End-of-Year Report

The results from all our research is as presented in the repository below:

https://github.com/TheDataMine/f2021-s2022-miso

Phase 1

Description: How do constraints impact solvers? Focus on **BINDING, CONSTRAINT_COUNT, CALCTIME_CORRECT**.

Outlier Detection

Outlier detection is data points that differ significantly from the rest of the data. In MISO's case, how constraints makes the runtime for the electric grid solution too fast or too long. We made a hypothesis that isolating data and analyzing data will give us better insight on the data before locating extreme solve times in the data. Our members used 2 different statistical approach which are - Box-plots and Z-score method. From both approach, we conclude that the median solve times is 62 seconds with 10344 outliers and in which 10325 are outliers that takes more time than normal. We then found 2 key takeaways which are simple outlier methods or even a 100 cutoff seems enough to determine outliers for now and there is not a strong dependence of outliers for constrains and binding constraints count.

Feature Importance

We leveraged feature importance analysis method to detect and predict high solve time cases. At this point in time, we did not have access to data from MISO so most of our analysis was done using publicly available data. The goal was to showcase how feature importance method could be implemented and used to analyze relevant features in a model or dataset. Our members utilized 4 different methods - Random Forest, CART Regression, PCA Score and Coefficients Logistic Regression.

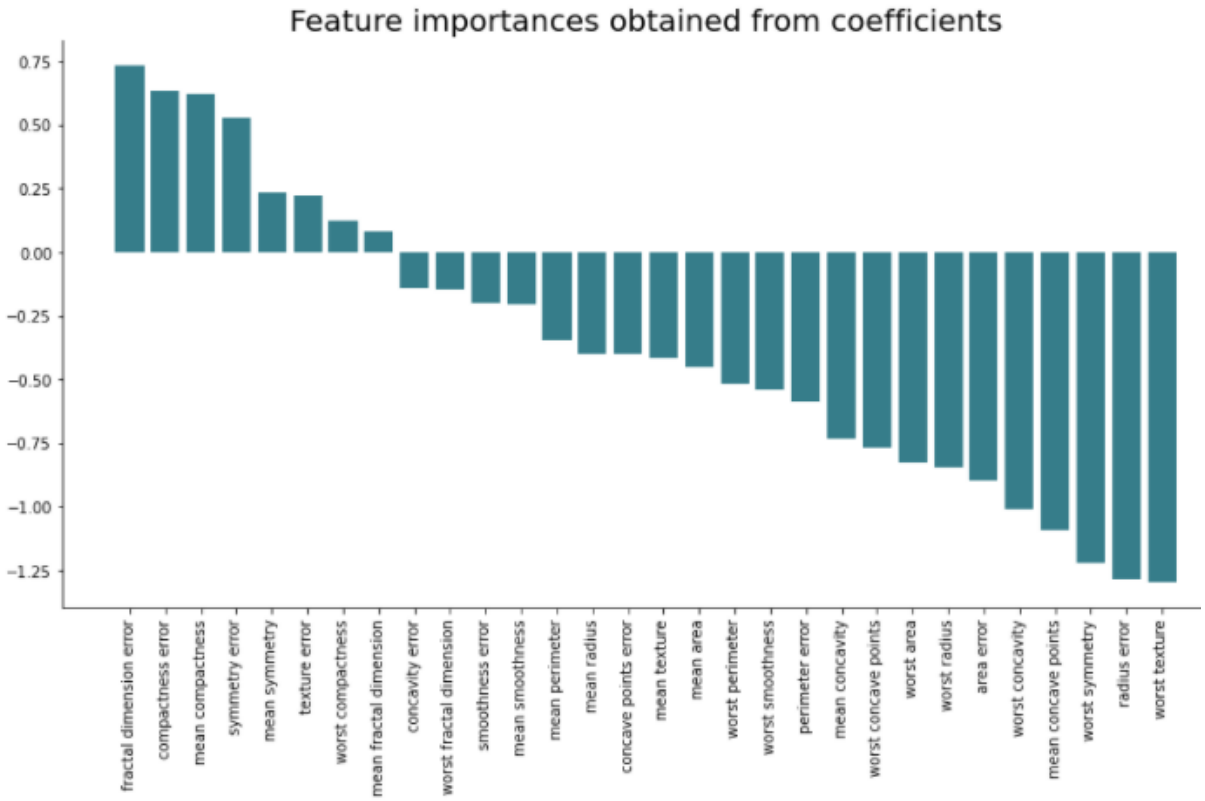
Random Forest is used to predict continuous variables based on data features such as constrains and solve time and create a dataset to test the accuracy of our model.

CART Regression is using feature importance implemented in scikit-learn and returned as the coefficient value for each feature.

PCA Score returns N principal components where N is the number of original features.

Coefficients Logistic Regression examines model's coefficients to understand its influence on the model. Larger the coefficient number, greater the influence on the prediction. This method was tested using a publicly available data to understand how the analysis method works.

After examining the results our analysis, we were able to relate it to the limited date provided to us by MISO and made assumptions. We believed some potential features that could contribute to higher/lower solve times were Queued Time, Exporter Lag, SPD Lag, and Calctime_Correct.



We wanted to use this testing method on MISO's data to understand which features affected solve time but unfortunately we could not receive access to more data from MISO.

Outlier Prediction

Leveraging Deep Neural Networks with Tensorflow, we were able to effectively predict whether a solve put into the engine would take a large amount of time to solve or not. We did this by creating a binary classification problem, considering any solve time of greater than 100s an outlier. Our best model was able to acquire a test data accuracy of **+.995636**, effectively predicting and solving the challenge set for the Phase 1 of our project.

The parameters used for training are as follows:

1. **Loss:** BinaryCrossentropy
2. **Optimizer:** Adam, learning rate - 1e-5
3. **Metrics:** Binary Accuracy, Precision, Recall
4. **Callback:** ReduceLROnPlateau

Phase 2

Collecting Data and Merging the dataset

We first search through the internet and collected a lot of data which can be found in the link below.. We then discussed the ideas and think of variables that might just effect the electricity grid from the increase in demand in EV. After that we added some datasets from the capstone team with our team and all the links can be found below.

Dataset	Link
MISO Daily Load Forecasting	https://www.misoenergy.org/markets-and-operations/real-time--market-data/market-reports/#nt=%2FMarketReportType%3ASummary%2FMarketReportName%3AHistorical Daily Forecast and Actual Load by Local Resource Zone (xls)&t=10&p=0&s=MarketReportPublished&sd=desc
Gasoline Prices	https://www.eia.gov/dnav/pet/hist/LeafHandler.ashx?n=pet&s=emm_epm0_pte_nus_dpg&f=m
PEV Sales by Model	https://afdc.energy.gov/data/10567
HEV Sales by Model	https://afdc.energy.gov/data/10301
Consumption and Pricing of Fossil Fuels	https://www.eia.gov/state/seds/seds-data-complete.php?sid=US#StatisticsIndicators
Charging Stations	https://afdc.energy.gov/stations/#/analyze?country=US&fuel=ELEC&ev_levels=all&access=public&access=private
Annual Population Estimates	https://www.icip.iastate.edu/tables/population/states-estimates
PEV + BEV Registrations 2019	https://evadoption.com/ev-market-share/ev-market-share-state/
Lithium Ion Battery Prices	https://www.bloomberg.com/news/articles/2021-11-30/battery-price-declines-slow-down-in-latest-pricing-survey
Historical Income	https://www.census.gov/data/tables/time-series/demo/income-poverty/historical-income-households.html
EV Registrations Yearly for each State	https://www.autosinnovate.org/resources/electric-vehicle-sales-dashboard
Gas Prices vs EV	https://www.self.inc/info/electric-cars-vs-gas-cars-cost/
Electric Vehicle Infrastructure Projection Tool (EVI-Pro) Lite	https://afdc.energy.gov/evi-pro-lite/load-profile

Next we combined the dataset into one huge dataset while subsetting the data only from years 2011-2018 and only states that are in MISO’s region - North Dakota, Minnesota, Iowa,

Wisconsin, Michigan, Illinois, Indiana, Missouri, Arkansas, Louisiana, Mississippi. Some of the data sources were not downloadable so we had to manually enter the data ourselves into an excel file and create a .csv file from it.

Below is one of our datasets that we had to manually enter values for as the source did not have an option to download the data.

> https://s3-us-west-2.amazonaws.com/secure.notion-static.com/79572f6c-2bde-462e-a01d-ce3d86bec077/US_EV_SalesData.csv

We spent a lot of time researching and gathering data as we could only rely on publicly available data.

EV Distribution

We researched data to understand how various factors and parameters have an effect on EV distribution. Some parameters we chose to focus on and found credible data on were Sales and Registration, Geo-spatial and Socioeconomic, Number of Vehicle Charging Stations, Household Income and Income Per Capita and Gas Prices vs. EV.

We found a lot of interactive data sources that we played around with to understand the data and how the parameters affect EV distribution. One source that was very helpful to us in the initial process of our research was a ‘Electric Vehicle Infrastructure Projection Tool’ from the US Energy Department. Another source that was helpful was ‘Electric Cars vs Gas Cars Cost in Each State 2022’ that provided us a foundational understanding of key parameters. We decided not to rely heavily on this second source as we could not validate the credibility of the data.

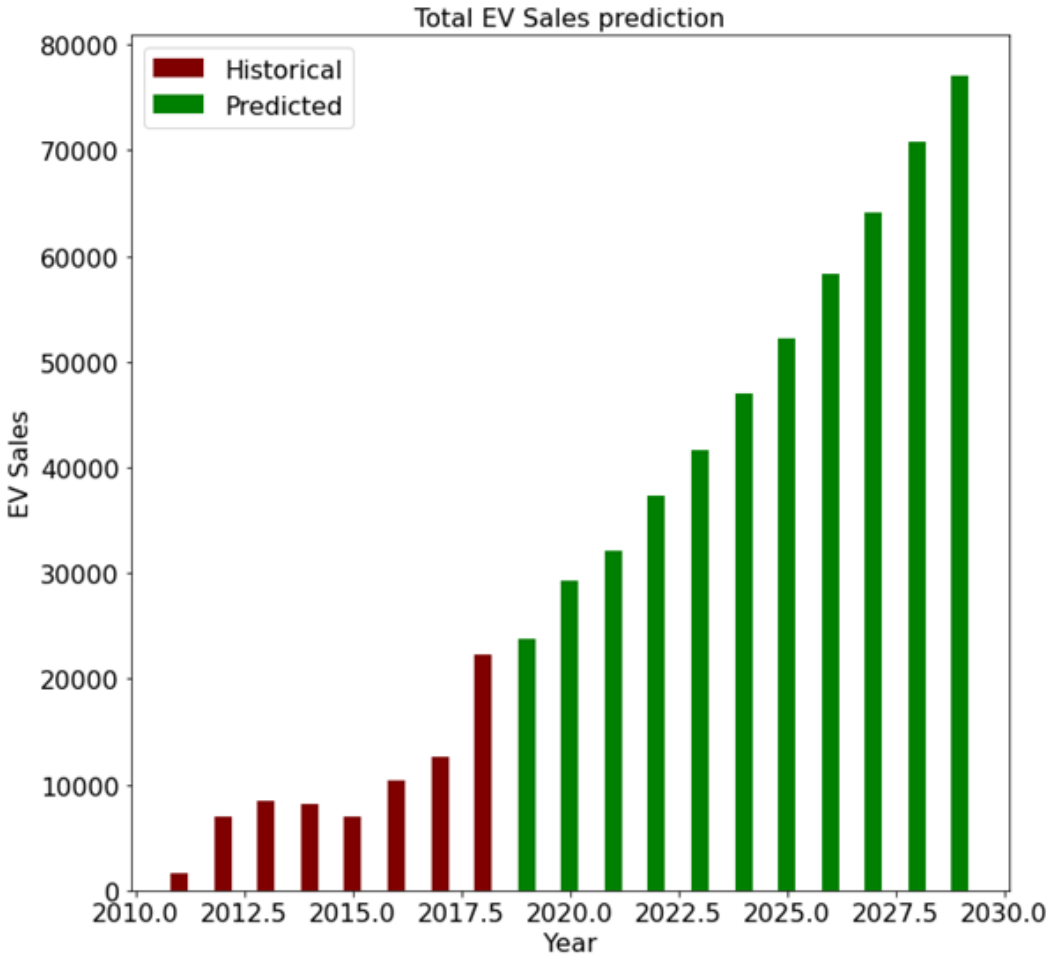
Load Profiles

We analyzed data from years 2011-2018. We cutoff our data at 2018 as we wanted to conduct our analysis for pre-COVID data. This made it easier to create a model as we were not sure how to factor in post-COVID data and its correlation.

Prediction Models

ARIMA

The objective of the ARIMA model is to predict sales of EV sales in states within MISO’s footprint. The methodology used here employs the ARIMA model with exogenous variables. We considered various socio-economic factors such as gas prices, state-wide median household incomes and price of lithium, which is the principal component of Li-ion batteries used in electric vehicles. The `statsmodels.tsa` module in Python was used to implement the in-built ARIMA model. The predictions from the same for each state were aggregated to obtain total EV sales predictions in states within MISO’s territory. The results are presented below:



It should be noted that while the general trend in prediction of EVs shows an increasing trend, the error margins associated with every prediction increase into the future. Furthermore, based on partial auto-correlation functions of sales data, we also noted that historical data had very little co-relation with current values.

However, we believe that this is a limitation arising due to use of open-source data, which is not granular and also of low reliability. Therefore, while the methodology presented here could be implemented on more accurate and sophisticated datasets, the significance of current predictions is limited in scope.

TensorFlow

The general idea of implementing TensorFlow model to predict future EV sales by state is to determine the correlation/significance of each variables (i.e. Gasoline Price, Charging Stations, etc) towards the end goal — future EV sales itself. We used TensorFlow since the model performs well in value forecasting.

To normalize the data, we used the StandardScaler function from the scikit-learn library. For the model itself, we used Adam optimizer to optimize the model with 10,000 epochs, 10^-3 min delta, and 20 patience. There is an individual model for each state.

The result that we found was that despite the model training on data well, evaluation on test data showed the model was overfitting on the training data, and was memorizing information as opposed to learning the underlying pattern within it.

With this, we reached the conclusion that we were largely training on white noise, and therefore were unable to accurately predict future losses. What we have is a fully fledged pipeline, that given better data, can identify with considerable accuracy energy load profiles and their subsequent impact by Electric Vehicles.