**1. Data & Integration Layer**

* Connectors to **CRMs** (Salesforce, HubSpot, etc.), **sales pipeline tools**, **marketing platforms**, **billing/subscription systems**
* ETL / ingestion pipelines (batch, streaming) with **scheduling, retry, error handling**
* Data cleaning, deduplication, normalization, validation
* Data mapping & schema management (so user can map fields)
* Staleness / freshness checks, data health alerts
* Audit logging for ingestion / transformation steps
* Support multiple data sources and unify into a canonical model

**2. Metrics & Core Computation Engine**

* Computation of key SaaS metrics (MRR, ARR, bookings, expansion, contraction, churn, net revenue retention, cohort metrics, LTV, CAC, etc.)
* Rolling/lagged metrics (e.g. trailing 3-month churn)
* Support for cohort analysis (by acquisition month, by segment, etc.)
* Time-based aggregation (daily, weekly, monthly, quarterly)
* Revenue attribution (by region, product line, customer segment, channel)
* Macro / external factor modifiers (allow manual adjustment / sensitivity inputs)

**3. Forecasting & Predictive Module**

* Multiple forecast methods:
  + Straight-line / trend / moving average
  + Historical growth model
  + Pipeline / stage-based / weighted pipeline
  + Cohort forecasting
  + Segmented forecasting (new vs renewal vs expansion)
  + AI / ML forecasting (time series, regression, etc.)
  + Ensemble / hybrid combining outputs
* Scenario modeling (best / base / worst)
* Forecast granularity (weekly, monthly, quarterly)
* Override & adjustment interface (allow manual corrections)
* Lag / activation delay modeling (to account for onboarding or ramp)
* Decay / staleness weighting for aged deals
* Forecast vs actual error tracking
* Confidence intervals, prediction bounds, sensitivity analysis
* Automated reforecasting / rolling forecasts

**4. Pipeline / Deal Management & Scoring**

* Import / sync deals with pipeline stages, probabilities, close dates, amounts
* Deal scoring / health indicators (based on historical patterns, signals)
* Mandatory fields and validation (e.g. close date, status, notes)
* Probability / win rate per stage
* Decay logic for deals stuck too long
* Alerts / flags on deals that deviate
* “What-if” adjustments (e.g. shifting close date) to simulate impact

**5. Dashboarding & Visualization**

* Modular dashboards per user / role (sales, finance, exec, etc.)
* Prebuilt templates for revenue forecast, churn analysis, pipeline health, cohort trends
* Drill-downs: from aggregate to segment to individual deal / customer
* Time series / trend views, bar / line charts, waterfall charts, funnel charts
* Forecast vs actual variance visuals
* Scenario comparison views
* Export / embed / share visual reports (PDF, CSV, image)
* Custom widget support or drag-and-drop dashboards

**6. Alerts, Insights & Notifications**

* Anomaly detection (unexpected churn spikes, revenue dips)
* Notifications / alerts for forecast deviations, pipeline shortfalls, at-risk deals
* Automated insights / recommendations (e.g. “renewal pipeline is weak, consider pushing incentives”)
* Natural language summaries or “insight cards”
* Scheduled reports / digests (daily, weekly) emailed or pushed to integrated channels (Slack, Teams, etc.)

**7. Workflow & Collaboration**

* Forecast review workflows (submit → review → approve)
* Comments / annotations on dashboards / forecasts / deals
* Versioning and snapshotting (save state of forecast at time X)
* Role-based access / permissions (who can edit, override, view)
* Audit trails (who changed what, when)

**8. Security, Reliability & Compliance**

* Encryption in transit & at rest
* Authentication (SSO, MFA)
* Row / column / object-level permissions
* Logging, monitoring, alerting on system health
* SLA / uptime guarantees
* Compliance with regional data laws (e.g. Kenya / GDPR / others)
* Backup & disaster recovery plans

**9. Admin & Configuration / Setup**

* Sales funnel / pipeline stage configuration
* Probability / weighting rules configuration
* Forecast model defaults & user override settings
* Time zone, currency, date format settings
* User management, roles, teams
* Onboarding wizard / guided setup
* Data import / seed data utilities
* Health check / data quality dashboards

**10. Learning, Adaptation & Model Improvement**

* Model retraining / adjustment over time based on forecast error
* Feedback loop from forecast vs actual to refine model parameters
* Ability to incorporate new features / signals (e.g. usage metrics, NPS, product engagement) into future forecasts
* Baseline benchmarking / comparison to industry or peer cohorts

**11. Integration & Ecosystem**

* APIs (read/write) to allow embedding or third-party apps to consume forecasts, dashboards
* Webhooks for alerting / events
* Integrations with communication tools (Slack, MS Teams), BI tools (Looker, PowerBI), spreadsheet tools
* Plug-ins / connectors for billing / subscription platforms, accounting systems
* Embeddable visual components (for putting dashboards in client portals, etc.)

**✅ Do’s & 🚫 Don’ts (Design / Product Strategy Guidance)**

Here are high-level rules and cautions when building or evolving jArvIs360, drawn from the articles + experience:

**Do’s**

1. **Begin with core, simple forecasting models** — don’t try to over-engineer AI early. Let users get value quickly, then layer complexity.
2. **Make it easy to override / adjust forecasts** — users want control and oversight over automated outputs.
3. **Enforce data hygiene and validation** — bad inputs lead to bad forecasts. Use guardrails.
4. **Provide explanation / transparency** — show how a forecast was derived (which deals, weights, assumptions).
5. **Support scenario planning** — let users see “what-if” outcomes under different growth, churn, backlog assumptions.
6. **Track forecast accuracy & learning** — make forecast error reporting a first class citizen.
7. **Include collaboration and review workflows** — forecasts often need manager review, notes, adjustment.
8. **Allow modular adoption** — some users may want only forecast + dashboards first, others full pipeline integration.
9. **Build extensibility** — allow custom metrics or signals (usage data, product engagement) to feed into forecast in future.
10. **Use short time windows first** — weekly or monthly forecasts tend to be more accurate and actionable.

**Don’ts**

1. **Don’t trust forecasts blindly** — always provide caveats, confidence intervals, and human review.
2. **Don’t lump revenue types blindly** — new sales, renewal, expansion behave differently; mixing them reduces accuracy.
3. **Don’t let stale / outdated deals dominate forecasts** — introduce decay or expiry logic.
4. **Don’t rely purely on sales rep gut calls** — blend them with data and signals.
5. **Don’t build for perfect accuracy** — instead build for consistent improvement and useful directionality.
6. **Don’t force monolithic adoption** — allow users to pick modules incrementally.
7. **Don’t neglect external variables** — macro trends, regulation, seasonality must be adjustable or simulated.
8. **Don’t obscure assumptions** — users must see which assumptions (growth rates, probabilities) feed forecasts.
9. **Don’t ignore user feedback** — build in mechanisms to review user edits / overrides and improve model over time.