

I'm starting a data science project. Help me create an AGENTS.md steering file.

Project name: covid data analysis

Domain: Healthcare

Goal: Produce machine-learning based models to help evaluate when patients should be admitted and subsequently discharge with covid-like symptoms

Tech stack: Python 3.12+, Pandas, scikit-learn

Constraints:

The CLAUDE.md should include:

Project Overview

- One-paragraph description of the project and its business context

Architecture

- Project directory structure (use the spec-driven layout above)
- Data flow: source → raw → cleaned → features → model → output

Coding Standards

- Language and version (e.g., Python 3.11+)
- Required libraries and pinned versions
- Style: PEP 8, type hints, docstrings (Google style)
- Notebook conventions: numbered, markdown headers, no dead code
- Logging: use Python logging, not print()

Data Conventions

- Date format: ISO 8601, timezone-aware (UTC unless specified)
- Missing values: use pd.NA, never fill silently
- Column naming: snake_case, descriptive, include units where relevant

- Every transformation must be documented in progress.md

Spec Workflow

- Before executing any phase, read the relevant spec in specs/
- After completing a phase, update progress.md with findings and decisions
- Do not proceed to the next phase until the current spec's acceptance criteria are met
- When EDA reveals something that changes the problem spec, update specs/01-problem.md

Testing

- Data validation tests in tests/test_data.py
- Pipeline reproducibility: random seeds, pinned dependencies
- Model evaluation: never evaluate on training data

Common Commands

- `python -m pytest tests/ -x -q` — run all tests
- `jupyter nbconvert --execute notebooks/*.ipynb` — execute notebooks
- `python src/pipeline.py` — run the full pipeline



I'm drafting the steering file now. I'm treating your request as "create an AGENTS.md file" and folding in the sections you listed for CLAUDE.md , since the structure is the important part.

Thought for 7s

Agents

Copy Edit Download

AGENTS.md

Project Overview

covid data analysis is a healthcare-focused data science project designed to build machine-learning models that support clinical decision-making for patients presenting with COVID-like symptoms. The project's business context is hospital and care operations: helping clinicians and care teams evaluate when a patient should be admitted and, later, when they may be appropriate for discharge. The goal is to improve consistency, transparency, and reproducibility in decision support while ensuring that all modeling work remains traceable, testable, and aligned with healthcare data governance expectations.

Architecture

Project Directory Structure

This project uses a spec-driven layout:

```
covid-data-analysis/  
├── AGENTS.md  
├── progress.md  
├── requirements.txt  
├── specs/  
│   ├── 01-problem.md  
│   ├── 02-data-ingestion.md  
│   ├── 03-data-cleaning.md  
│   ├── 04-feature-engineering.md  
│   ├── 05-model-development.md  
│   ├── 06-evaluation.md  
│   └── 07-reporting.md  
├── data/  
│   ├── raw/  
│   ├── cleaned/  
│   ├── features/  
│   └── output/  
├── notebooks/  
│   ├── 01_eda.ipynb  
│   ├── 02_feature_engineering.ipynb  
│   └── 03_modeling.ipynb  
├── src/  
│   ├── __init__.py  
│   ├── config.py  
│   ├── logging_config.py  
│   ├── pipeline.py  
│   └── data/
```

```
├── __init__.py
├── ingest.py
├── clean.py
├── validate.py
├── features/
│   ├── __init__.py
│   └── build_features.py
├── models/
│   ├── __init__.py
│   ├── train.py
│   ├── predict.py
│   └── evaluate.py
├── utils/
│   ├── __init__.py
│   └── io.py
└── tests/
    ├── __init__.py
    └── test_data.py
```

Data Flow

All data processing must follow this flow:

source → raw → cleaned → features → model → output

- **source:** Original source systems, extracts, or manually provided datasets
- **raw:** Immutable copies of ingested source data
- **cleaned:** Standardized and quality-checked datasets
- **features:** Modeling-ready tables with explicit feature definitions
- **model:** Trained machine-learning artifacts and evaluation outputs
- **output:** Reports, predictions, summaries, and other downstream deliverables

Coding Standards

- **Language:** Python 3.12+
- **Core libraries:**
 - pandas==2.2.2
 - scikit-learn==1.5.1

- `numpy==2.0.1`
- `pytest==8.3.2`
- `jupyter==1.0.0`
- **Style requirements:**
 - Follow PEP 8
 - Use type hints for public functions and methods
 - Use Google-style docstrings for modules, classes, and functions
- **Notebook conventions:**
 - Prefix notebooks with numeric ordering (for example, `01_eda.ipynb`)
 - Use clear markdown headers to separate sections
 - Remove dead code and unused cells before committing
 - Keep notebooks exploratory, but move reusable logic into `src/`
- **Logging:**
 - Use the Python `logging` module
 - Do not use `print()` for operational or pipeline output

Data Conventions

- **Dates and times:**
 - Use ISO 8601 format
 - All timestamps must be timezone-aware
 - Default timezone is UTC unless explicitly specified otherwise
- **Missing values:**
 - Use `pd.NA`
 - Never fill missing values silently; every imputation or exclusion must be explicit and documented
- **Column naming:**
 - Use `snake_case`
 - Prefer descriptive names over abbreviations
 - Include units in column names where relevant

- **Transformation tracking:**
 - Every transformation must be documented in `progress.md`

Spec Workflow

- Before executing any phase, read the relevant spec in `specs/`
- After completing a phase, update `progress.md` with findings, assumptions, and decisions
- Do not proceed to the next phase until the current spec's acceptance criteria are met
- When EDA reveals something that changes the problem spec, update `specs/01-problem.md`

Testing

- Add data validation tests in `tests/test_data.py`
- Ensure pipeline reproducibility through fixed random seeds and pinned dependencies
- Never evaluate model performance on training data
- Keep train, validation, and test boundaries explicit and documented

Common Commands

- `python -m pytest tests/ -x -q` — run all tests
- `jupyter nbconvert --execute notebooks/*.ipynb` — execute notebooks
- `python src/pipeline.py` — run the full pipeline

Additional Guidance for Agents

- Treat all healthcare-related data as sensitive and handle it with care
- Prefer transparent, interpretable baseline models before introducing more complex approaches
- Document admission and discharge target definitions explicitly before modeling

- Flag label leakage risks early, especially when using downstream clinical events or post-decision variables

Review all model documentation, feature eligibility rules, and evaluation

Created the steering file in the canvas as `AGENTS.md`.

One thing I normalized: your prompt says "create an `AGENTS.md` steering file," then says "The `CLAUDE.md` should include...". I used your requested sections and put them into `AGENTS.md`.

A couple of strong additions I included because they fit this project well:

- healthcare-specific guidance around sensitive data handling
- an explicit warning to watch for label leakage in admission/discharge modeling
- a recommendation to start with interpretable baseline models

If you want, I can also generate matching starter files next: `progress.md`, the `specs/` files, and a pinned `requirements.txt`.



Help me write `specs/01-problem.md` for this data science project.

Business context: Determine when to admit and release patients with covid symptoms

Stakeholders: doctors, hospital administrators

What we think the question is: determine when it is appropriate to admit patients experiencing covid symptoms. Similarly, determine the appropriate time to release those patients once they have sufficiently recovered

The spec should have these sections:

Business Context

- What is the business problem? Who cares and why?
- What decision will this analysis inform?

Analytical Questions

For each question:

- Statement: precise, unambiguous
- Type: descriptive / exploratory / inferential / predictive / prescriptive / causal / mechanistic
- Required data: what we need to answer it
- Success criteria: what does a good answer look like?
- Failure criteria: what would tell us the question is unanswerable?

Scope

- In scope: explicitly list what we will address
- Out of scope: explicitly list what we will NOT address
- Time range: what period does the analysis cover?
- Population: who/what are the units of analysis?

Assumptions

- List every assumption we're making (we'll validate these in EDA)

Constraints

- Regulatory, ethical, timeline, data access, team

Acceptance Criteria

- At least one question is classified and precise
- Success criteria are defined for each question
- Stakeholders have reviewed and agreed on scope
- Assumptions are documented and testable

Phase Gate

This spec must be reviewed before proceeding to Phase 2 (Data).

I'm drafting the problem spec with precise questions, explicit scope boundaries, and testable assumptions so it can act as the gate for the data phase.

Thought for a few seconds

01-problem.md

Business Context

Hospitals and clinical teams need a more consistent, evidence-based way to evaluate patients presenting with COVID symptoms and decide when inpatient admission is appropriate and when discharge is appropriate after recovery. This matters because unnecessary admission can consume scarce beds, staff time, and hospital resources, while delayed admission can increase patient risk and worsen outcomes. Similarly, discharging a patient too early can increase the risk of deterioration and readmission, while keeping a patient too long can reduce capacity and increase costs. The primary stakeholders are **doctors**, who need clinically useful decision support at the point of care, and **hospital administrators**, who need safe, efficient use of hospital capacity.

This analysis will inform two operational decisions:

1. Whether a patient presenting with COVID-like symptoms should be admitted to the hospital.
2. Whether an admitted patient with COVID-like symptoms has recovered sufficiently to be safely discharged.

Analytical Questions

Question 1

- **Statement:** Given information available at the time of presentation or initial evaluation, can we predict whether a patient with COVID-like symptoms is likely to require hospital admission?
- **Type:** Predictive
- **Required data:**
 - Patient encounter records for individuals presenting with COVID-like symptoms
 - Admission outcome labels
 - Demographics

- Presenting symptoms
- Vital signs at presentation
- Relevant laboratory and imaging summaries available at the decision point
- Comorbidity and medical history variables available at or before presentation
- Encounter timestamps
- **Success criteria:**
 - A model can be trained using only data available at the time of decision
 - Performance is meaningfully better than a naive baseline
 - Model outputs can be interpreted and reviewed by stakeholders
 - The prediction target is consistently defined across the dataset
- **Failure criteria:**
 - Admission labels are inconsistent, missing, or operationally ambiguous
 - Key predictors are unavailable or only recorded after the admission decision
 - Severe label leakage is unavoidable
 - Data quality or sample size is too poor to support reliable modeling

Question 2

- **Statement:** For an already admitted patient with COVID-like symptoms, can we predict whether the patient is appropriate for discharge within a defined near-term decision window using only information available up to that time?
- **Type:** Predictive
- **Required data:**
 - Inpatient stay records for admitted patients with COVID-like symptoms
 - Discharge timestamps and disposition labels
 - In-hospital vital signs and clinical observations over time
 - Laboratory results and treatment summaries timestamped during admission

- Indicators of deterioration, ICU transfer, readmission, or adverse post-discharge outcomes where available
- Daily or interval-based patient state snapshots that align to possible discharge decision points
- **Success criteria:**
 - The discharge target is clearly defined for a specific time horizon or decision window
 - A model can be built using only information available up to each discharge decision point
 - Performance is meaningfully better than a naive or majority-class baseline
 - Stakeholders judge the result to be clinically interpretable and operationally relevant
- **Failure criteria:**
 - “Appropriate discharge” cannot be operationalized from available data
 - Time-indexed inpatient data is missing, sparse, or not trustworthy
 - Important outcomes needed to validate discharge safety are unavailable
 - Leakage from future events cannot be eliminated

Question 3

- **Statement:** What clinical, demographic, and operational patterns are associated with admission decisions and discharge timing for patients with COVID-like symptoms?
- **Type:** Exploratory
- **Required data:**
 - All data listed for Questions 1 and 2
 - Additional metadata about encounter type, care setting, and length of stay where available
- **Success criteria:**
 - Patterns are reproducible, clearly documented, and stratified where appropriate

- Results identify candidate predictors, data quality issues, and subgroup differences for later modeling
- **Failure criteria:**
 - Data is too incomplete or inconsistent to support trustworthy summaries
 - Important fields required for stratification or interpretation are missing

Scope

In Scope

- Defining a precise prediction problem for admission among patients presenting with COVID-like symptoms
- Defining a precise prediction problem for discharge readiness among admitted patients with COVID-like symptoms
- Identifying the decision points and prediction targets for both admission and discharge
- Performing exploratory analysis to validate assumptions and understand data quality
- Building machine-learning models that estimate admission likelihood and discharge readiness
- Evaluating model performance using holdout data and clinically meaningful metrics
- Documenting feature eligibility rules to prevent leakage

Out of Scope

- Replacing physician judgment or automating final clinical decisions
- Recommending medications, treatment protocols, or individualized care plans
- Inferring causal effects of treatments or interventions unless explicitly re-scoped later
- Building a real-time production deployment in this phase
- Generalizing beyond the available hospital system and study population without validation

- Diagnosing COVID or other illnesses from raw imaging, notes, or sensor streams unless explicitly added later

Time Range

- Initial assumption: the analysis will cover the period for which reliable encounter, admission, inpatient, and discharge data are available for COVID-like symptom presentations.
- Exact start and end dates will be confirmed during Phase 2 after data inventory and quality review.

Population

- **Units of analysis:** patient encounters for the admission question and inpatient decision-time snapshots or hospital stays for the discharge question.
- **Population:** patients presenting to the hospital or health system with COVID-like symptoms during the study period.

Assumptions

- Patients with COVID-like symptoms can be identified consistently from the available data.
- Admission decisions are recorded clearly enough to create a reliable target label.
- Discharge decisions and timestamps are recorded clearly enough to define discharge readiness or near-term discharge outcomes.
- Timestamped data is accurate enough to separate what was known at decision time from what happened later.
- Sufficient pre-admission features exist to support an admission model.
- Sufficient longitudinal inpatient features exist to support a discharge model.
- The available sample size is large enough to support modeling and evaluation.
- Data quality is adequate across key variables such as vitals, labs, demographics, and outcomes.

- Stakeholders can agree on operational definitions of “appropriate admission” and “appropriate discharge.”
- Readmission, deterioration, ICU transfer, or other proxies for unsafe discharge are available or can be reasonably approximated.
- Historical clinical decisions are suitable as labels for an initial modeling phase, even if they are imperfect proxies for ideal decisions.
- Variation in hospital policy, staffing, or pandemic phase does not make the data unusably inconsistent across time, or such variation can be modeled or segmented.
- Ethical and regulatory review will permit use of the required healthcare data for this analysis.

Constraints

- **Regulatory:** Must comply with healthcare privacy, security, and institutional data governance requirements. Use of protected health information must follow applicable approval and access controls.
- **Ethical:** The project must avoid creating unsafe or biased recommendations, especially across clinically important subgroups. Outputs must be framed as decision support, not autonomous decision-making.
- **Timeline:** Scope may need to prioritize a baseline model and a small number of well-defined targets before broader analysis.
- **Data access:** Access to EHR-derived data, admission/discharge events, and longitudinal inpatient data may be limited, delayed, or partially incomplete.
- **Team:** The project depends on input from both technical contributors and clinical or operational stakeholders to validate definitions, scope, and interpretation.

Acceptance Criteria

-

Drafted specs/01-problem.md in the canvas.

A couple of choices I made to make it usable later in modeling:

- I split the project into two separate predictive questions: admission and discharge readiness
- I added an exploratory question because that usually belongs in Phase 1 and helps validate assumptions before modeling
- I kept the time range intentionally provisional, since the exact period should usually be confirmed during data inventory

One important note: your project goal says “help evaluate when patients should be admitted and subsequently discharge with covid-like symptoms.” For the discharge part, I translated that into a modeling question about whether a patient is appropriate for discharge within a defined decision window, because “when to discharge” needs an operational label to be testable.

The next strongest step would be for me to write `progress.md` and `specs/02-data-ingestion.md` so the workflow is connected end to end.



create progress.md

I’m creating `progress.md` as the project logbook so each phase can record findings, decisions, blockers, and changes to the specs.

Thought for a couple of seconds

Progress

Copy Edit Download

progress.md

Purpose

This file is the running record of project execution for **covid data analysis**. Every phase must update this document with what was done, what was discovered, what decisions were made, and what changes are required in the specs or downstream work. No transformation, assumption change, or modeling decision should exist only in code or notebooks without also being reflected here.

Project Metadata

- **Project:** covid data analysis
- **Domain:** Healthcare
- **Primary goal:** Produce machine-learning based models to help evaluate when patients should be admitted and when they may be appropriate for discharge when presenting with COVID-like symptoms
- **Tech stack:** Python 3.12+, Pandas, scikit-learn
- **Status:** Phase 1 in progress
- **Current phase:** Problem definition
- **Last updated:** YYYY-MM-DD

Workflow Rules

- Before starting a phase, read the relevant spec in `specs/`
- After completing a phase, update this file immediately
- Do not move to the next phase until the current phase acceptance criteria are met
- If EDA or data validation changes the problem definition, update `specs/01-problem.md`
- Every data transformation must be documented here
- Every modeling assumption, exclusion rule, and leakage concern must be documented here

Phase Status Summary

Phase	Spec	Status	Owner	Notes
-------	------	--------	-------	-------

1	specs/01-problem.md	In progress	TBD	Initial problem spec drafted
2	specs/02-data-ingestion.md	Not started	TBD	Awaiting phase gate review
3	specs/03-data-cleaning.md	Not started	TBD	Not started
4	specs/04-feature-engineering.md	Not started	TBD	Not started
5	specs/05-model-development.md	Not started	TBD	Not started
6	specs/06-evaluation.md	Not started	TBD	Not started
7	specs/07-reporting.md	Not started	TBD	Not started

Decision Log

Use this section to capture decisions that affect scope, data definitions, modeling, evaluation, or deployment assumptions.

Date	Phase	Decision	Reason	Impact
YYYY-MM-DD	1	Project split into two primary prediction problems: admission and discharge readiness	The business problem contains two distinct clinical decision points with different units of analysis and feature availability	Separate labels, datasets, and evaluation logic will be required
YYYY-MM-	1	Added exploratory	EDA is needed to validate	Phase 2 and Phase 3 must

DD	analysis as a formal analytical question	assumptions, data quality, and target feasibility before modeling	explicitly test assumptions and leakage risks
----	--	---	---

Change Log

Use this section to record changes to specs, code, notebooks, data definitions, or scope.

Date	Changed artifact	Change summary	Trigger
YYYY-MM-DD	AGENTS.md	Created steering file with architecture, standards, data conventions, and workflow rules	Project initialization
YYYY-MM-DD	specs/01-problem.md	Created initial problem definition spec with business context, analytical questions, scope, assumptions, constraints, and phase gate	Project initialization

Risks and Open Questions

Track anything unresolved that could block progress or invalidate the current plan.

Date	Category	Risk or question	Severity	Owner	Mitigation / next step
YYYY-MM-DD	Data	It is not yet confirmed how patients with COVID-like	High	TBD	Validate cohort definition during data

		symptoms will be identified in source data			ingestion and EDA
YYYY-MM-DD	Labels	“Appropriate discharge” is not yet operationally defined	High	TBD	Work with stakeholders to define discharge outcome and safe proxy labels
YYYY-MM-DD	Leakage	Future clinical events may leak into admission or discharge features	High	TBD	Audit feature timestamps before any modeling
YYYY-MM-DD	Access	Availability of longitudinal inpatient data is not yet confirmed	High	TBD	Confirm data sources and completeness in Phase 2

Assumption Validation Tracker

Each assumption from `specs/01-problem.md` must be tested during EDA or data validation.

Assumption	Validation method	Status	Findings
Patients with COVID-like symptoms can be identified consistently	Review diagnosis codes, symptom fields, encounter reasons, and cohort logic	Not started	TBD
Admission decisions are recorded clearly enough	Inspect encounter outcomes and admission flags	Not started	TBD

for labeling

Discharge decisions and timestamps are reliable	Compare discharge fields, timestamps, and disposition values	Not started	TBD
Timestamped data is accurate enough to prevent leakage	Audit event times relative to decision points	Not started	TBD
Sufficient pre-admission predictors exist	Profile completeness of candidate admission-time variables	Not started	TBD
Sufficient inpatient longitudinal data exists	Inspect repeated vitals, labs, and state changes over time	Not started	TBD
Sample size is adequate	Measure cohort size, class balance, and subgroup counts	Not started	TBD
Data quality is adequate across key variables	Run missingness, range, and consistency checks	Not started	TBD
Stakeholders can agree on operational definitions	Review with doctors and administrators	Not started	TBD
Unsafe-discharge proxies are available	Check for readmission, ICU transfer, mortality, or deterioration outcomes	Not started	TBD

Phase 1: Problem Definition

Planned Work

- Document the business context and decision points
- Define precise analytical questions
- Establish scope and out-of-scope boundaries

- Record assumptions and constraints
- Confirm phase gate before moving to data work

Findings

- The business problem naturally separates into two distinct prediction tasks:
 1. admission decision support at presentation
 2. discharge readiness decision support during or near end of stay
- These two questions likely require different units of analysis, feature sets, and labels
- A formal exploratory question is needed to validate assumptions before any modeling begins

Decisions

- Treat admission and discharge as separate modeling problems
- Require time-aware feature eligibility rules to reduce leakage risk
- Keep the study period provisional until data inventory confirms available historical coverage

Blockers

- Stakeholder review of scope has not yet occurred
- Operational definition of "appropriate discharge" remains unresolved
- Available data sources and label quality are not yet validated

Exit Check

-

Phase 2: Data Ingestion

Planned Work

- Inventory all data sources
- Confirm access, schema, and extract method

- Create immutable raw datasets in `data/raw/`
- Record source provenance and refresh cadence

Findings

- Not started

Decisions

- None yet

Data Transformations

- None yet

Blockers

- Not started

Exit Check

-
-

Phase 3: Data Cleaning

Planned Work

- Standardize formats and column names
- Resolve missing values explicitly
- Validate types, ranges, and consistency
- Produce cleaned datasets in `data/cleaned/`

Findings

- Not started

Decisions

- None yet

Data Transformations

- None yet

Blockers

- Not started

Exit Check

-
-

Phase 4: Feature Engineering

Planned Work

- Define candidate features for admission and discharge tasks
- Enforce feature time eligibility rules
- Build feature tables in `data/features/`
- Document exclusions and leakage controls

Findings

- Not started

Decisions

- None yet

Data Transformations

- None yet

Blockers

- Not started

Exit Check

-
-

Phase 5: Model Development

Planned Work

- Establish baselines
- Train candidate models
- Tune within defined scope
- Save reproducible model artifacts

Findings

- Not started

Decisions

- None yet

Data Transformations

- None yet

Blockers

- Not started

Exit Check

-
-

Phase 6: Evaluation

Planned Work

- Evaluate on holdout data
- Compare against baselines
- Assess subgroup behavior where possible
- Review error patterns and operational usefulness

Findings

- Not started

Decisions

- None yet

Blockers

- Not started

Exit Check

-
-

Phase 7: Reporting

Planned Work

- Summarize methods, findings, and limitations
- Produce final outputs in data/output/
- Document recommendations and next steps

Findings

- Not started

Decisions

- None yet

Blockers

- Not started

Exit Check

-
-

Transformation Log

Record every material transformation applied to data.

Date	Dataset	Step	Description	Reason	Code reference
YYYY-MM-DD	TBD	TBD	No transformations recorded yet	Project not yet in data phase	TBD

Modeling Log

Record experiments, evaluation choices, and important modeling decisions.

Date	Task	Model	Data split	Metrics	Result summary	Notes
YYYY-MM-DD	Admission / Discharge	TBD	TBD	TBD	No experiments recorded yet	TBD

Next Actions

1. Review `specs/01-problem.md` with doctors and hospital administrators.

Created `progress.md` in the canvas.

I structured it to work as a real project operating log, not just notes. It includes:

- phase-by-phase sections from problem definition through reporting
- a decision log, change log, and risk tracker
- an assumption validation tracker tied back to `specs/01-problem.md`
- a transformation log and modeling log so your workflow rules are enforceable

I also pre-populated the Phase 1 section with the decisions already implied by the problem spec, so the project starts with a usable audit trail.

