ISYE 6501
Introduction to Analytics Modeling
Course Project

This project should be done individually.

The web sites https://www.sas.com/en_us/customers.html, https://www.ibm.com/case-studies/search?search, and https://www.informs.org/Impact/O.R.-Analytics-Success-Stories (among others) contain brief overviews of some major Analytics success stories. In this course project, your job is to think carefully about what analytics models and data might have been required.

(1) Browse the short overviews of the projects. Read a bunch of them – they're really interesting. But don't try to read them all unless you have a lot of spare time; there are lots!

(2) Pick a project for which you think at least three different Analytics models might have been combined to create the solution.

(3) Think carefully and critically about what models might be used to create the solution, how they would be combined, what specific data might be needed to use the models, how it might be collected, and how often it might need to be refreshed and the models re-run. DO NOT find a description online (or elsewhere) of what the company or organization actually did. I want this project to be about your ideas, not about reading what someone else did.

(4) Write a short report describing your answers to (3).

For your reference, the original source of the project overview
https://www.ibm.com/case-studies/bbva-argentina

MY SOLUTION WILL START ON THE NEXT PAGE

Problem Statement

BBVA aims to optimize its Foreign Trade Process using a process mining tool to enhance adaptability to regulatory compliance changes, considering specific constraints and optimize the process. The project involves identifying tasks that require rework due to unintentional errors, understanding repetitive tasks, quantifying the impact, analyzing the root causes, and uncovering bottlenecks caused by the limited availability of the Central Bank verification service. Additionally, the project seeks to understand the behavior of demand within the operational hours of client companies.

Breakdown of the Problem Statement

1. Enhancing Regulatory Compliance Adaptability
   1. Improve the process adaptability to changes in regulatory compliance requirements
2. Process Optimization
   1. Identify rework tasks by understanding repetitive tasks
   2. Uncover and address bottleneck caused by the limited availability of the Central Bank verification service
   3. Demand behavior analysis

Proposed Solutions

For the problem statement 1-1, we can use random forests or gradient boosting to predict the impact of regulatory changes on the process.

For the problem statement 2-1, we can use K-Means clustering to identify tasks with a high-risk for rework.

For the problem statement 2-2, we can use linear programming to address bottlenecks caused by the limited availability of the Central bank verification service. It also helps optimize resource allocation and scheduling within the operating hours while respecting the sequence constraints.

For the problem statement 2-3, we can apply time-series forecasting models to predict demand patterns based on historical data.

For 1-1:

Required Data

- Historical regulatory changes (dates, types, descriptions)
- Process performance metrics (processing times, error rates, throughput)
  - This dataset will be used as the targets for adaptability
- On top of the above, we need to create features that capture the timing and nature of regulatory changes

Feature Engineering

- Regulatory change flags: create binary indicators for recent regulatory changes
- Time since last change: calculate the time since the last regulatory change

Define Targets for Adaptability

- Speed of compliance implementation (time taken to adjust processes)
- Changes in processing time or error rates following regulatory changes

Given the set of data above, we can use Gradient Boosting model to simulate the impact of potential future regulatory changes. This can help in proactively identifying necessary adjustments to processes.

For 2-1:

Required Data

- task_id: uid for each task
- operator_id: uid for the operator performing the task
- task_type: a category of the task (e.g., review, verification, etc.)
- processing_time: time taken to complete the task
- error_rate: percentage of errors associated with the task
- completion_date: date of the completion of the task
- rework_indicator: indication whether a task required rework or not (e.g. yes/no)

Additional Data Through Feature Engineering

- task_frequency: number of times each task type has been performed
- average_processing_time: average time taken for each task type
- error_count: total number of errors associated with each task type

Given the set of required data, we first need to normalize the data to ensure that all variables contribute equally to the model. We can then use K-Means clustering to examine the characteristics of each cluster, identify patterns, and look for clusters with high error rates or long processing times, which indicate potential rework tasks. Based on these findings, we can consider process improvements by revising processes related to the tasks identified as high-risk for rework and continuously monitor the performance of these identified task types.

For 2-2:

Required Data

- t_i: processing time for task i
- x_i: binary variable that is 1 if take i is processed, and 0 otherwise
- S_ij: binary variable that is 1 if task i is completed before task j, and 0 otherwise

Given the set of required data, we can use a linear programming optimization to maximize the total number of trade transaction documents processed within the Central Bank’s verification service hours while respecting the sequence of the tasks

Here’s a set up for the linear programming optimization.

Decision Variables:

- t_i: processing time for task i
- x_i: binary variable that is 1 if take i is processed, and 0 otherwise
- S_ij: binary variable that is 1 if task i is completed before task j, and 0 otherwise

Objective Function:

- Maximize total sales. Let's denote S_i as the sales per unit of shelf space for product i.

$$Maximize \sum_i x_i$$

Constraints:

- Processing Time Constraint:

  $\sum_i t_i * x_i \leq H$, where H is total available hours (9 hours)
- Sequencing Constraints (assume that there are 4 tasks):
  - For Task 1 to precede Task 2:

    $$S_{12} \leq x_1$$
    $$S_{12} \leq x_2$$
  - For Task 3 to precede Task 4:

    $$S_{34} \leq x_3$$
    $$S_{34} \leq x_4$$

This model will help you identify which tasks can be processed within the Central Bank's operating hours, thereby minimizing delays and optimizing the use of available time. If you need to consider other constraints (e.g., task dependencies, priorities), you can further refine the model accordingly.

For 2-3:

Required Data

- process_request_id: uid for the process request
- requested_ts: a timestamp for the requested time
- completed_ts: a timestamp for the requested process completion (optional)
- num_requests: total number of requests every minute

Assumptions

- Process requests come in only between 8AM and 5PM
- To make the time-series data analysis less complex (excluding external factors such as different customer, seasonal effect, public holidays, etc.), ignore the date and consider only the time from the timestamp data
- The granularity of the time-series data is in minutes

Given the set of required data and assumptions, we can use time-series forecasting models such as ARIMA to understand/predict demand patterns based on minute-level historical data recorded between 8am and 5 pm.

Data Collection Methods
The set of analytical solutions requires a lot of data. Acquiring and managing this data in itself would require a team of some data professionals. As the breakdowns show, there are different piece of information needed so I am providing data tables for the data collection for each problem statement. For simplicity, only table name, not table schema, is provided. Also, I assume that there exists data input tools for each table or a set of tables.

For the problem 1-1

- Regulatory Compliance table
  - For each regulatory compliance, keep its date of release, type, version, updated date, and description. A combination of regulatory compliance id and version would be a unique identifier for the table
- Task table
  - For each task requiring regulatory compliance, keep its processing time and error rate/number
  - For more details for each task, here's a list of additional fields
    - task_id: uid for each task
    - operator_id: uid for the operator performing the task
    - task_type: a category of the task (e.g., review, verification, etc.)
    - processing_time: time taken to complete the task
    - error_rate: percentage of errors associated with the task
    - completion_date: date of the completion of the task
    - rework_indicator: indication whether a task required rework or not (e.g. yes/no)
    - task_group: identifies the group of tasks assigned to each trade transaction
    - task_order: specifies the sequence in which tasks within a task group should be completed.

For the problem 2-1

- This requires only Task table that's already defined in the problem 1-1.

For the problem 2-2

- This requires only Task table that's already defined in the problem 1-1.

For the problem 2-3

- Request table
  - process_request_id: uid for the process request
  - requested_ts: a timestamp for the requested time
  - completed_ts: a timestamp for the requested process completion (optional)

Analytics Models Update Frequency
The frequency at which we update our analytics models depends on the nature of our data and the specific use case. Therefore, model update frequency should be provided differently for different problem.

For the problem 1-1

The frequency of updates on regulations and foreign import requirements can vary widely depending on the region, specific regulations in question, and prevailing economic and political circumstances. Data should be updated regularly and also in response to significant events such as economic shifts, geopolitical developments, or technological advancements. In other words, the model should be updated regularly (i.e., monthly) and also in response to significant events.

For the problem 2-1 & 2-2

The process of trade transactions (using Task table) occurs between 8 am and 5 pm on weekdays. One important metric for the model is the error rate, which should be updated periodically in response to the amount of newly created data; therefore, the update frequency of the model is recommended to be on a weekly cadence.

For the problem 2-3

Since demands are recorded between 8 am and 5 pm at the minute level, new data is available daily. However, the model should be updated by weekly cadence since the timeframe of my interest is in between 8AM to 5PM on weekdays and I only consider the time section of the timestamp and distribute them into this time frame.