THE MACHINE LEARNING CANVAS

Designed for: predictive e-scooter-battery maintenance Designed by: Robin Hegenberg Date: 30.06.2023 Iteration: 2

PREDICTION TASK

Type of task? Entity on which predictions are made? Possible outcomes? Wait time before observation?

Type: Regression: predicting the remaining useful life of rechargeable batteries

Entity: individual rechargeable batteries of publicly shared e-scooters

Outcome: days until battery is useless (or use charge cycles).

Wait: until actual useful life ends: Fails to reach predefined minimum distance if fully charged. Distance is calculated by observed distance travelled and capacity used for each trip.

DECISIONS

How are predictions turned into proposed value for the end-user? Mention parameters of the process / application that does that.

Decision: Replace batteries only when they fall under a predicted useful life threshold (yes/no->send alert -> schedule immediate repair)

Such an application could take other parameters like accuracy and reliability of the predictions, the cost and availability of replacement batteries, and the expected impact on customer satisfaction into account

VALUE PROPOSITION

Who is the end-user? What are their objectives? How will they benefit from the ML system? Mention workflow/interfaces.

End-user: operators or managers of e-scooter sharing systems

Benefit: Reducing maintenance costs and improving customer satisfaction by minimizing downtime and maximizing battery life (avoiding unnecessary battery changes)

Workflow: In 2 steps: Integrate predicted remaining useful battery life into existing workflow as helpful additional information first. Create new workflow after validation phase: E-scooter sends alert -> schedule immediate repair -> change battery by technician

Interface: new dashboard over all e-scooters, display remaining useful life

DATA COLLECTION

Strategy for initial train set & continuous update. Mention collection rate, holdout on production entities, cost/constraints to observe outcomes.

Include initial battery data like initial expected life estimate, battery type, initial capacity, expected percentage of capacity when life ends

At: Every charge, for every e-scooter:

At charging start, Add the Charge cycle number ID, distance travelled ->total, capacity used, capacity left, and the average temperature of the trip

At charging end, add the capacity charged, the capacity itself and the charging time. ->predict

Costs to observe: current, temperature and distance sensors and maintain them (they last for ~15 years, which is much longer than the battery + often already included)

Use 10% of e-scooters to create a test set (holdout)

DATA SOURCES

Where can we get (raw) information on entities and observed outcomes? Mention database tables, API methods, websites to scrape, etc.

Create own dataset.

Get initial battery data from here: TUMFTM/TechnoEconomicCellSelection: Helps you select the optimal cell to electrify a long-haul truck from a database containing 160 cells. (github.com) (e-scooters batteries are not different), and/or from battery specification/ from initial measurement

Continuous data from sensors: Store current (capacity,…), temperature, distance sensor data in own database. Sensor data is often already available and shown in interfaces (speedometer, battery low/high), but not stored ->use same APIs, but store in own database.

IMPACT SIMULATION

Can models be deployed? Which test data to assess performance? Cost/gain values for (in)correct decisions? ~~Fairness constraint~~?

Deployment: At first, provide ML information as additional information to technicians in normal process (normal: check every half a year)-> compare if they agree to prediction.

After high agree-rate (>95%): Change process: Call technicians only if needed. Technicians do normal battery check if called to confirm need of change. Also, choose 5% of good e-scooters randomly -> to technicians every month to confirm that technicians still agree that

MAKING PREDICTIONS

When do we make real-time / batch pred.? Time available for this + featurization + post-processing? Compute target?

Start predictions after new data is available (end of every charging).

Real-Time is preferred but not necessary, it is sufficient to finish the prediction each day (technicians can only be called daily, not in real-time)

Compute on 2 servers

BUILDING MODELS

How many prod models are needed? When would we update? Time available for this (including featurization and analysis)?

One model only (all e-scooters typically use a similar battery, else: one model per battery type)

Build a new model if:

-the old one performs badly (customers reported not working batteries that the prediction said were ok, technicians disagreed with prediction (1 is enough))->build ASAP (max. 1 day)

-or when there is a lot more data to be taken into account (~every week)->build until next week

FEATURES

Input representations available at prediction time, extracted from raw data sources.

Initial battery information: initial expected life estimate, initial capacity, expected capacity at life end

Trip information: average temperature, distance travelled, capacity used, calculated max distance that the user potentially could've travelled with a full battery

Charging information: Loading time, capacity at start and end and capacity charged, charging cycle number those supposedly fine batteries don't need change.

Test data: holdout set of scooters

Cost/gain: battery cost + maintenance cost (technician salary) + user satisfaction influences

MONITORING

Metrics to quantify value creation and measure the ML system's impact in production (on end-users and business)?

Number of battery changes

Customer + technician feedback on battery (failure rate)

For each e-scooter: Calculated max distance possible to travel with full battery (calculated from last trip capacity usage) vs. Predefined minimum distance (e-scooter must fulfill this) vs. Prediction (->prediction is wrong if it says that battery still has a lot of time left when the e-scooter is dead) (is a measurement for a potentially bad impact)

Why machine learning?: New process avoids current flaws: instead of checking batteries regularly every half a year and sometimes unnecessarily replace them based on only one charge cycle loading time that is fluctuating a bit over the battery lifespan and depends on external values like temperature use ML to take all these values into account.