SkyGeni – Sales Intelligence Challenge

Part 1: Problem Framing

1. What do you think is the real business problem here?

I believe the core business problem here isn't just a dropping win rate, it is a "Quantity vs Quality" disconnect. The healthy pipeline volume suggests that marketing is active, but the failure to convert it indicates either a drop in lead quality or a breakdown in closing the deal. Also, marketing is likely hitting volume targets with lower quality leads causing resource misallocation i.e., the sales reps might be wasting capacity on deals that appear healthy but will never convert which can be the root cause of the revenue decline.

2. What key questions should an AI system answer for the CRO?

Some key questions that AI system need to answer for the CRO are:

i. Is there a specific segment like region, industry, product or lead source driving the win rate decline?
ii. Are we losing deals early because of poor lead quality or losing late due to sales rep management?
iii. What are some early risk signals in current open deals that can predict a loss?
iv. Has our ICP shifted or is there a market shift?

3. What metrics matter most for diagnosing win rate issues?

Some of the most important metrics to diagnose win rate are:

i. **Loss Distribution by Stage (deal_stage):** This is an important metric for identifying the "where" of the process failure. The first step in the diagnosis is to know if a deal died at "Qualified" stage (Marketing issue) or at "Negotiation" (Sales issue) stage.
ii. **Win Rate by Source (lead_source):** In the current high volume-low win rate scenario, this metric helps to isolate where the "bad volume" is coming from. It tells you if the drop is caused by a specific marketing channel (e.g., Outbound) flooding the pipe with low-intent leads.
iii. **Avg Sales Cycle Length (sales_cycle_days):** Time kills deals. This metric helps correlate speed with success. If the average cycle for "Lost" deals is significantly longer than "Won" deals, it diagnoses a lingering pipeline where sales reps aren't cutting losses fast enough.

4. What assumptions are you making about the data or business?

i. The first assumption I made is about the dataset to ensure outcome integrity i.e., when a deal is actually lost, the sales rep marked “outcome” as “Lost” in the system promptly. This is critical because if reps are leaving dead deals as open, our win rate calculations will be mathematically wrong as we would be analyzing “zombie” data.

ii. Also, since the dataset includes global regions, I’ve assumed that “deal_amount” parameter has already been normalized to a single currency (e.g., USD) as otherwise our metrics will be mathematically meaningless.

iii. Lastly, I’m assuming that the deal stages in the dataset follows a ordinal hierarchy in the following sequence Qualified -> Demo -> Proposal -> Negotiation -> Closed

---

Part 2: Data Exploration & Insights

- ## Custom Metrics:

    i. **Win Rate by Region:**

    - **Definition:** This metric determines Segment Health. They reveal if the win rate drop is a systemic company-wide issue or isolated to a specific market. The win rate is calculated specifically for each geographic segment (NA, EU, APAC, India etc.) per quarter.
    - **Why I built it:** A global win rate averages out the bad performance. Thus, by isolating regions, we can prevent strong markets (like North America) from masking the failure in struggling markets (for e.g. APAC), allowing for targeted intervention.

    ii. **Pipeline Velocity (PV):**

    - **Definition:** This single truth metric balances volume against quality. It reveals if a healthy pipeline volume is actually generating revenue or just bloating the system with slow, low-probability deals.
      *PV = (Total Deals * Avg Deal Value * Win Rate) / Avg Sales Length Days*
    - **Why I built it:** Using only standard metrics like “Pipeline Volume” (count of deals) can be misleading because they don't account for speed or win rate. A pipeline can look full but be moving too slowly to hit revenue targets. So, to counter it, this metric reveals the true health of the engine.

2. Business Insights:

i. **Insight 1: Seasonality Factor Influencing Drop**

- **Observation:** While the drop in pipeline velocity for Q1 2024 looks alarming on paper, a closer look at the entire data i.e., the 5-quarter trend reveals this could be a recurring pattern. For instance, Q1 2023 showed a similar dip in pipeline velocity ($167k) before recovering in Q2 ($204k). This cycle aligns perfectly with the fiscal calendars of our dominant regions (North America & Europe), where Q4 represents a rush to spend remaining budget, followed inevitably by a slow Q1 where new budgets are still being approved and procurement teams are sluggish.
- **Why does it matter?** Context is everything. The CRO might currently be interpreting a seasonal shift as a performance crisis. If we treat this as a broken sales process, we might make rash decisions like slashing prices or changing sales strategies that could damage the business just as demand naturally rebounds in Q2. So, we are likely seeing a market pause not a product failure.
- **Action:** To validate this theory, pull historical data from last 3-4 years to overlay Q1 performances. If the drop appears every year, we can confirm this is in fact seasonality and shift expectations to Q2-Q4. And if the drop is new, we now know this is an anomaly, and we must urgently investigate external factors (e.g., a new competitor or pricing change.)

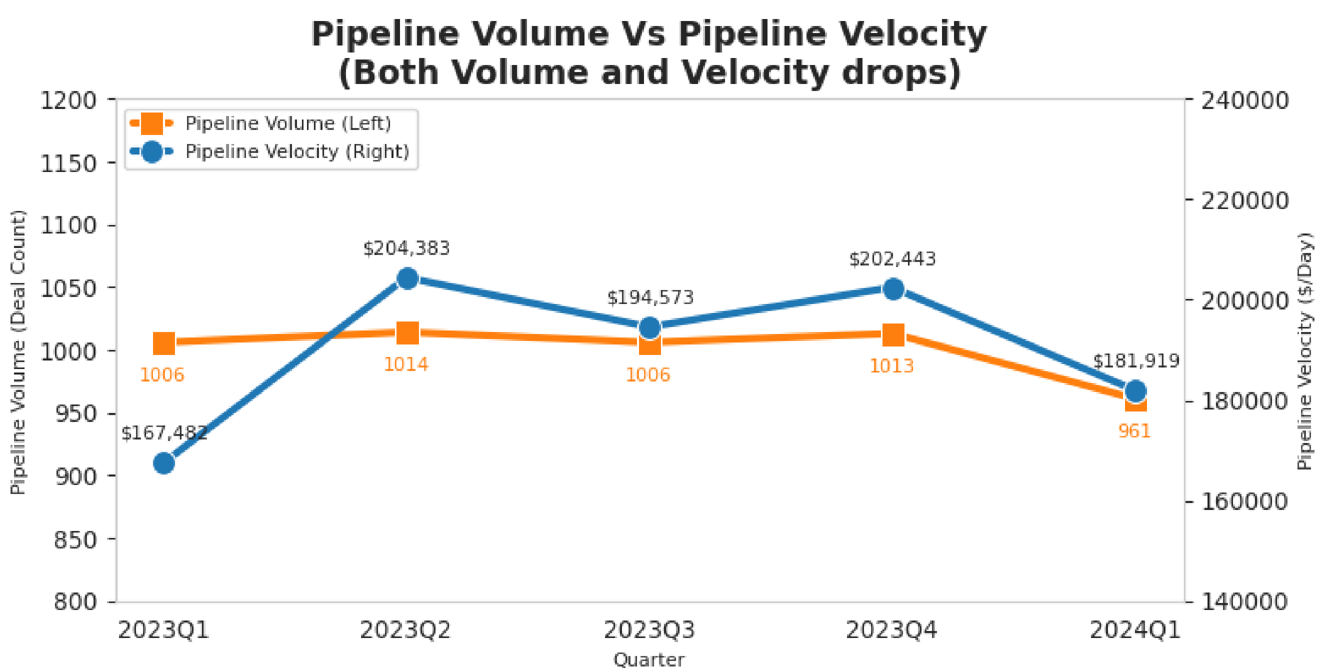

Fig. Pipeline Volume Vs Pipeline Velocity Over Last 5 Quarters

ii. **Insight 2: Win Rate Collapse in Western regions**

- **Observation:** The drop-in global win rate is almost entirely driven by two specific regions: North America and Europe. Their win rates plummeted, while APAC and other regions remained relatively stable.
- **Why does it matter?** It diagnoses the problem as localized, and not systemic. It shows that there is nothing wrong with the product globally, but there is something wrong with the execution, pricing, or competition specifically in these Western markets.
- **Action:** Launch a localized investigation in NA/EU. Are competitors undercutting price there? Did we hire a new batch of inexperienced reps in those regions? Or as discussed before, is it a new fiscal year reset? The fix must be regional, not global.

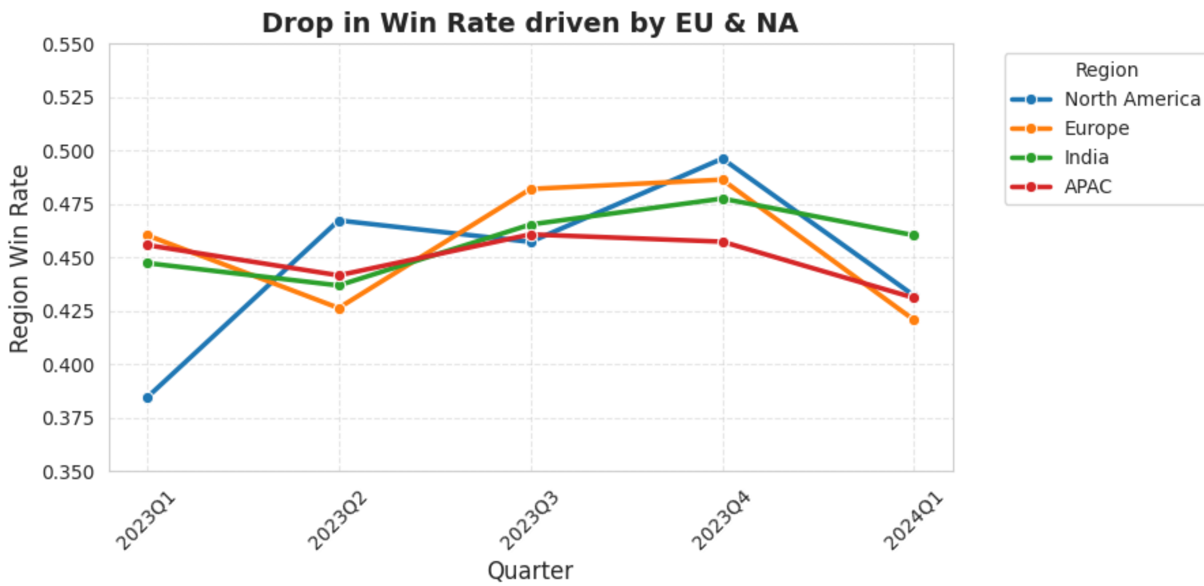

Fig. Drop in Win Rate by Region

iii. **Insight 3: Deals Lost at "Closed" Stage**

- **Observation:** Comparing the good quarter (Q4 of 2023) to the bad quarter (Q1 of 2024), we see a significant spike in deals dying at the final "Closed" stage.
- **Why does it matter?** This indicates "False Hope." Deals are passing all qualification stages (Demo, Proposal, Negotiation) only to fall apart at the signature line. This suggests a failure in closing skills, negotiation leverage, or final pricing strategy.
- **Action:** Develop new closing strategies and implement closing clinics for reps immediately. Review the discount approval process and late-stage competitive intelligence to understand why customers are walking away at the 11th hour.

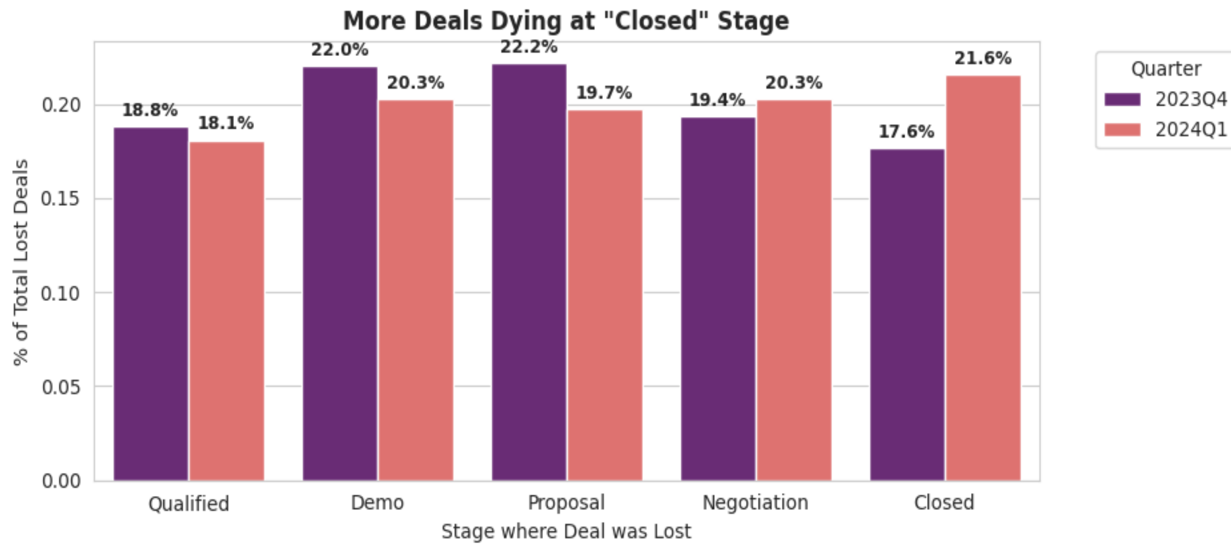

Fig. Percentage of Deals Lost at Each Stage

Part 3: Build a Decision Engine

Option A: Deal Risk Scoring

- **Problem Statement:** The CRO has 500+ open deals. They don't know which ones are healthy and which ones are “zombies” i.e., deals that look alive but are actually dead.
- **Goal:** To build a Deal Risk Score that assigns a probability of loss (0-100%) to every open deal in the pipeline.
- **Business Value:** This moves the sales team from a Reactive state ("Why did we lose that deal last month?") to Proactive state (“This deal has an 85% risk score, let's intervene today to save it.”)

Model Building:

- To address this problem of “zombie deals” cluttering the pipeline, I developed a Deal Risk Scoring Engine. The objective is simple - shift the sales team from a reactive post-mortem approach to a proactive intervention strategy. I selected Random Forest Classifier for this task because, unlike simple linear models, it excels at capturing non-linear relationships such as how a “high deal value” is generally positive but becomes a major risk factor when combined with specific regions or excessive stagnation. Furthermore, Random Forest offers high interpretability, allowing us to explain why a deal is risky, not just that it is risky.

- For data preparation, I engineered specific “risk signals” such as *deal_age_days* (to detect stagnation) and one-hot encoded categorical variables like *region* to capture geographical performance gaps. Crucially, I avoided a standard random train-test split. Instead, I implemented a time-based split, training the model strictly on last year's data (2023) and testing it on the current 2024 Q1 pipeline. This approach eliminates data leakage and accurately simulates a real-world production environment where the model must predict the future based solely on the past.

- The final deliverable is a Deal Risk Scoring Engine that assigns a probability of loss (0–100%) to every open opportunity available. Rather than overwhelming the CRO with raw data, the system generates a "Hit List" of high-value deals with critical risk scores (e.g., >75%). Each flagged deal includes its risk driving factors (e.g., Deal amount, length of the deal in days and stage score), empowering sales managers to immediately deploy resources to the deals that are most likely to be lost without intervention.

Part 4: Mini System Design

Concept: A risk monitoring layer that sits on top of a customer's CRM (Salesforce/HubSpot). It doesn't replace the CRM, it acts as a "Check Engine" for every deal.

1. High-Level Architecture

We will use a Microservices Architecture to ensure scalability across multiple customers.

- **Integrations Service (The Connector):** A dedicated service that uses OAuth to securely connect to the customer's CRM (Salesforce/HubSpot) and fetch incremental changes (deltas) every hour. It normalizes the messy data into a standard "SkyGeni Format."
- **The Intelligence Core (The Brain):** This is where our Python/Random Forest model lives. It runs in a containerized environment (Docker/Kubernetes). It spins up, scores the new data, and spins down to save costs.
- **Insight Store (The Memory):** A time-series database (like PostgreSQL or InfluxDB) that tracks the history of a deal's risk score. This allows us to show a "Risk Trend Line" (e.g., "This deal was safe last week, but spiked to High Risk today").
- **Notification Engine (The Voice):** A service that pushes alerts to where the users work: Slack, Microsoft Teams, or Email.

2. Data Flow (The "Life of a Data Point")

1. **Ingest:** At 1:00 AM, the system pulls the last 24 hours of deal updates from the customer's CRM.
2. **Clean & Enrich:** The system validates the data (e.g., checks for negative deal amounts) and adds "Enriched Features" (like days_in_stage or win_rate_for_this_region).
3. **Score:** The model calculates the **Loss Probability (0-100%)** and uses SHAP values to identify the **Top 3 Risk Factors** (e.g., "Stagnation", "Competitor Presence").
4. **Diff Check:** The system compares the *new* score to the *old* score.

    - *If Change > 15%:* Trigger an "Anomaly Alert."
    - *If Score > 75%:* Trigger a "Critical Risk Alert."
5. **Sync Back:** The system writes the Risk Score *back* into a custom field in the customer's CRM so they can see it without leaving Salesforce.

3. User Experience (UX) – Product Features

A. The "Risk Radar" Widget (Embedded in CRM)

- **What it looks like:** A small panel on the Salesforce Opportunity page.
- **Content:**
    - **Speedometer:** Shows the current Risk Score (e.g., 82% - Red Zone).
    - **"Why?" Bullet Points:**
        - *Stagnation:* Deal hasn't moved in 45 days.
        - *Historical Win Rate:* We only win 12% of deals in this Region.
- **Action Button:** "View Coaching Tips" (Suggests actions based on the risk factors).

B. The "Monday Morning" Manager Report (Email/Slack)

- **Subject:** "3 Deals Needing Attention This Week"
- **Content:** A prioritized list of high-value, high-risk deals.
    - *Deal A ($150k):* Risk spiked to 88%.
    - *Deal B ($90k):* Risk steady at 75%.
- **Value:** Saves the manager from digging through dashboards.

4. Operational Details (SaaS considerations)

- **Run Frequency: Hourly for Scoring, Weekly for Training.**
    - *Why Hourly?* Sales happens fast. If a rep updates a deal stage, they want to see the score change today, not tomorrow.
- **Model Retraining: Monthly (Automated).**
    - The system automatically retrains the model on the latest closed deals to learn new patterns (e.g., "Q4 seasonality").
- **Multi-Tenancy:** The system must keep Customer A's data strictly separate from Customer B's data (Logical Isolation).

5. Failure Cases (The "What Ifs")

- **The "Dirty Data" Defense:** If a customer's CRM has missing fields (e.g., no "Region"), the model falls back to a "Global Average" score rather than crashing. A warning is sent to the Admin.
- **API Rate Limits:** CRMs have limits on how much data you can pull. The system must implement "Exponential Backoff" (retrying slowly) to avoid getting blocked by Salesforce.

Part 5: Reflection

1. **Weakest Assumption:** In Part 1, I hypothesized the core issue was "Quantity vs. Quality"assuming marketing was flooding the pipeline with bad leads. My initial assumption was weak because it risked misdirecting the CRO to fix top-of-funnel marketing (lead generation), when the data actually points to a breakdown in bottom-of-funnel sales execution (closing). If we acted on my initial instinct, we would have solved the wrong problem.

2. **What Would Break in Production:** In a real CRM, sales operations teams frequently change field names (e.g., renaming "Stage 2: Demo" to "Stage 2: Discovery"). My current pipeline relies on exact string matching. If a field name changes even slightly, the One-Hot Encoding would fail silently or crash, effectively killing the deal risk scoring engine until an engineer manually fixes the code.

3. **What I Would Build Next (Given 1 Month):** Currently, the system is a "fire and forget" alert. I would build a mechanism to track User Outcomes. So, basically when we flag a deal as "High Risk", does the rep actually take action? And if they do, does it save the deal? We need to capture this data to re-train the model, turning it from a static predictor into a self-improving system.

4. **Area of Least Confidence:** I've selected a 75% risk score as the cutoff for alerts which is somewhat arbitrarily. I'm least confident that this is the mathematically optimal point. If it's too low, we spam the VP (False Positives) and if it's too high, we might miss critical risks (False Negatives). In a real deployment, I would need to run an A/B test or a precision-recall analysis to further tune this threshold properly.
