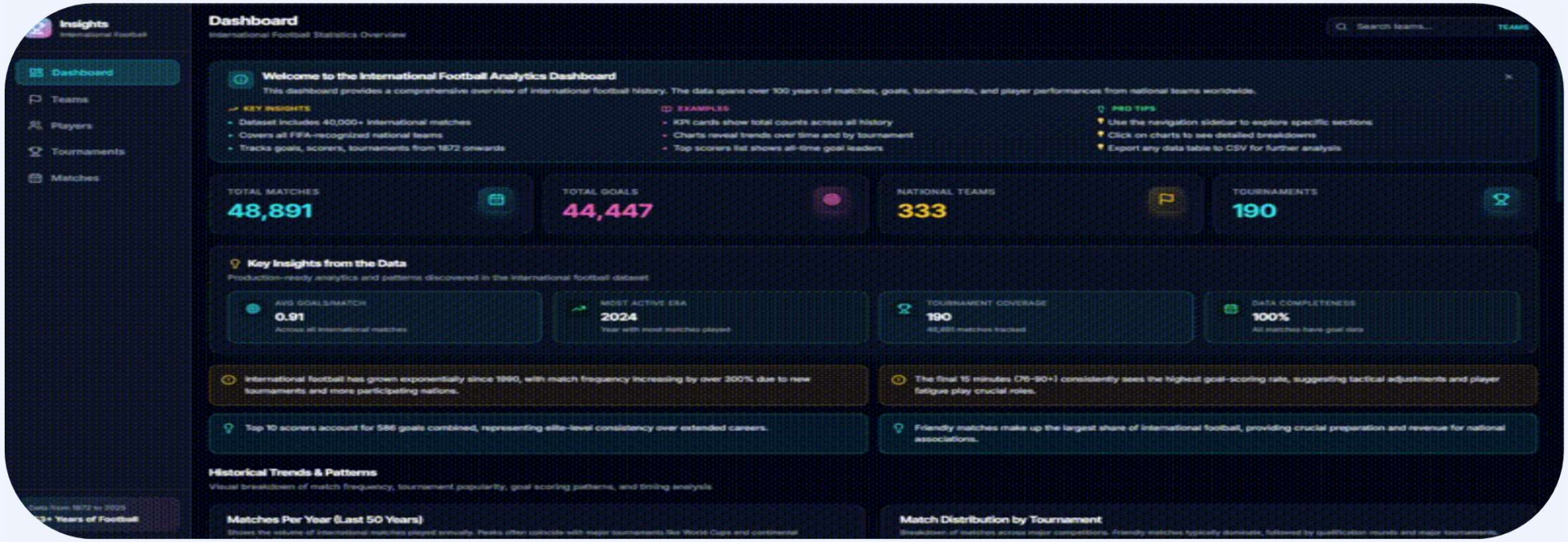


See how Claude Code autonomously builds, debugs, and deploys code in real time.
https://www.youtube.com/shorts/RCKo7OdjrCw

My Work with Claude Code

Using Claude Code to build sports analytics dashboards

Dashboarding



Building interactive sports dashboards with AI-assisted code generation

Use Cases in Sport

Admin Task Automation

Automate long-running admin tasks like grant finding, report generation, and data entry.

Machine Learning Exploration

Rapidly prototype ML models, test different algorithms, and iterate on feature engineering for sports datasets.

Performance Dashboards

Build interactive dashboards for athlete monitoring, match analysis, and performance tracking.

Data Pipeline Automation

Create automated pipelines for cleaning, transforming, and analysing sports data from multiple sources.

Useful AI Products

PPT

Claude in PowerPoint

AI assistant integrated directly into PowerPoint. Create, edit, and design presentations using natural language.

AI

Chatbots

ChatGPT, Claude.ai, Gemini — conversational AI interfaces for research, writing, analysis, and brainstorming.

XLS

Claude in Excel

AI-powered data analysis, formula generation, and spreadsheet automation directly within Excel.

NB

NotebookLM

Google's AI research assistant. Upload documents and have interactive conversations with your sources.

Key Takeaways

1. Understand the difference between models, providers, and products
2. AI capabilities extend far beyond text, models now understand documents, images, video, audio, and code
3. Tool use and RAG connect models to your real data and systems, making them genuinely useful at work
4. Practical AI products are available today

Thank You

Questions are welcome!

Scan to connect on LinkedIn

Visit my blog for more on Generative AI: sinankprn.com

LinkedIn
Sinan Koparan