**INTRODUCTION**

**Motivation:** Use existing data to develop tools/models that add value to Tesla generating leads that help Tesla enhance factory operations.

**Goal:** Given historical text and reliability data our goal is to create a model to automate the process of classifying open text maintenance records into standardized groupings. We also aim to predict future downtime and identify poor-performing equipment.

---

**DATA PREPROCESSING & EXPLORATION**

- NaN removal, cleaning, formatting, stopword removal, vectorization.
- Time series modeling: X = [2021-10-12 19:24:14.97]; Observation: A
- "failure.mechanical": 0, "failure materia": 1, ...

**RTP** = Return to Production, PM = Preventative Measures, HMI = Human-Machine Intervention

- Removed words like "a, an, the, and, it, for, or, or, but, ...

**Integer Encodings:**
- Text = [61, 73, 87, 102, 201, 194, 81, 150, 18, 124, 267, 3, 24, 48, 206, 2, 321, 0, 0, 0, 0, 0]
- Sparse Vectorization: Text = [0.0, 1.0, 0.1, 0.0, ...; 1.0, 0.0, 1.0, ...]

**Clustering**
We used distance-based KMeans clustering to analyze the dispatches and see patterns in words.

**ADASYN**
ADASYN was applied to handle class imbalance and generate synthetic vectors for the minority classes using weighted distribution. These vectors balance the dataset, resulting in improved performance for the classification of the minority classes.

---

**NATURAL LANGUAGE PROCESSING**

We trained, evaluated, and compared models using **Dense Neural Network and Multinomial Naive Bayes** to classify the dispatch data with high accuracy using the semi-supervised approach.

**Confusion Matrix:** Diagonal elements correspond to correctly predicted cases.

**PREDICTION METHODS**

**Detect anomalies within the battery production data and predict when and what type of failures will occur in each hour.**

**Methods:** We separated trend, seasonality, and residual components of the time-series data through season/trend decomposition.

**Anomaly Detection:** Use binary detection to flag a trend as an anomaly and send an alert.

**Time-Series Forecasting:** By training ARIMA, we forecast the number of future failures on each machine providing an estimated time of failure.

**LSTM Prediction:** Training LSTM RNN to predict the number of future failures on each machine providing an estimated time of failure.

---

**PREDICTION PROCESS & RESULTS**

- Data cleaning process:
  1. Delete the missing values 2. Normalize data 3. Parse time stamp
- Detrend the data: Used regression and smoothing to remove the cyclical component. Feature engineering: We implemented a rolling average and moving standard deviation.

- Time-series forecasting: The trend line represents the residual plot for the Autocorrelation ARIMA.
- Input: timestamps, number of items produced per hour.
- Output: predicted number of items next hour

---

**FUTURE WORK**

- NLP: 1. Fine-tune the models by changing and testing with different hyperparameters and sampling techniques. 2. Update the models so that it classifies the dispatches into sub-categories for each failure group.

**Prediction:** 1. Update the logic for pretraining process of input data of LSTM model to the machine level. 2. Incorporate the results from the NLP analysis, refine the provided failure data and update the prediction model increasing granularity down to the part/component level.

---

**ACKNOWLEDGEMENTS**

Our utmost gratitude to Tesla for their collaboration with the Data Mine. Thank you to our mentors Casey Justus, Allor Okoye, Doug Pillars, and Victor Reynosa for their continued support. Thank you to the Data Mine and the accompanying staff.