

Mia Skinner

Heather Koo

CIS41B - Final Project

Project Overview:

For our project, we designed a shipment forecasting tool utilizing machine learning. We chose to build regression models since our goal was to predict continuous valued output. We pulled a dataset with over 80,000 product shipment records and split it into training (80%) and testing (20%) sets using the sklearn module to avoid overfitting. To determine the best fitting model we utilized multithreading to find which degree (2-8) resulted with the highest R^2 value, where the better the test data fits the graph, the closer R^2 is to 1.

We then plot the graph with the best degree after collecting user input from the GUI. Inside the GUI, users can adjust model variables, such as forecast duration, start time, and product. Each product is modeled separately because they all have unique shipment patterns. Products with less than 10 (MIN_DATA_PTS) data points were excluded from the user's choice list because they did not have enough history to model an appropriate graph. After selecting input, the plot of the model is displayed.

If the user would like to save the forecast results for later, they can select the 'Save' button. This triggers the variables needed to reconstruct the visualization to be saved to a pickle file, an entry to be created in the database, and a .csv file to be saved to the user's preferred location to work with the model outside the program. After re-opening the program, the main window reads from the saved 'Forecast.db' database to display the saved entries along with their MAPE calculation that describes how accurate the forecast is, as a percentage from 0-100%, where 100% is perfect.

File Name	Contents
final_gui.py	Contains the program's user interface and main().
final_visualization.py	Contains the machine learning Forecast model and charting methods for final_gui.py

forecastDB.py	Generates Forecast.db
shipmentsDB.py	Generates and populates Shipments.db
Forecast.db	Contains saved forecast entries. Generated by forcastDB.py, used by final_gui.py
Shipments.db	Contains historical shipments entries. Generated by shipmentsDB.py, used by final_gui.py and final_visualization.py
data_201811191543.json	Dump of shipments data in JSON format. Used by shipmentsDB.py
CIS41B final project plan.docx	Original project proposal

What We Learned:

We definitely learned a lot from this project. To create the machine learning model, we read Statistics/ML blogs and watched a Udemy course for Machine Learning with Python. This lead us to incorporating **numpy**, **pandas**, and **sklearn libraries**. While it was a fun challenge to understand the results and tune the model, we learned that polynomial regression is tough to fit all scenarios. There are some products that do not get a great forecast from the model. Even though we excluded those kinds of models whose R^2 value is lower than 0.2(MIN_R2) and MAPE value is greater than 100% from displaying to work around this issue, we learned to always consider that model results may be inaccurate.

Within the GUI, we learned how to collect validated date input and create a search bar that automatically filters for matching results in the listbox. We also learned how to implement double axis on the graph.

We also found out how important the design process was during the course of this project. We started working on our project early and spent a lot of time planning a solid design, which helped us to implement what we envisioned. While coding, we realized that some of our original project concepts did not make sense, while other, new ideas emerged. We originally had planned a one week forecast option, however we omitted this choice when we realized the x-axis for the plot would be in months for the other options. We also chose to implement features such as a search box and date

selector in place of displaying selections via text, as we wanted to explore new concepts instead of repeating ones already learned in class.

Overall, we realized that we had strong foundations. Even though we had to study something new on our own, it was not hard for us to understand others' explanations. We put a lot of effort into understanding the mathematical theories and statistical concepts underneath the code itself.

Last but not least, we worked well together as a team. While working on our parts separately, we shared our thoughts and inspired each other, which improved our work. We definitely learned a lot from each other as well.