- Predictive Resource Optimization Platform (PROP) - KPMG Case Challenge Submission
  - 🏆 Team 7
  - 🎯 Project Overview
    - Platform Showcase
  - 🌟 Key Capabilities
  - 💼 Business Objectives and Optimization Targets
    - Value Extraction Coefficient (VEC)
      - Interpreting Financial Health (using VEC and Burn)
    - Resource Optimization
    - Key Performance Indicators (KPIs)
  - 🏗️ Architecture
    - Technology Stack
    - Database Approach
      - Data Model
      - Vacation Tracking Integration
  - 🛣️ Production Deployment and Dashboard Improvements Suggestions
    - Synthetic Elements and Production Replacements
  - 📝 Implementation Notes
    - Dashboard Enhancement Roadmap
  - 🚀 Getting Started
  - 📈 Performance Metrics Example
  - Appendix 1: AI Usage
  - Appendix 2: Predicting Optimal Allocation of Employee Resources
    - Implementation Strategy
  - Appendix 3: Modelling Assumptions
    - Critical Assumptions
    - Supporting Assumptions
  - Appendix 4: Key Metrics Derivation
    - Database Field References for Key Metrics
      - Budget at Completion (BAC)
      - Weighted Average Chargeout Rate (Estimation Aid)
      - Staffing Ratio Per Phase
      - Hours Required / Estimated Duration (Estimation Aids)
      - Days Elapsed
      - Percentage Schedule Elapsed
      - Actual Cost (AC)
      - Value Extraction Coefficient (VEC)
      - Budget Burn
      - Planned Value (PV)
      - Earned Value (EV)
      - Schedule Performance Index (SPI)
      - Weekly Benching Rate

        - Weekly Capacity Utilization Rate
        - Weekly Assignment Realization Rate (Internal Metric)
- Appendix 5: Mathematical Notation Conventions
    - Suffix Conventions
    - Core Variables and Factors
        - Financial Metrics
        - Adjustment Factors
        - Optimization Parameters
        - Utilization Metrics
- ETL (Extract, Transform, and Load) Pipeline
- Getting Started
    - Pipeline Components
        - 1. Data Extraction (excel_to_csv.py)
        - 2. Data Transformation (data_transformation.py)
        - 3. Database Loading (DML_writer.py)
        - 4. Unified Pipeline (pipeline.py)
    - ETL Workflow
    - Production Implementation
        - Salesforce Integration
        - Direct SQL Server Connection
        - Power Automate Integration
    - Making the Pipeline Robust
- Advanced Data Analysis
    - Analysis Components
        - 1. Data Fetcher (fetcher.py)
        - 2. Timesheet Analysis (analyze_time_entry.py)
        - 3. Anomaly Detection (anomaly_detection.py)
        - 4. Resource Allocation (allocation.py)
    - Analysis Workflow
    - Using the Analysis Tools
    - Next Steps

Predictive Resource Optimization Platform (PROP) - KPMG Case Challenge Submission

PYTHON 3.9 OR HIGHER | POWER BI DATA VISUALIZATION | SQL DATABASE QUERIES | EXCEL DATA ANALYSIS

🏆 Team 7

Team Member	GitHub	Linkedin	Background
Minh	@vibqetowi	hminh-software-eng	Software Engineering, previous experience in software development and project management
Carter	@carterj-c	cartercameronfina	Previously Mechanical Engineering now in Finance, previous experience in aerospace engineering
Casey	@cassius	casey-jussaume	Finance & Accounting, previous experience in financial modeling and research
Romero	@geekpapi	romero-p-faustin	Economics & Computer Science, previous experience in data analysis

🎯 Project Overview

Team 7 is pleased to present its submission for the KPMG case challenge: the **Predictive Resource Optimization Platform (PROP)** focusing on consulting assignment optimization. While our backgrounds are primarily in engineering, economics, project management, and finance rather than consulting, we've leveraged our technical expertise and project coordination experience to develop a solution that addresses resource allocation challenges common to professional services organizations. Our solution is a production-ready dashboard designed to implement enhanced earned value management (EVM) principles to optimize resource allocation across engagements.

This dashboard is specifically engineered to address critical business needs prevalent in consulting practices:

- Optimizing consultant utilization
- Minimizing consultant bench time
- Maintaining engagement schedules effectively
- Maximizing profitability through efficient resource allocation

Platform Showcase

Dashboard Preview *(A preview of the main dashboard interface)*

🌟 Key Capabilities

PROP delivers significant advantages across several key operational areas:

1. Financial Performance & Profitability

- **Enhanced Profitability Visibility:** Provides clear, real-time insight into engagement profitability.

- **Rapid Financial Analysis:** Delivers instant financial interpretations, includingvalue Extraction Coefficient (VEC), Burn Rate, and Schedule Performance Index (SPI).

2. Resource Management & Optimization

- **Optimized Resource Planning:** Supports improved work-life balance through realistic and effective capacity planning.
- **Intelligent Consultant Allocation:** Features an algorithm for optimizing consultant assignments.
- **Comprehensive Resource Tracking:** Monitors key resource utilization metrics, including bench time and overall capacity utilization.
- **Integrated Absence Management:** Incorporates vacation and time-off tracking to prevent resource allocation conflicts and ensure realistic capacity assessments.

3. Project Execution & Performance Management

- **Automated EVM Calculations:** Automatically calculates crucial Earned Value Management (EVM) metrics, such as SPI, Earned Value (EV), and Planned Value (PV).
- **Actionable Performance Insights:** Offers clear visualizations of engagement performance for informed decision-making.
- **Proactive Variance Management:** Enables effective monitoring of schedule and cost variances to maintain engagement control.
- **Improved Client Satisfaction:** Facilitates consistent adherence to project deadlines, enhancing client satisfaction. *(This outcome is directly linked to effective project execution)*

💼 Business Objectives and Optimization Targets

Our solution focuses on key business metrics that drive profitability and efficiency in consulting organizations: maximizing both budget utilization and maintaining target chargeout rates (captured by VEC), while ensuring timely delivery (SPI) and optimal resource usage (Benching Rate, Capacity Utilization Rate).

Value Extraction Coefficient (VEC)

Traditional project management metrics often fail to capture the dual financial objectives of consulting: utilizing the budget (not leaving money on the table) while maximizing the value realized from chargeout rates and resource mix. Our Value Extraction Coefficient (VEC) provides executives with a consolidated measure of *financial realization efficiency* relative to the budget for an engagement or phase.

The VEC is calculated as follows (shown for a specific phase ph):

$$\mathrm{VEC}_{ph} = \frac{1}{\mathrm{BAC}_{ph}} \times \sum_{t=1}^{n} [\mathrm{AC}_t \times d_t \times ea_c \times eo_c]$$

Where:

- BAC_{ph}: Budget At Completion for phase ph. The $\frac{1}{\mathrm{BAC}_{ph}}$ term normalizes the extracted value against the phase budget baseline. (VEC can also be calculated at the engagement level).
- $\sum_{t=1}^{n}$: Summation over all individual financial transactions t (typically timesheet entries) contributing to the phase's actual cost up to the point of measurement.
- AC_t: Actual Billed Amount (or Cost equivalent) for transaction t.
- d_t: Rate Efficiency Factor for transaction t, measuring how the actual chargeout rate used compares to the standard rate for the consultant c associated with transaction t.

$$d_t = \frac{\mathrm{Chargeout}_t}{\mathrm{StandardChargeout}_c}$$

- ea_c: External Adjustment Factor for consultant c (e.g., 1.0 for internal, 0.9 for external).

$$ea_c = 1 - 0.1 \times \mathrm{isExternal}_c$$

- eo_c: Onboarding/New Hire Adjustment Factor for consultant c (e.g., 1.0 for experienced, 0.8 for new).

$$eo_c = 1 - 0.2 \times \mathrm{isNew}_c$$

Interpreting Financial Health (using VEC and Burn)

Budget Burn for a phase is calculated as $\mathrm{Burn}_{ph} = \frac{\sum AC_t}{\mathrm{BAC}_{ph}}$. VEC and Burn provide a nuanced view of financial health:

	VEC > Burn (Premium Realization)	**VEC ≈ Burn (Standard Realization)**	**VEC < Burn (Value Dilution)**
Burn < 1	Excellent	On Track	Warning
Burn ≈ 1	Excellent	On Track	Warning
Burn > 1	Ambiguous	Critical	Critical

Interpretation Notes:

- **Excellent:** High efficiency, within or at budget. Strong performance.
- ctations.
- **Warning:** Inefficiency (dilution) detected, even if currently within budget. Potential future problems if trend continues.
- **Ambiguous:** Over budget, but spend appears efficient. Needs investigation: Is the high efficiency justifying the cost (e.g., scope change, premium value), or masking poor cost control?

- **Critical:** Over budget combined with standard or poor efficiency. Clear problem with cost control and/or value realization. Requires immediate attention.

Resource Optimization

Our solution addresses the core optimization challenge facing consulting organizations by recommending optimal consultant assignments. The goal is to dynamically adapt staffing while maintaining project health and efficiency.

Objective Function:

The optimization seeks the assignment matrix A (consultant c, engagement en, hours in week w) that maximizes overall value, penalized by switching costs and deviations from recent staffing patterns:

$$\max_{A} \sum_{en \in EN} [w_{en} \times r_{en} \times (\alpha \times \text{SPI}_{en} + \beta \times \text{VEC}_{en})] - \sum_{c,en,w} \text{pse} \times \text{is\_switch}_{c,en,w} - \sum_{en,l,pr,w} \text{Penalty}_{l,pr,en,w}$$

Where:

- A: The assignment matrix being optimized for the upcoming week(s) w.
- w_{en}: Strategic value weight of engagement en.
- r_{en}: Delivery risk coefficient for engagement en.
- α, β: Weighting coefficients balancing schedule (SPI) vs. value (VEC).
- $\text{SPI}_{en}, \text{VEC}_{en}$: Predicted Schedule Performance Index and Value Extraction Coefficient for engagement en *resulting from* the assignment A.
- pse: Phase Switching Efficiency factor (penalty for task switching).
- $\text{is\_switch}_{c,en,w}$: Binary variable: 1 if consultant c is newly assigned to engagement en in week w compared to week $w-1$.
- `Penalty_{l,pr,en,w}`: A penalty term discouraging significant deviations from the previous week's actual staffing mix for level l and practice pr on engagement en. Calculated as:

$$\text{Penalty}_{l,pr,en,w} = \gamma_{l,pr,en} \times \left| \frac{\sum_{c \in C_{l,pr}} \text{Hours}_{c,en,w}}{\sum_{c \in C} \text{Hours}_{c,en,w}} - \text{ActualRatio}_{l,pr,en,w-1} \right|$$

  - $\gamma_{l,pr,en}$: A configurable penalty weight. Higher values enforce stricter adherence to the previous week's ratio for that level/practice/engagement.
  - $\text{Hours}_{c,en,w}$: The *planned* hours for consultant c on engagement en in week w (part of the decision variable A).
  - $\text{ActualRatio}_{l,pr,en,w-1}$: The *actual* ratio of hours worked by consultants of level l and practice pr on engagement en during the *previous* week ($w-1$), calculated dynamically from

the `timesheets` data:

$$\text{ActualRatio}_{l,pr,en,w-1} = \frac{\sum_{c \in C_{l,pr}} \text{ActualHours}_{c,en,w-1}}{\sum_{c \in C} \text{ActualHours}_{c,en,w-1}}$$

Implementation Notes:

- The calculation of `ActualRatio_{l,pr,en,w-1}` from `timesheets` needs to be performed *before* running the optimization for week w.
- The penalty term adds complexity to the objective function, potentially requiring non-linear optimization solvers depending on the specific formulation and solver capabilities.
- Tuning the `gamma` parameters will be crucial to balance ratio adherence against other objectives. They could be global or specific (e.g., stored in `optimization_parameters` or linked to engagements).

Key Performance Indicators (KPIs)

Due to these business objectives, our solution focuses on the following core KPIs enabling proactive management:

1. **VEC (rolling)**: Value Extraction Coefficient tracking financial realization efficiency.
2. **Budget Burn (rolling)**: Ratio of Actual Cost to Budget ($\frac{\sum AC}{\text{BAC}}$) tracking cost control. *Used with VEC for financial health.*
3. **SPI (rolling)**: Schedule Performance Index tracking delivery progress against plan.
4. **Benching Rate (internal/external)**: Weekly rolling percentage of available capacity *not assigned* to engagements.
5. **Capacity Utilization Rate (internal/external)**: Weekly rolling ratio of *actual billable hours logged* versus standard capacity.

🏗️ Architecture

solution_architecture

Technology Stack

- **Data Analysis & Optimization**: Python (Pandas, NumPy, PuLP)
- **Data Storage**: SQL Server database (T-SQL), targeting Azure SQL compatibility.
- **Visualization**: Power BI
- **Integration Reference**: Sample pipeline script (`/scripts/pipeline/pipeline.py`) demonstrates ETL from Salesforce to SQL Server. *Requires adaptation for KPMG's specific environment.*

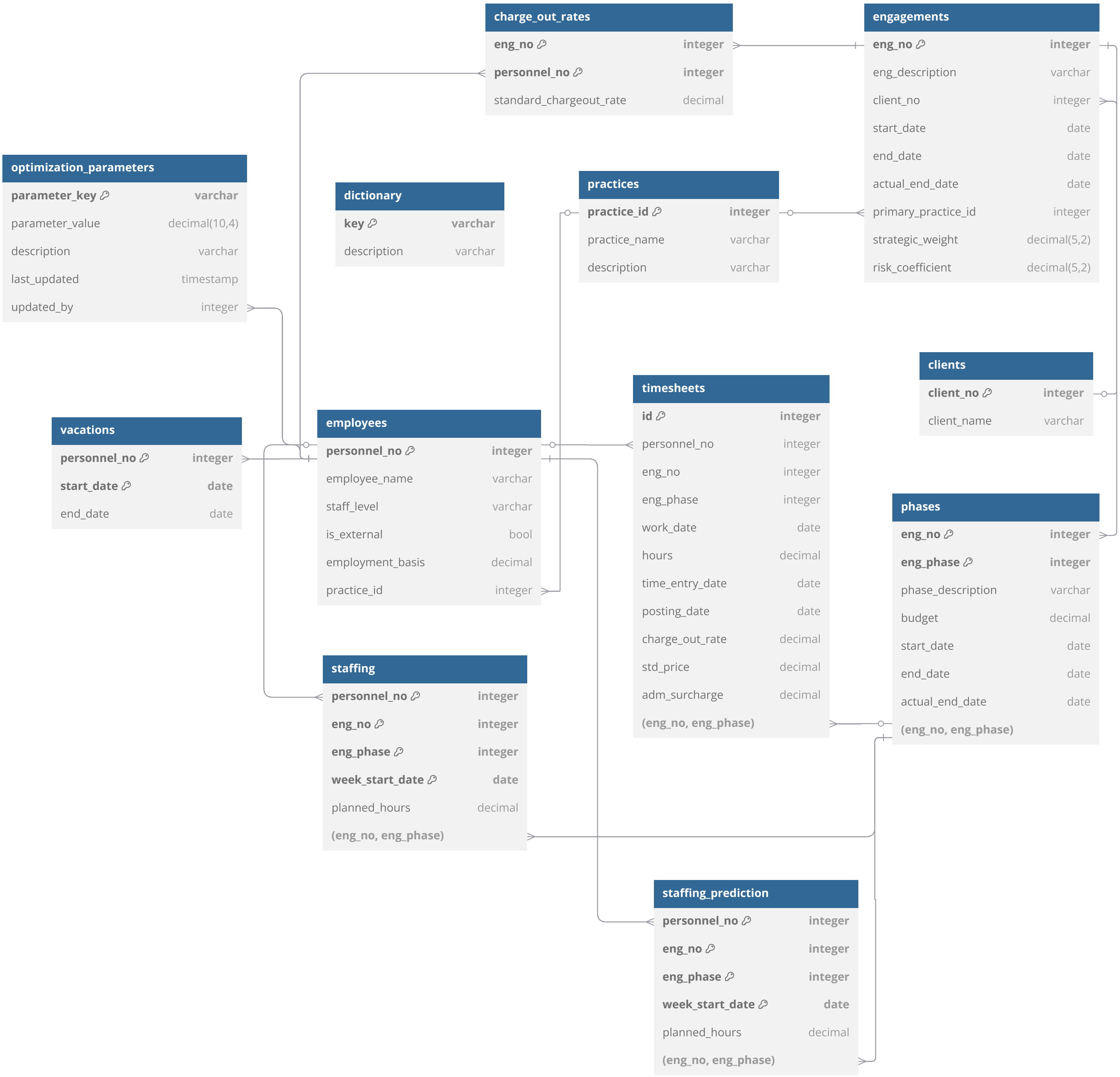
Database Approach

A dedicated database setup is essential:

1. **Advanced Analytics**: Enables calculations beyond Power BI's native capabilities.
2. **Complex Optimization**: Provides relational structure for the optimization engine.
3. **Historical Tracking**: Allows persistent storage for trend analysis.
4. **Scalability**: Supports future expansion.

Data Model

Interactive ERD: here. Static version:



charge_out_rates	
eng_no	**integer**
personnel_no	**integer**
standard_chargeout_rate	decimal

engagements	
eng_no	**integer**
eng_description	varchar
client_no	integer
start_date	date
end_date	date
actual_end_date	date
primary_practice_id	integer
strategic_weight	decimal(5,2)
risk_coefficient	decimal(5,2)

optimization_parameters	
parameter_key	**varchar**
parameter_value	decimal(10,4)
description	varchar
last_updated	timestamp
updated_by	integer

dictionary	
key	**varchar**
description	varchar

practices	
practice_id	**integer**
practice_name	varchar
description	varchar

clients	
client_no	**integer**
client_name	varchar

timesheets	
id	**integer**
personnel_no	integer
eng_no	integer
eng_phase	integer
work_date	date
hours	decimal
time_entry_date	date
posting_date	date
charge_out_rate	decimal
std_price	decimal
adm_surcharge	decimal
(eng_no, eng_phase)	

vacations	
personnel_no	**integer**
start_date	**date**
end_date	date

employees	
personnel_no	**integer**
employee_name	varchar
staff_level	varchar
is_external	bool
employment_basis	decimal
practice_id	integer

phases	
eng_no	**integer**
eng_phase	**integer**
phase_description	varchar
budget	decimal
start_date	date
end_date	date
actual_end_date	date
(eng_no, eng_phase)	

staffing	
personnel_no	**integer**
eng_no	**integer**
eng_phase	**integer**
week_start_date	**date**
planned_hours	decimal
(eng_no, eng_phase)	

staffing_prediction	
personnel_no	**integer**
eng_no	**integer**
eng_phase	**integer**
week_start_date	**date**
planned_hours	decimal
(eng_no, eng_phase)	

The ERD is implemented in the database, serving the Power BI layer.

Vacation Tracking Integration

Incorporates a dedicated `vacations` table:

- Stores employee time-off.
- Integrates with capacity calculations.
- Provides visual conflict indicators.
- Designed for integration with HR systems (uses synthetic data in PoC).
- Helps avoid resource conflicts due to unaccounted absences.

🛣️ Production Deployment and Dashboard Improvements Suggestions

PoC uses synthesized data; production requires actual KPMG data.

Synthetic Elements and Production Replacements

PoC Implementation	Recommended Production Data Source	Benefit of Production Data
Vacation Schedules: Generated synthetic	Integration with HR / PTO systems	Accurate capacity planning
Internal/External Status: Derived heuristically	Direct classification from HR / Vendor systems	Precise resource costing and VEC calculation
Employment Basis: Assumed 40-hour week	Actual contracted hours per resource	Accurate capacity for part-time/flexible staff
Engagement/Phase Timelines: Inferred from billing	Actual start/end dates from CRM/Project Mgmt	Precise SPI calculation and forecasting
Practice Areas: Defaulted ('SAP')	Actual department/practice assignments (HR/CRM)	Correct staffing mix validation & reporting
Standard Chargeout Rates: Assumed based on level	Official rate cards per level/practice	Accurate VEC Rate Efficiency (d_t) calculation
Strategic Weight (w_{en})/Risk (r_{en}): Constant	Defined via business rules/CRM flags	Optimization reflects true priorities/risk

📝 Implementation Notes

Recommended steps for production deployment:

1. **System Integration**: Establish robust API integration (e.g., Salesforce, HR).

2. **Replace Temporary Values:** View above table for reccomendations.
3. **Database Deployment**: Implement the SQL schema in KPMG's environment.
4. **Data Refresh Scheduling**: Set up regular (e.g., weekly) data refreshes.
5. **User Access Control**: Implement role-based security.
6. **Automated Alert System**: Develop alerts for KPI thresholds (e.g., low SPI).

Dashboard Enhancement Roadmap

Recommended enhancements:

1. **Historical Performance Integration**: Factor past consultant performance into optimization.
2. **Skills Taxonomy**: Implement detailed skills matching.
3. **Client Priority Weighting (w_{en})**: Allow dynamic setting of strategic weights.
4. **VEC Deeper Analytics**: Add views to analyze VEC trends by practice, client, etc.
5. **Chargeout Discount Monitoring**: Visualize rate realization (d_t) trends.
6. **Calibrated Switching Penalty (pse)**: Set realistic pse based on historical data.

These steps evolve the PoC into an essential operational tool.

🚀 Getting Started

```
# Clone repository
git clone [repository-url]

# Navigate to directory
cd [repository-directory]

# Install Python dependencies
pip install -r requirements.txt

# Configure database connection (update connection strings as needed)
# Ensure SQL Server instance is running and schema is deployed

# Open Power BI dashboard
open ./dashboards/project_performance.pbix
# Refresh data within Power BI
```

📈 Performance Metrics Example

For a sample engagement phase (Code ending 365, Phase 000010), metrics observed:

- Budget at completion (BAC - Phase): $2.3M
- SPI: [Value calculated]
- VEC: [Value calculated]
- Burn: [Value calculated]

- *Other relevant KPIs like Benching/Capacity Utilization can be shown here.*

Appendix 1: AI Usage

ChatGPT (free version) and GitHub Copilot were utilized during development for research assistance, conceptual brainstorming, content generation, and code development support.

Appendix 2: Predicting Optimal Allocation of Employee Resources

The optimization formula is solved algorithmically:

Implementation Strategy

1. **Data Preparation Phase (Scheduled)**
    - Calculate current engagement statuses (SPI, VEC, Burn).
    - Retrieve consultant availability (including vacations).
    - Determine required staffing ratios ($r_{l,pr,en}$) for active **engagements/phases**.
    - Retrieve strategic weights (w_{en}) and risk coefficients (r_{en}).
2. **Optimization Algorithm Selection**
    - Implemented using Mixed Integer Linear Programming (MILP) via Python PuLP.
    - Runs weekly to recommend assignments for the upcoming period.
    - Assignment matrix A (hours per consultant c, per engagement en, per week w).
3. **Constraint Implementation**
    - engagement schedule constraints (SPI_{en}).
    - engagement value constraints (VEC_{en} vs Burn_{en}).
    - Staffing ratio constraints ($r_{l,pr,en}$).
    - Consultant capacity constraints (≤ 40 hours, adjusted for vacation).
    - Vacation awareness.
    - Internal priority constraint.
    - engagement switching efficiency (pse) applied via objective function penalty.
4. **Enhanced Objective Function (Conceptual)**

```
# Pseudocode structure
model += pulp.lpSum([
    engagement_weights[en] * risk_coeffs[en] * (
        alpha * calculate_projected_SPI(en, assignments) +
        beta * calculate_projected_VEC(en, assignments)
    ) - sum(pse_factor * is_engagement_switch[c, en, w] for c in consultants for w
in weeks) # Penalty term
    for en in engagements
])
```

```
    # Internal consultant priority constraint logic...
    # Ensures internal consultants are assigned first up to capacity.
```

5. **Post-Optimization Processing & Feedback Loop**
    - Optimized assignments (A) written to a prediction table.
    - Dashboard visualizes recommendations.
    - Optional notifications for resource managers.
    - **Internal Model Tuning:** The deviation measured by the *Assignment Realization Rate* (Actual Billed vs. Assigned Hours, see Appendix 4) from the previous week can be used as an input signal or error metric (part of a loss function) to potentially adjust parameters (like consultant-specific efficiency factors or even α, β) or constraints in the optimization model for the ***next*** week's run. This creates a feedback mechanism to improve prediction accuracy over time. The specific algorithm for this adjustment (e.g., simple heuristic, gradient descent on a related metric) requires further development beyond the current PoC.
6. **Practical Considerations**
    - Configurable weights α, β.
    - Potential two-phase optimization for large scale (pre-filter then optimize).
    - Future stability constraints to minimize reassignments.
    - Future skills matching constraints.
    - Calibrate engagement switching factor (pse) based on data.

Appendix 3: Modelling Assumptions

This appendix outlines the key assumptions underpinning the analysis presented in this document. These assumptions were necessary due to data limitations or for simplification in the modelling process. They are categorized into critical and supporting assumptions.

Critical Assumptions

1. **Cost Performance Index (CPI) is set at 0.98.**
    - **Basis:** This value is derived from standard project management contingency practices (estimated at 10%) and an assumption regarding the operational efficiency of the KPMG workforce.
    - **Impact:** Enables preliminary project timeline estimations based on budget utilization. It is recommended that this assumption be replaced with empirically derived CPI values based on actual project performance data and VEC calculations in future operational applications.

Supporting Assumptions

1. **Phase Duration is Based on Staff Allocation.**

   - **Impact:** Project phase durations are estimated by dividing the total required effort (in hours) for the phase by the number of personnel allocated. This approach facilitates the projection of project completion dates based on planned resource assignments.

2. **Staffing Distribution is Consistent Across Phases.**

   - **Basis:** Analysis of sample engagement data indicates relatively consistent ratios in staff levels (e.g., Consultant, Manager) across different phases.
   - **Impact:** Allows for the calculation of reliable weighted average chargeout rates for engagements.

3. **Project Phases Progress Linearly.**

   - **Impact:** Projects are assumed to progress sequentially through defined phases without significant overlap. This simplification facilitates the calculation of Planned Value (PV) and project schedule projections.

4. **Project/Mandate Start Date is the First Logged Work Date.**

   - **Impact:** In the absence of explicitly defined start dates within the dataset, this definition provides a consistent reference point for timeline analysis and management.

5. **Client Identity is Determined by Client Number.**

   - **Basis:** Consistent with standard database management principles, the unique client number is used as the primary identifier for each client.
   - **Impact:** Ensures data integrity and enables consistent client tracking and aggregation across different engagements.

6. **Staff at Equivalent Levels are Interchangeable.**

   - **Impact:** Personnel within the same practice and at the same designated level are considered functionally interchangeable for resource allocation modelling purposes, assuming equivalent productivity (Productivity Substitution Effect, $pse = 1$). This simplifies calculations for cross-project assignments but represents an idealization; it should be refined with performance metrics where available.

7. **All Projects are Equally Important.**

   - **Basis:** Due to the absence of specific prioritization criteria in the available data, all projects within the analyzed dataset are assigned equal importance (project weight, w_p = constant).
   - **Impact:** Simplifies initial optimization modelling. Prioritization factors should be incorporated in subsequent analyses if available.

8. **All Projects Carry Uniform Risk.**

   - **Basis:** Project-specific risk data was not available for this analysis. Consequently, all projects are assumed to carry an equal level of risk (project risk, r_p = constant).

- **Impact:** Facilitates simplified preliminary analysis; risk differentiation should be included when data permits.

9. **Negative Hour Logs Offset Work on Other Projects for the Same Client.**
   - **Basis:** Observed data patterns suggest this correlation, aligning with common project management practices for budget adjustments.
   - **Impact:** Negative hours are interpreted as adjustments within the client's project portfolio. Further investigation is warranted to fully validate this interpretation.
10. **Time Reporting Behaviours Differ Between Internal and External Consultants.**
    - **Basis:** Analysis revealed significant variations in timesheet submission timeliness, with delays more pronounced among certain senior internal staff.
    - **Impact & Application:** A heuristic was applied for modelling: consultants with an average reporting lag < 3 days were provisionally classified as 'external' (excluding managers+). This allows for differential analysis but requires validation with actual employment status data.
11. **External Consultants Reduce Profitability by 10%.**
    - **Basis:** A preliminary estimate applied due to lack of specific cost data. This rate requires validation.
    - **Impact:** Adjusts the VEC calculation to account for potential differences in resource costs, informing staffing mix decisions.
12. **Chargeout Rates are Negotiated Per Engagement.**
    - **Basis:** Data analysis shows consultant rates are consistent within one engagement but vary across different engagements for the same individual.
    - **Impact:** Enables the tracking of engagement-specific chargeout rates, facilitating more accurate VEC calculations reflecting specific financial terms.

Appendix 4: Key Metrics Derivation

Database Field References for Key Metrics

This section provides the specific database field mappings for metrics described conceptually in the main document.

Budget at Completion (BAC)

- **Database Source**: `phases.budget` for engagement phase
- **Purpose**: Financial baseline for a phase.

Weighted Average Chargeout Rate (Estimation Aid)

- **Formula**: ‘$\text{Weighted Rate}_{ph} = \sum_{l=1}^{m}(\text{charge\_out\_rates.standard\_chargeout\_rate}_l \times \frac{\sum \text{staffing.planned\_hours}_l}{\sum \text{staffing.planned\_hours}})$‘
- **Purpose**: PoC estimation of average rate for a phase based on planned mix.

Staffing Ratio Per Phase

- **Formula**: $\text{StaffingRatio}_{l,ph} = \frac{\sum \text{timesheets.hours}_{l,ph}}{\sum \text{timesheets.hours}_{ph}}$

Hours Required / Estimated Duration (Estimation Aids)

- **Formulae**:
  - $\text{Hours}_{\text{required\_est},ph} = \frac{\text{phases.budget}_{ph}}{\text{Weighted Rate}_{ph}}$
  - ‘$\text{Duration}_{\text{est},ph} = \frac{\text{Hours}_{\text{required\_est},ph}}{\sum(\text{staff\_count}_l \times \text{daily\_hours})}$‘
- **Purpose**: PoC estimation of total effort/timeline for a phase.

Days Elapsed

- **Formula**: $\text{Days}_{\text{elapsed},ph} = \text{CURRENT\_DATE} - \text{phases.start\_date}_{ph}$
- **Purpose**: Measures time progression.

Percentage Schedule Elapsed

- **Formula**: $\text{Schedule\%}_{ph} = \frac{\text{Days Elapsed}_{ph}}{(\text{phases.end\_date}_{ph} - \text{phases.start\_date}_{ph})}$
- **Purpose**: Standardizes schedule progress measurement for PV calculation.

Actual Cost (AC)

- **Formula**: $\text{AC}_{\text{to date},ph} = \sum_{t=1}^{n}(\text{timesheets.std\_price}_t + \text{timesheets.adm\_surcharge}_t)$
- **Purpose**: Calculates actual expenditure to date.

Value Extraction Coefficient (VEC)

- **Formula**: $\text{VEC}_{ph} = \frac{1}{\text{phases.budget}_{ph}} \times \sum_{t=1}^{n}[(\text{timesheets.std\_price}_t + \text{timesheets.adm\_surcharge}_t) \times d_t \times ea_c \times eo_c]$

Where:

- $d_t = \frac{\text{timesheets.charge\_out\_rate}_t}{\text{charge\_out\_rates.standard\_chargeout\_rate}_c}$
- $ea_c = 1 - 0.1 \times \text{employees.is\_external}_c$
- $eo_c = 1 - 0.2 \times isNew_c$ (derived field, not in database)

Budget Burn

- **Formula**: $\text{Burn}_{ph} = \frac{\sum_t(\text{timesheets.std\_price}_t + \text{timesheets.adm\_surcharge}_t)}{\text{phases.budget}_{ph}}$

Planned Value (PV)

- **Formula**: $\text{PV}_{ph} = \text{phases.budget}_{ph} \times \text{Schedule\%}_{ph}$
- **Purpose**: Budgeted cost of work scheduled.

Earned Value (EV)

- **Formula (PoC Method)**: $\text{EV}_{ph} = \sum_t (\text{timesheets.std\_price}_t + \text{timesheets.adm\_surcharge}_t) \times 0.98$
- **Purpose**: Represents value of work completed with efficiency factor.

Schedule Performance Index (SPI)

- **Formula**: $\text{SPI}_{ph} = \frac{\text{EV}_{ph}}{\text{PV}_{ph}}$
- **Purpose**: Quantifies schedule efficiency.

Weekly Benching Rate

- **Formula**: ‘$\text{Benching Rate}_{c,w} = (1 - \frac{\sum_{en,ph} \text{staffing.planned\_hours}_{c,en,ph,w}}{\text{employees.employment\_basis}_c}) \times 100\%$‘
- **Definition**: Percentage of employee's contracted capacity not assigned to engagements.
- **Purpose**: KPI measuring unallocated bench time.

Weekly Capacity Utilization Rate

- **Formula**: ‘$\text{Capacity Utilization}_{c,w} = \frac{\sum_{en,ph} \text{timesheets.hours}_{c,en,ph,w}}{\text{employees.employment\_basis}_c} \times 100\%$‘
- **Definition**: Percentage of employee's contracted capacity utilized in billable work.
- **Purpose**: KPI measuring productive engagement.

Weekly Assignment Realization Rate (Internal Metric)

- **Formula**: ‘$\text{Assignment Realization Rate}_{c,w} = \frac{\sum_{en,ph} \text{timesheets.hours}_{c,en,ph,w}}{\sum_{en,ph} \text{staffing.planned\_hours}_{c,en,ph,w}}$‘
- **Definition**: Ratio of actual hours to assigned hours.
- **Purpose**: Internal metric for planning accuracy assessment.

Appendix 5: Mathematical Notation Conventions

This appendix documents the standardized notation used in formulas throughout this document to ensure consistency and clarity.

Suffix Conventions

All variables in formulas use consistent suffixes to denote the entity they refer to:

Suffix	Entity	Example	Description
`c`	Consultant/Employee	ea_c (External Adjustment)	Refers to a specific consultant/employee
`t`	Transaction/Timesheet	AC_t (Actual Cost)	Refers to a specific timesheet entry
`ph`	Phase	VEC_{ph} (VEC of a phase)	Refers to a specific engagement phase
`en`	Engagement	SPI_{en} (SPI of engagement)	Refers to a specific engagement
`l`	Staff Level	$StaffingRatio_{l,ph}$	Refers to a specific staff level (e.g., Manager)
`pr`	Practice	$C_{l,pr}$ (Consultants in practice)	Refers to a practice area (e.g., SAP)
`w`	Week	$Hours_{c,en,w}$ (Weekly hours)	Refers to a specific week (time period)

Core Variables and Factors

Financial Metrics

Variable	Definition	Formula/Source
BAC_{ph}	Budget at Completion for phase	`phases.budget`
AC_t	Actual Cost for transaction	`timesheets.std_price + timesheets.adm_surcharge`
PV_{ph}	Planned Value for phase	$BAC_{ph} \times Schedule\%_{ph}$
EV_{ph}	Earned Value for phase	$\sum_t AC_t \times CPI_{proxy}$ (PoC method)
SPI_{ph}	Schedule Performance Index	EV_{ph}/PV_{ph}
VEC_{ph}	Value Extraction Coefficient	$\frac{1}{BAC_{ph}} \times \sum_t [AC_t \times d_t \times ea_c \times eo_c]$
$Burn_{ph}$	Budget Burn Rate	$\frac{\sum_t AC_t}{BAC_{ph}}$

Adjustment Factors

Factor	Definition	Formula	Purpose
d_t	Rate Efficiency Factor	$\frac{timesheets.charge\_out\_rate_t}{charge\_out\_rates.standard\_chargeout\_rate_c}$	Measures charge-out rate efficiency
ea_c	External Adjustment Factor	$1 - 0.1 \times employees.is\_external_c$	Accounts for external consultants' impact on profitability
eo_c	Onboarding/New Hire Adjustment Factor	$1 - 0.2 \times isNew_c$	Accounts for newer consultants' learning curve
pse	Phase Switching Efficiency	Typically 0.9 (configurable)	Productivity loss factor when switching engagements

Optimization Parameters

Parameter	Definition	Range/Default	Source
α	SPI weight in optimization	0.5 (default)	`optimization_parameters` table
β	VEC weight in optimization	0.5 (default)	`optimization_parameters` table
w_{en}	Strategic weight of engagement	Default: 1.0	`engagements.strategic_weight`
r_{en}	Risk coefficient of engagement	Default: 1.0	`engagements.risk_coefficient`
$r_{l,pr,en}$	Required staffing ratio	Calculated from historical data	Derived from timesheet data

Utilization Metrics

Metric	Definition	Formula
Benching Rate	Percentage of capacity not assigned	$(1 - \frac{\sum_{en,ph} staffing.planned\_hours_{c,en,ph,w}}{employees.employment\_basis_c}) \times 100\%$
Capacity Utilization Rate	Percentage of capacity utilized in billable work	$\frac{\sum_{en,ph} timesheets.hours_{c,en,ph,w}}{employees.employment\_basis_c} \times 100\%$
Assignment Realization Rate	Ratio of actual hours to assigned hours	$\frac{\sum_{en,ph} timesheets.hours_{c,en,ph,w}}{\sum_{en,ph} staffing.planned\_hours_{c,en,ph,w}}$

ETL (Extract, Transform, and Load) Pipeline

This pipeline is a proof of concept (POC) for automating data extraction, transformation, and loading into a SQL Server database for the KPMG Data Challenge project. In this POC, we're working with CSV and Excel files, but the production implementation would connect directly to Salesforce

Getting Started

1. Ensure all dependencies are installed: `pip install -r requirements.txt`
2. Configure database connection in `.env` file
3. Run the unified pipeline script:

   ```
   python pipeline.py --excel-file <path_to_excel> --output-dir <path_to_output_dir>
   ```

4. Or run the pipeline components individually:

   ```
   python excel_to_csv.py <excel_file> --output-dir /path/to/output
   python data_transformation.py
   python DML_writer.py
   ```

5. Execute the generated DML.sql file in SQL Server

Pipeline Components

1. Data Extraction (`excel_to_csv.py`)

- **excel_to_csv.py**: Utility for converting Excel files to CSV format, useful for initial data ingestion from Excel sources.
- **Missing Component**: A direct Salesforce connector that would replace the Excel/CSV ingestion in the production implementation.

2. Data Transformation (`data_transformation.py`)

- Processes raw data files into standardized tables matching the database schema
- Handles data cleaning, type conversion, and foreign key validation
- Creates proper relationships between entities (employees, clients, engagements, etc.)
- Outputs transformed CSV files ready for database loading

3. Database Loading (`DML_writer.py`)

- Generates SQL Server T-SQL merge statements (upsert operations) from transformed data
- Handles batching for large datasets
- Manages proper insertion order for foreign key constraints
- Creates a single DML file that can be executed in SQL Server

4. Unified Pipeline (`pipeline.py`)

- Orchestrates the entire ETL process by calling each component in sequence
- Handles errors and ensures proper data flow between components
- Provides a single entry point for executing the complete pipeline

ETL Workflow

1. **Data Extraction**: Raw data is retrieved from Excel files and converted to CSV
2. **Transformation**: Data is cleaned, validated, and transformed to match database schema
3. **SQL Generation**: T-SQL statements are generated for database loading
4. **Database Update**: SQL scripts are executed to update the database

Production Implementation

For a production environment, this pipeline would be modified in the following ways:

Salesforce Integration

- Replace CSV input with direct Salesforce API integration
- Use the Salesforce Bulk API for large dataset extraction
- Implement OAuth 2.0 authentication for secure Salesforce access
- Schedule regular syncs via cron jobs or schedulers

Direct SQL Server Connection

- Eliminate the DML file generation step
- Use pyodbc or SQLAlchemy to execute SQL statements directly
- Implement transaction management for atomic operations
- Add comprehensive logging and error handling

Power Automate Integration

Advanced analyses could be triggered via Microsoft Power Automate:

- Create Power Automate flows triggered by database updates
- Connect Power BI reports to automatically refresh
- Send notifications when key metrics change
- Trigger downstream processes based on data changes

Making the Pipeline Robust

The current pipeline is a proof of concept. To make it production-ready and robust:

1. **Error Handling & Logging**
    - Implement comprehensive error handling at each stage
    - Set up centralized logging with different verbosity levels
    - Add monitoring for pipeline failures with alerts

2. **Validation & Testing**
    - Add data validation checks between each stage
    - Implement unit and integration tests for all components
    - Create test datasets for regression testing

3. **Recovery & Resilience**
    - Add checkpointing to allow restart from failure points
    - Implement retry logic for transient failures
    - Create backup/rollback mechanisms for critical operations

4. **Security & Compliance**
    - Implement proper authentication and authorization
    - Add audit logging for all data access and modifications
    - Ensure compliance with data governance policies

5. **Deployment & Automation**
    - Set up CI/CD for automated testing and deployment
    - Containerize the pipeline for consistent deployment
    - Implement scheduling for regular automated runs

Advanced Data Analysis

Analysis Components

1. Data Fetcher (`fetcher.py`)

- Provides a flexible data access layer for advanced analysis scripts
- Automatically attempts to connect to the SQL Server database first
- Falls back to CSV files if database connection fails
- Offers standardized data access methods for consistent analysis

2. Timesheet Analysis (`analyze_time_entry.py`)

- Analyzes employee time entry patterns and delays
- Calculates average delay between work date and entry date
- Identifies employees with the longest and shortest entry delays
- Detects unusual time entry behavior (e.g., entries made before work date)

3. Anomaly Detection (`anomaly_detection.py`)

- Uses machine learning (Isolation Forest) to detect unusual patterns in timesheet data
- Identifies anomalies in hours logged and time entry delays
- Generates summary statistics and visualizations
- Highlights employees with significant anomaly percentages

4. Resource Allocation (`allocation.py`)

- Calculates project performance metrics using Earned Value Management (EVM)
- Analyzes project completion rates against schedules
- Computes Cost Performance Index (CPI) and Schedule Performance Index (SPI)
- Provides insights into project health and resource allocation efficiency

Analysis Workflow

1. **Data Retrieval**: The `fetcher.py` utility retrieves data from the database or CSV files
2. **Data Processing**: Analysis scripts transform and clean the data for specific analyses
3. **Analysis Execution**: Specialized algorithms are applied to detect patterns or anomalies
4. **Results Generation**: Analysis results are displayed or saved for further reporting

Using the Analysis Tools

To run the advanced analysis tools:

```
# For timesheet entry analysis
python advanced_data_analysis/analyze_time_entry.py

# For anomaly detection
python advanced_data_analysis/anomaly_detection.py

# For resource allocation analysis
python advanced_data_analysis/allocation.py
```

Each script will automatically attempt to connect to the database, fall back to CSV files if needed, and generate the appropriate analysis output.

Next Steps

- Integrate analysis scripts with Power BI for interactive dashboards
- Create scheduled analysis jobs for regular monitoring
- Implement alerting based on analysis results
- Expand the analysis toolkit with additional specialized algorithms
- Connect analysis outputs to project management and resource planning systems