











<p>PREDICTION TASK </p> <p>Type of task? Regression (predicting a continuous value - the amount of electricity generated).</p> <p>Entity on which predictions are made? Photovoltaic (solar) panels.</p> <p>Possible outcomes? Predicted watt-hours of electricity produced by the panels.</p> <p>Wait time before observation? Immediate for predictions; actual observation wait depends on the data collection intervals (e.g., every 15 minutes).</p>	<p>DECISIONS </p> <p>How are predictions turned into proposed value for the end-user? Predictions can be integrated into a Google Calendar, allowing the administrator to visualize and manage energy production schedules, maintenance, and usage optimization. This aids in planning and operational efficiency without manual inference.</p>	<p>VALUE PROPOSITION </p> <p>Product: Friendly user calendar with the prediction of electricity.</p> <p>Alleviates: The users will just have to look at the prediction and not have to infer themselves about the production.</p> <p>Advantages: Better agreement between teams inside Uliege and easier access and/or use of the photovoltaic panel maintenance and usage</p>	<p>DATA COLLECTION </p> <p>Strategy for initial train set & continuous update: Utilize historical data from the University's dataset.</p> <p>Trying to get new data with the help of the University.</p>	<p>DATA SOURCES </p> <p>Where can we get (raw) information on entities and observed outcomes? Data will be sourced from the University's private dataset. Additional data may be obtained from integrated meteorological sensors and APIs that provide real-time weather data, from the weather station of the University.</p>
<p>IMPACT SIMULATION </p> <p>Can models be deployed? Yes, models can be deployed after testing.</p> <p>Cost/gain values for (in)correct decisions? Cost includes potential inaccuracies in energy management and maintenance scheduling; gains from accurate predictions include cost savings, optimal resource utilization, and improved maintenance scheduling.</p> <p>Fairness constraint? Ensure the model's predictions do not inadvertently favor certain times/days.</p>	<p>MAKING PREDICTIONS </p> <p>Time available for this + featurization + post-processing? Immediate processing required; limited to a few seconds to ensure timely updates.</p> <p>Compute target? Must be lightweight enough to run on existing university servers or cloud infrastructure without significant cost increases.</p>		<p>BUILDING MODELS </p> <p>Techniques: Start with basic models (Linear Regression, Decision Trees), then explore ensemble methods) and a small neural network for better performance as well as a neural network that outputs some distribution parameters..</p>	<p>FEATURES </p> <p>Input representations available at prediction time, extracted from raw data sources: Features might include time of day, historical energy production data, current and forecasted weather conditions. WE want to have some sort of dashboard that will display the results for decision making.</p>
		<p>MONITORING </p> <p>Metrics to quantify value creation and measure the ML system's impact in production (on end-users and business)? Key performance indicators like Mean Squared Error (MSE)</p>		

ONLINE COURSE

Master the Machine Learning Canvas

Learn a step-by-step process to get to a complete and detailed Machine Learning Canvas. This will help you...

- Validate the feasibility of your ML use case ideas.
- Boost collaboration within your team.
- Anticipate issues that would otherwise come up during implementation or in production.

More details at ownml.co/plan

The screenshot shows a Miro board titled "Brainstorming MLC" with a Machine Learning Canvas template. The canvas is divided into several sections, each containing specific tasks and considerations:

- CHURN**
 - Prediction Task**
 - ENTITY: customer
 - WHAT FOR: subscription to and over a 3 mo.
 - OUTCOMES
 - Churn (Positive) - 10%
 - Renewal (Negative) - 90%
 - Decisions**
 - Filter out customers due to revenue in > 3 mo., inaction for 12 mo. or in highest cost
 - Filter out customers whose churn_prob < 80%
 - Predict if customers can be retained (separate system)
 - Filter out customers whose retention_prob < 80%
 - Start by discarding churn_prob < yearly_revenue * retention_prob
 - Value Proposition**
 - OBJECTIVES: reduce churn rate among OwnML subscribers; improve success rate of retention efforts by understanding why customers may churn.
 - WORKFLOW: Every month, get list of subscribers likely to churn (i.e. 2.5K / 12 * 90% = 1875). Target top 25% cluster according to past explanations, and retention response in 10% discount to each cluster.
 - QUESTIONS to answer:
 - Who are the most important and most fragile customers?
 - Who can we retain among them?
 - Do we understand with a separate system?
 - Which attributes are shared among customers? What does churn prediction? Are there patterns that could help us better retention response?
 - Data Collection**
 - INITIAL: snapshots taken at time T0... T0+3 of customers due to revenue 3 mo. later - wait until T0+3 to observe outcome. 2.5K / 12 * 90% = 1875 customers.
 - CONTINUOUS: Hidden test at time T. 50% of all customers due to revenue at T+3, selected at random - 90% customers. Hidden decision about any email. Observed outcome will be added to test set.
 - Data Sources**
 - Design for payment info (source: Stripe)
 - HighlyNetwork for OwnML platform analytics
 - ConversKit for email engagement
 - Linktree for professional info including role, company size
 - Relevancy for custom info (survey notes...)
 - Impact Simulation**
 - TEST SET: at time T, we observed outcome time T+3. Here are all customers due to revenue at time T+3 and not targeted (-100K)
 - GAIN vs COSTS
 - Among targeted customers:
 - TP provide gain of subscription_value * 90% * retention_prob
 - FP incur cost of subscription_value * 10%
 - DEPLOYMENT: total gain > 0, when using numbers of TP and FP on test set and applying self-pain values
 - Making Predictions**
 - Every month, for 1875 customers all customers not filtered out by application, and customer in highest cost
 - Make predictions for all of these (i.e. 1000-1000 * 90%)
 - Get explanations for targeted customers (25)
 - All counter-idea and oversight?
 - Results stored in database
 - Building Models**
 - FREQUENCY: 1 snapshot every month (3 months)
 - Features**
 - Design for payment info (source: Stripe)
 - HighlyNetwork for OwnML platform analytics
 - ConversKit for email engagement
 - Linktree for professional info including role, company size
 - Relevancy for custom info (survey notes...)
 - Live Monitoring**
 - Get explanations for targeted customers (25)
 - All counter-idea and oversight?
 - Results stored in database

machinelearningcanvas.com by Louis Bessard, Ph.D.