# Predicting manufacturing defects for 3D printed parts

*The growth in 3D parts has been dramatic—can we really rely on suppliers for critical components?*


*Pick a supplier...*

*Order complex parts...*

*Did these really come out right?*

The additive manufacturing services sector (or **3D printing** contract manufacturers) grew 20% in 2023, and will grow 22.7% in 2024 to reach $7.5 billion.

"Because 3D printing is so versatile, covering plastics and metals, it has not only cornered the prototype market but is becoming more popular for bridge manufacturing and medium-volume end-use parts"*

*Source: https://www.forbes.com/sites/carolynschwaar/2024/05/29/3d-printing-the-bright-spot-in-us-based-manufacturing/

Final presentation and analysis for GC2. Author: Tom Keating

# How do we best rate suppliers in order to avoid defective part shipments?

*Defective incoming parts for JIT manufacturers can cause costly disruptions and delays*


Problems facing purchasers

- Supplier ratings don't always reflect defect rates*

- Suppliers don't always share defect rates (but do share indirect metrics)

- Numerous suppliers in the market, and many different techniques for 3D printing

Goals of this analysis

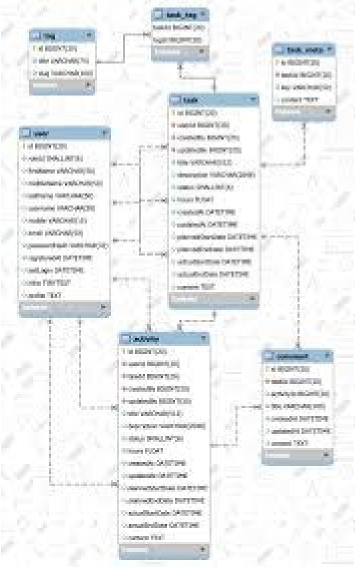
- Create a model which enables better supplier ratings by predicting defect rates

- If possible, use the model to flag shipments with high defect rates

*In fact, the 'supplier quality' rating in this dataset lacked any meaningful relationship to actual defect rates.

# We have a dataset, but can we build a successful predictive model with it?

*We are using the Predicting Manufacturing Defects dataset* from Kaggle*



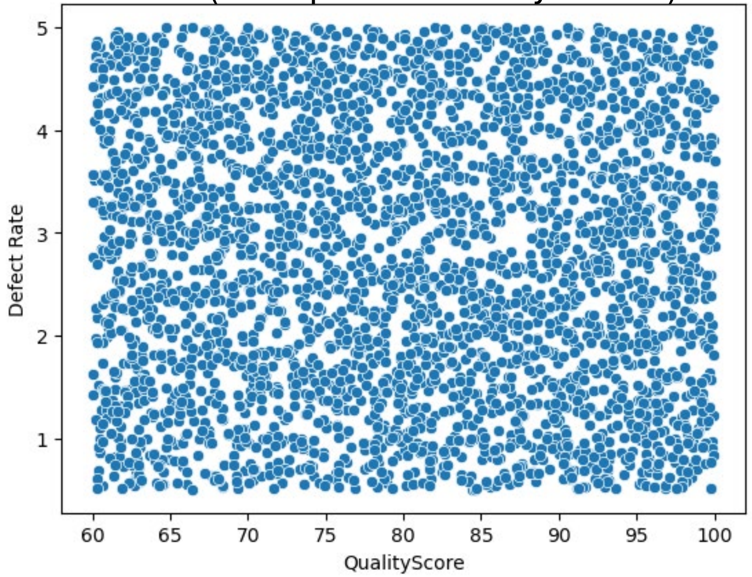
Dataset strengths

- 3240 3D records of part shipments
- Measured actual defect rates (parts/1000)
  - 44% with high defects
- Numerous other metrics supplied

Dataset weaknesses

- Low correlation, noisy data
- Unclear linkage between certain supplier metrics and defect rates

Scatterplot showing defect rate vs dataset variable (most plots were very similar)




***Can we find a signal in this noise?***

# The database contains a rich set of manufacturing-related metrics

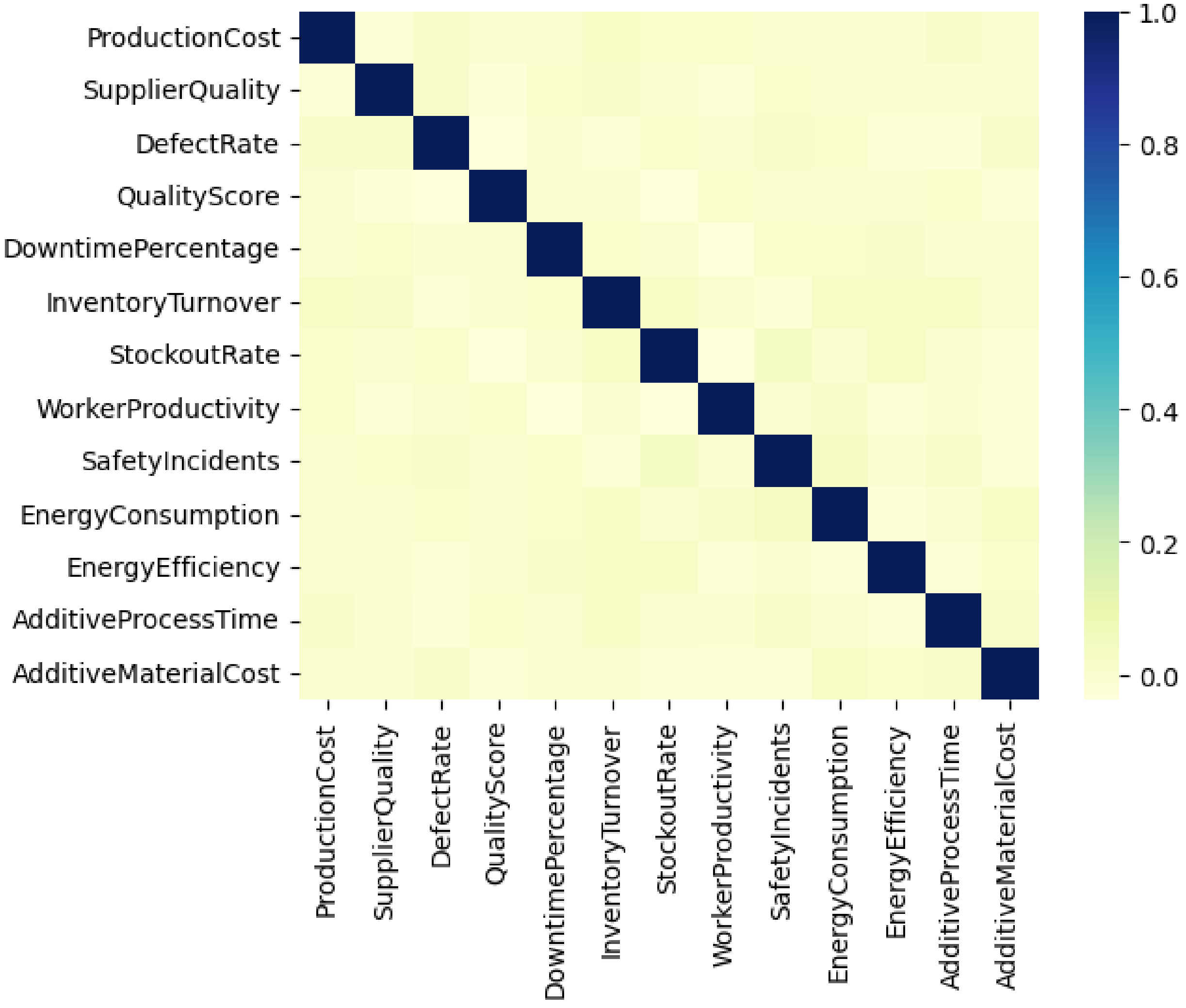
*Summary of metrics and key characteristics*

| Metric | Range | Comments |
|---|---|---|
| **Production Volume** | 100-1000 units/day | relatively uniform (but noisy) |
| **Production Cost** | $5000-20000 | |
| **Supplier Quality** | 80%-100% | Rating criteria unknown—excluded from model |
| **Quality Score** | 60-100% | Rating criteria unknown (“overall quality assessment”) |
| **Maintenance Hours** | 0-24 hours/week | Relatively uniform |
| **Stockout Rate** | 0-10% | |
| **Inventory Turnover** | 2-10 (ratio of turnover) | |
| **Worker Productivity** | 80-100% | (low predictive value; excluded from model) |
| **Safety Incidents** | 0-10 incidents (per month) | Somewhat bimodal |
| **Energy Consumption** | 1000-5000 kWh | |
| **Energy Efficiency** | 0.1 to .5 | |
| **Additive Process Time** | 1-10 hours | (low predictive value; excluded from model) |
| **Additive Material Cost** | $100-500 | |
| **Delivery Delay** | 0-5 days | |
| **Downtime Percentage** | 0-5% | |
| **Defect Status** | | Author-supplied labels (not used in the model) |
| **Defect Rate** | 0.5 to 5 defects/thousand parts | Target variable |

- Analysis involved selecting the most informative among these
- Further data processing was required

# Finding patterns in the data was not easy

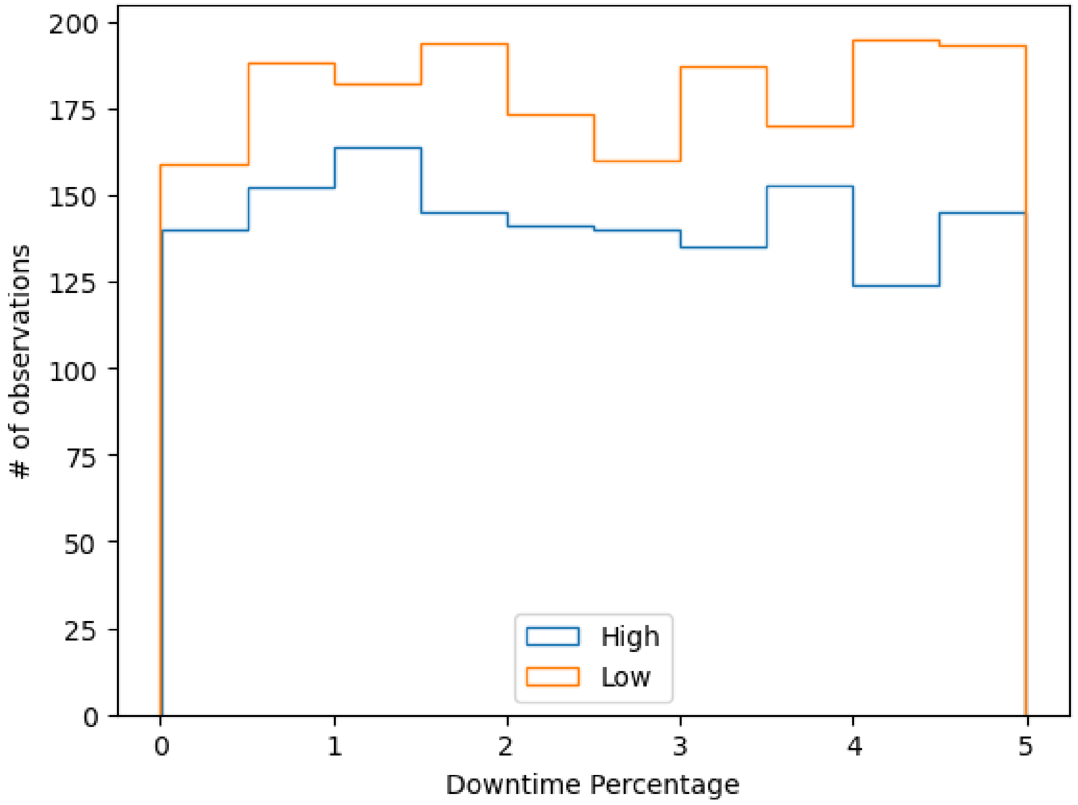
*The solution was to divide the dataset into high/low defects*




The correlation heat map showed only weak relationships between the metrics and the defect rate

# A better approach was to divide the data into shipments with high or low defect rates

*Patterns began to emerge at this stage: ‘High’ defects mean > 3/thousand parts*




**Trends:**
Low defect shipments follow an increasing trend as downtime increases (the opposite is true for high defect shipments)

**Rationale:**
Very low downtime could mean insufficient time spent inspecting/calibrating printers

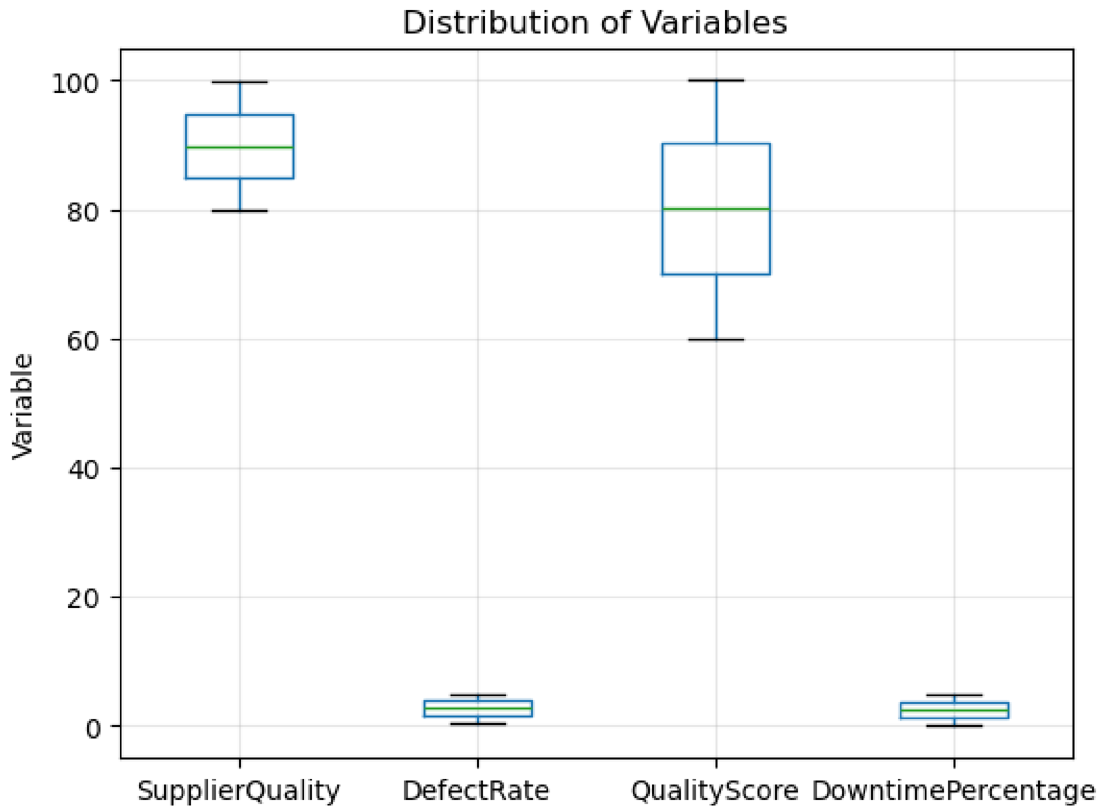
*Finding a trend and a plausible rationale usually meant we kept the variable for modeling.*

*Accordingly, Downtime Percentage was used in the models.**

*The final model used all variables except supplier quality, worker productivity, and additive process time, which did not meet the above criteria.

# Additional processing was necessary before modeling

*Variables kept for use in the model needed further cleaning*



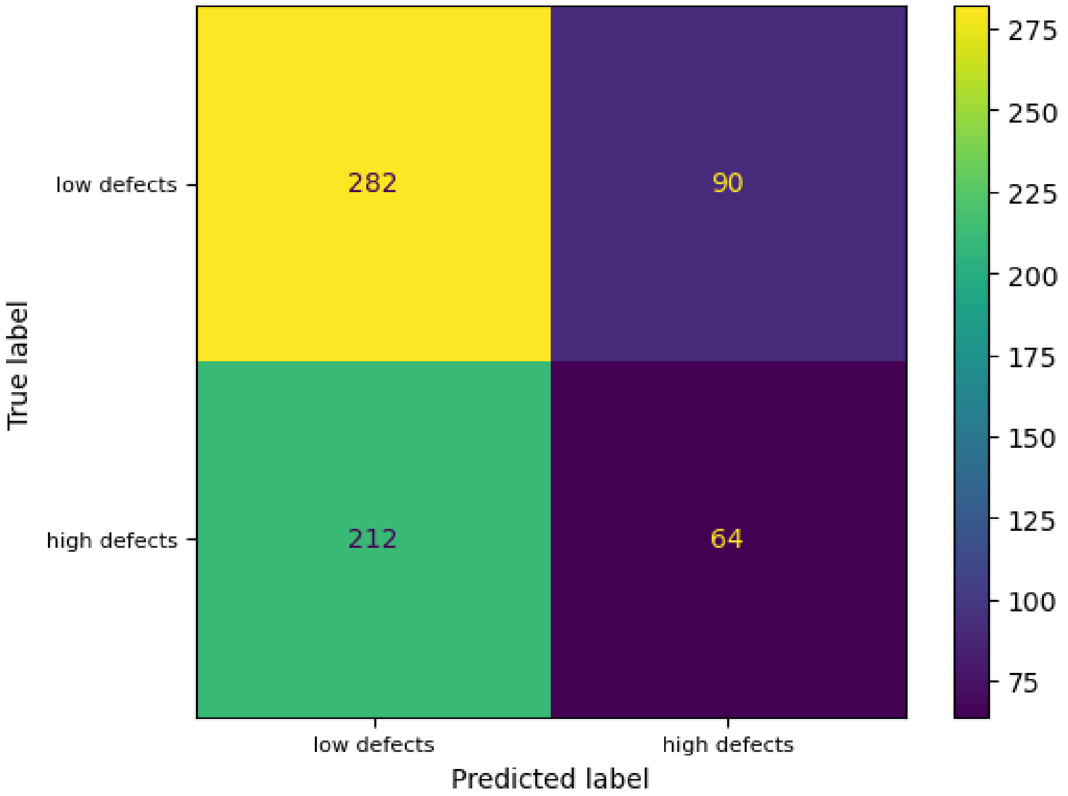

While the dataset did not require special treatments for outliers or missing data, nevertheless:

- Rescaling and normalization was needed
- Some features were treated as categorical (e.g., maintenance hours) rather than integer or continuous

*Note the markedly different means and ranges for selected variables.*

# Initial models showed some promise in predicting low-defect shipments

*But even with tuning and optimization, logistic regression models' performance was limited*




*Logistic regression model, after parameter optimization, found 76% of low defect shipments*

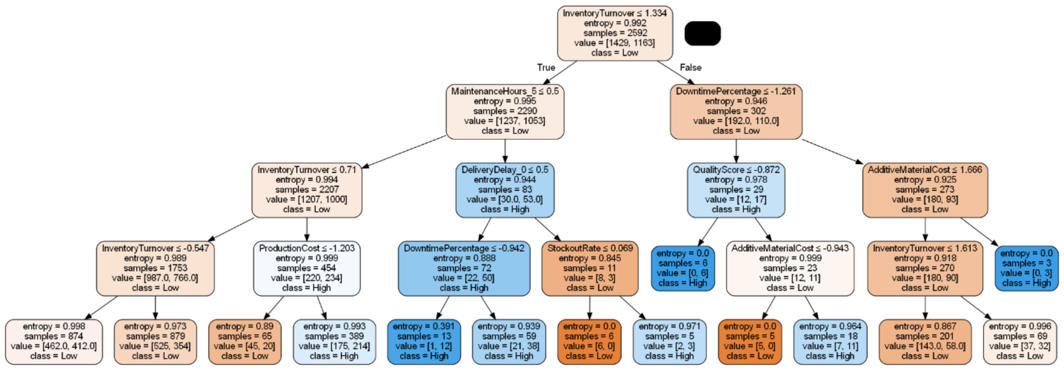
<u>Problems to be resolved</u>

- 44% of shipments have high defects, but the model is only finding about half of these (23%).
- The accuracy* was 53.4%, only slightly better than random guessing

*Accuracy is measuring how many shipments are labeled correctly as high or low out of the total.

# Decision tree models provided insights for further improvements

*The short decision tree model below utilized only the most informative variables.*




## Decision tree  model metrics - max depth 4

| | |
|---|---|
| Accuracy: | 54.3% |
| Balanced accuracy: | 50.0% |
| Precision score for "High" | 42.6% |
| Precision score for "Low" | 57.4% |
| Recall score for "High" | 21.0% |
| Recall score for "Low" | 79.0% |

## Key takeaways

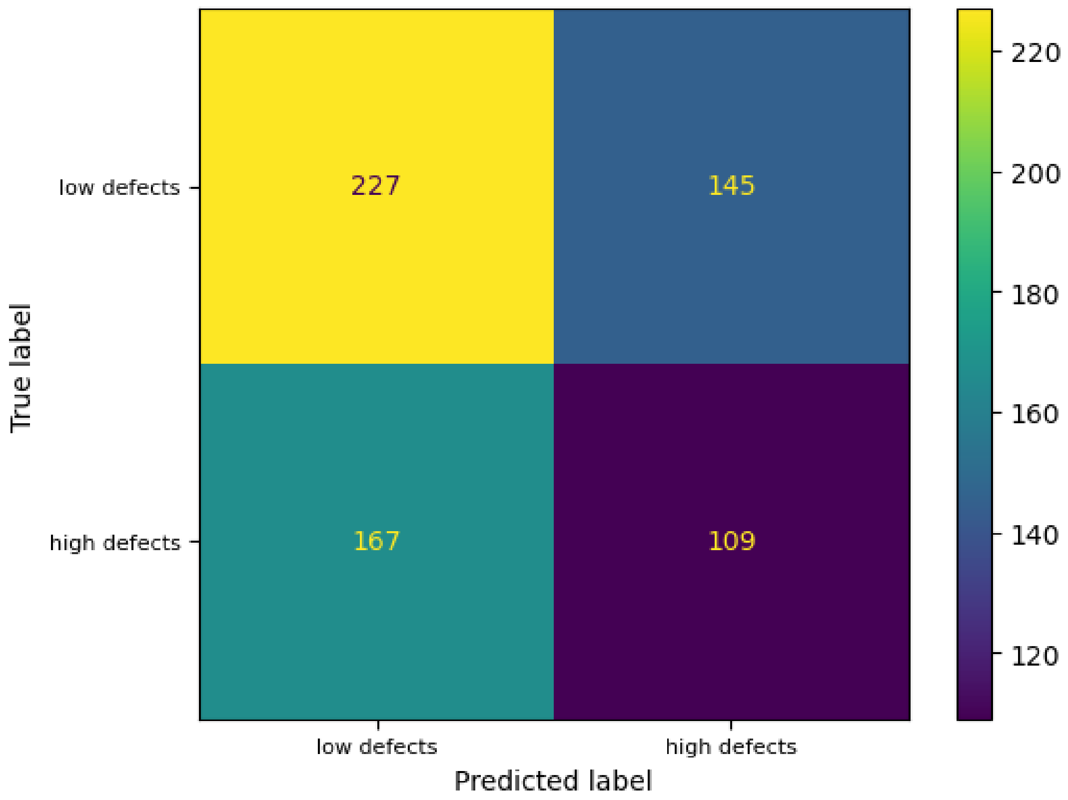
The model shown relied heavily on: *inventory turnover, maintenance hours, downtime percentage, delivery delay, and quality score;* each of which can be connected to defect rate with some plausible rationale.

Results for a single tree showed modest improvement over random guessing

Given the noisy data, averaging over many trees looked promising for the next modeling step

# Random forests led to more balanced results than other models

*Competing models could identify low-defect shipments well but often mislabeled or missed high defects*




*Confusion matrix for the (gradient boosted) random forest model: among all the models, this was the best at identifying high defect shipments*

<u>Model results and performance trade-offs</u>

- Accuracy was still low for the final Random Forest model (52%), but better than random guessing
    - Use model results to augment existing supplier rating models
- Models with higher accuracy are possible but come at the expense of missing a higher percentage of high-defect shipments
- All models were extensively optimized and validated for best performance
    - Tested but set aside: logistic regression, support vector machines

# Overall results are modest, but there is potential for improvement

*Some modeling decisions could be re-visited depending on performance required for final application*

| Proposed experiments | Rationale and potential benefits |
| --- | --- |
| Change the target: set the threshold for 'high defects' to >4% , for example | The choice to label a shipment as 'high' in defects was somewhat arbitrary; the choice could instead be driven by a threshold chosen by the end user. Subsequent models may work better in finding outliers, rather than picking lots that are somewhere in the middle range of defect rate. |
| Introduce engineered variables | There may be ways to combine variables that add value; e.g., ratio of additive cost to shipment cost, etc. Any improvement in explanatory variables would be useful, considering the lack of strong correlations to the target |
| Try other variations of SVM models | Not all variations were exhaustively tested (e.g. different kernels, SMVs versus SVCs, etc.) |