AI for Finance

Professor: Gerard Sánchez Vidal
Email: Stream88rt@gmail.com
Office hours: by appointment

Course Description

This course is a practical, finance-oriented introduction to applied artificial intelligence — specifically, how large language models and AI agents are changing financial research, analysis and reporting. It is not a machine-learning course (predictive modelling is covered elsewhere in the program) and it is not a data-analysis course: the focus is on building AI systems that read, reason about and act on financial information. Students work with real, institutional-grade financial data and quantitative tools exposed through an API and a Model Context Protocol (MCP) server, and learn to orchestrate them with AI.

The course is built around real agents. Using a financial MCP that exposes a large library of tools over funds, equities, SEC filings, insider, congressional and institutional (13F) trades, macro data and earnings-call transcripts, students learn to connect an AI assistant to live markets, build research and due-diligence agents on a modern agent harness, manage what the model sees through context engineering, and assemble multi-agent pipelines like the ones behind professional fund, equity and macro reports — a writer agent, a regulatory critic and a deterministic numeric verifier. Throughout, the emphasis is on the responsible use of AI in a regulated industry: grounding answers in sources, evaluating outputs, controlling hallucination, defending agents against adversarial content, protecting confidential and inside information, and respecting both the relevant regulation and the line between an automated analysis and a signed investment recommendation.

Course objectives

This course aims to give you the skills to design, build and critically evaluate AI systems for finance — large language models and agents that read financial documents, query live financial data, and produce analysis a professional can defend. You will learn what today's AI can and cannot do in finance, how to ground it in real data and sources, and how to orchestrate tools and agents to automate genuine financial workflows: research, screening, due diligence and reporting.

Rather than building models or pipelines from scratch, you will leverage an existing institutional data-and-tools layer (the BQuant API and MCP) and concentrate on the AI layer on top of it. You will connect AI assistants to live data, build a financial-research agent on a modern agent harness, and study real multi-agent architectures, always with an eye on evaluation, cost, security, regulation and the advisory boundary.

Having successfully completed this course, you will:

(i) Understand what large language models and AI agents can and cannot do in finance, and how to prompt and ground them for reliable financial tasks.

(ii) Be able to apply AI to financial documents — extraction, classification and summarization of filings, news and earnings calls, including tables and scanned documents — and build retrieval-augmented (RAG) and context-engineering workflows that cite their sources.
(iii) Be able to connect an AI assistant to live financial data and tools through the Model Context Protocol (MCP), and ask markets questions in natural language.
(iv) Be capable of designing and building financial agents on a modern agent harness — the loop, tool use, context management and subagents — rather than hand-rolling it, and, where it pays off, composing multi-agent pipelines with a writer, a critic and a deterministic verifier.
(v) Understand the engineering and governance of AI in a regulated setting — evaluation, hallucination control, cost, traceability and the boundary between automated analysis and personalized investment advice.
(vi) Understand the specific risks of AI in finance — prompt injection and adversarial content, confidentiality, privacy and inside information, and the regulatory and model-governance context (EU AI Act, model risk, suitability) — and how to mitigate them.

Prerequisites

Basic Python and familiarity with notebooks are recommended, but you do not need to be a programmer. The course provides ready-made financial data and tools through an API and MCP, so the emphasis is on using and orchestrating AI for finance rather than on software engineering. Curiosity about markets and a willingness to experiment matter more than a coding background.

Methodology

You will learn through lectures, live agent demonstrations and hands-on case studies using real financial data, applying each concept to genuine market, fund and company problems.

Sessions are supported by slides and notebooks. Every student receives access to the financial data API and MCP server used in class (BQuant Finance) — the same lever used in the examples — so practice happens by building real agents on real institutional data and tools, not on toy datasets. The notebooks and the data access are the fundamental support for personal preparation before and after each session.

Much of the work is project-based. Students develop an intermediate deliverable and a final project, presented and defended in class, in which they build an AI agent for a financial use case of their choice.

Evaluation criteria

(A) Intermediate work – 20%

A mid-term deliverable in which students apply the tools seen so far to a financial task — for example, a RAG workflow that answers questions over a set of filings, a prompt-based extraction of structured data from earnings calls, or a small agent that queries the MCP to screen and compare funds or stocks. Submitted with the supporting notebook or agent configuration. Students may use AI tools, but must be able to explain and defend every step.

(B) Final project + presentation – 80%
An end-of-course project, presented and defended in class: a working AI agent or multi-agent system for a financial use case of the student's choice, built on the API and MCP — for instance, a research agent, a fund or equity due-diligence agent, a macro agent, a smart-money convergence agent, or an agentic portfolio that proposes candidate allocations from signals. Students must justify their design, show how the agent grounds its answers and controls hallucination, separate deterministic computation from AI judgment, and discuss security, the advisory boundary and the limits of their system. Assessment weighs how well the agent works, the soundness of the AI and governance decisions, and the clarity of the presentation. The final result must reflect the students' own understanding, even where AI tools were used.

Students are required to attend 80% of classes. Failing to do so without justified reason may lead to suspension from the program.

As with all courses taught at the UPF BSM, students who fail the course during regular evaluation will be allowed ONE re-take of the examination/evaluation. Students that pass any Retake exam should get a **5 by default as a final grade for the course.** If the course is again failed after the retake, students will have to register again for the course the following year.

Plagiarism is to use another's work and to present it as one's own without acknowledging the sources in the correct way. All essays, reports or projects handed in by a student must be original work completed by the student. By enrolling at any UPF BSM Master of Science and signing the "Honor Code," students acknowledge that they understand the schools' policy on plagiarism and certify that all course assignments will be their own work, except where indicated by correct referencing. Failing to do so may result in automatic expulsion from the program.

Calendar and Contents

Tentative topics (30 hours):

Generative AI for finance: foundations (5h):

-Large language models explained for finance: what they are, how they behave, capabilities and limits.
-Hallucination, context and reliability: why an LLM is not a database, and when to trust it.
-Prompting for financial tasks: getting consistent, defensible output.
-Adapting AI to a task: the ladder from prompting to retrieval (RAG) to tools and agents, and why fine-tuning is rarely the first answer in finance.
-Working with AI responsibly: the 4D framework (delegation, description, discernment, diligence) and the advisory boundary (informational, not financial advice).

AI over financial documents: context engineering, RAG and extraction (6h):

-Financial text tasks: extraction, classification and summarization of filings (10-K, 10-Q, 8-K, proxy), news and earnings-call transcripts.

-Tables, charts and scanned filings: extracting structured data from financial statements and image/PDF documents (multimodal).
-Structured outputs: turning unstructured filings and calls into usable, validated data.
-Retrieval-augmented generation (RAG) and context engineering: what to put in the context window; agentic search vs. semantic (vector) search; grounding answers, source citation and verifiable, auditable outputs.
-Use cases: reading a 10-K or a fund prospectus, summarizing an earnings call, automated due diligence.

AI agents and harnesses: MCP and the BQuant lever (8h):

-Workflow vs. agent: choosing the simplest design that works; when a deterministic chain beats an autonomous agent (reliability over sophistication).
-From chatbot to agent: the agent loop (model + tools + iteration); a raw API call vs. an agent harness.
-The harness in practice: tool execution, context management (compaction) and subagents — the Claude Agent SDK as a concrete example.
-Tool use and the Model Context Protocol (MCP): connecting Claude, Cursor or a notebook to live financial data and quantitative tools; the BQuant lever (a financial MCP with around 190 tools over funds, equities, SEC filings, smart money, macro and transcripts).
-Working with a large tool library: loading hundreds of tool definitions is costly, so scaling matters — tool search and deferred loading, code execution with MCP, and programmatic tool calling (which also keeps sensitive data out of the model's context).
-Building financial agents (real examples): a research agent (question → query → sourced answer), a screening agent, a fund/equity due-diligence agent, a macro agent, and a smart-money convergence agent.
-Packaging expertise as Agent Skills (SKILL.md): turning a due-diligence or compliance workflow into a reusable agent capability.

Composing agents, evaluation and production (6h):

-When to compose agents — and when not to: least autonomy and the judgment sandwich (deterministic core, the model only where judgment is irreducible); orchestrator-worker patterns often run as a single agent, not many.
-Multi-agent pipelines: the orchestrator-worker pattern (a lead agent, parallel subagents and a separate citation/verification pass), and the writer + critic + deterministic numeric verifier behind professional fund, equity and macro reports — largely a deterministic pipeline, with the agent only where judgment is needed.
-Determinism vs. judgment: deterministic computation for the numbers, AI for the narrative and the judgment — where the agent earns its place, and where it must not sign.
-Evaluating financial agents (evaluation first): start with a small set of representative queries; hallucination control, numeric verification and source citation; LLM-as-judge with a rubric; cost and latency.

-From prototype to production: the Claude Agent SDK and managed/hosted agents — sessions, observability, sandboxing and credential isolation (keeping data and secrets away from the model).
-End-to-end use cases on the BQuant lever: portfolio review, fund and equity research, and macro analysis — from question to auditable, published output.
-Capstone — agentic portfolios (the BQuant Agents case): a portfolio modelled not as an orchestrator calling tools but as a distributed field of beliefs over a manifold of assets, where orthogonal signal and analyst agents each hold a view and the edge is convergence across independent agents, not any single signal (rigorous validation; most signals do not survive). A portfolio-manager layer arbitrates which views to promote, or holds cash, with full traceable memos, proposing candidates and never signing the decision — the frontier case where composing agents genuinely earns it, framed geometrically (which regimes are adjacent, how far apart two allocations really are, a committee that updates beliefs rather than averaging votes).

AI risk, security and compliance in finance (5h):

-Adversarial security: prompt injection and data exfiltration through retrieved content (filings, news, web, email); treating retrieved content as data, not instructions.
-Defending financial agents: least-privilege tool access, human approval for actions with effect, sandboxing, and red-teaming your own agent (OWASP Top 10 for LLM applications).
-Confidentiality, privacy and inside information: what you can and cannot send to a model; client data and PII (GDPR), material non-public information and information barriers, data retention and zero-data-retention.
-Regulation and model governance: the EU AI Act (high-risk uses such as credit scoring and insurance), model risk management, MiFID II suitability, and auditability and record-keeping.
-Human-in-the-loop and accountability: explainability, the advisory boundary, and who is responsible for an AI-assisted decision.

Reading Materials

Anthropic Engineering — Building Effective Agents, Effective Context Engineering for AI Agents, and How We Built Our Multi-Agent Research System (anthropic.com/engineering).
Anthropic Engineering — Writing Effective Tools for Agents, Code Execution with MCP, and Effective Harnesses for Long-Running Agents (anthropic.com/engineering).
Claude Agent SDK, Model Context Protocol, and the Citations and tool-use features — documentation (code.claude.com, docs.claude.com, modelcontextprotocol.io).
Anthropic Academy — AI Fluency, Building with the Claude API, and the MCP courses (anthropic.com/learn).
OWASP Top 10 for Large Language Model Applications, and an overview of the EU AI Act and AI model-risk guidance, provided in class.
BQuant Finance — financial data API and MCP documentation and tool catalogue; course notebooks and real agent examples provided throughout the term.

Biography of the Professors

Gerard Sánchez Vidal – Associate Professor
Graduated in Business Administration and Management and Master's in International Finance from EADA. Founder of BQuant Finance, a quantitative finance and data platform, and AI lead at Singular Solving, where he works on agentic AI for financial institutions. Extensive experience as a professor and instructor, delivering Python, quantitative finance and AI training with a specialization in financial data and AI agents. Content creator with years of experience on social media (@Gsnchez). Professor in the M.Sc. Finance & Banking at UPF-BSM, and lectures at Universidad de Navarra and IEB.