Leveraging Analytics to Improve the Efficiency of Assembly Line

Abhishek Bhambhu, Deeksha Goyal, Mengying Sun, Sudarshan Ananthakrishnan, Yang Wang
Purdue University, Department of Management, 403 W. State Street, West Lafayette, IN 47907
abhambhu@purdue.edu; goyal64@purdue.edu; sun907@purdue.edu; ananthas@purdue.edu;
wang3346@purdue.edu

ABSTRACT

This study investigates multiple machine learning methodologies that an operations manager can use to allocate appropriate tasks to suitable workstations efficiently. The motivation of the study is to help operations managers device strategies to improve the efficiency of their operations in assembly lines. With the use of predictive models, our goal is to help them not only understand but also evaluate the capability and performance of each workstation and assign tasks to them accordingly. Within the scope of the study, models have been developed to predict the completion times of incoming tasks so that an apt workstation can be scheduled well in advance for the planned tasks to improve production line efficiency. In addition to this, using machine learning models, various factors, and measures impacting the performance of the production line have been assessed. Also highlighted in this study are the possible features that are missing and could have been influential in this analysis. By collaborating with a structural building components manufacturer and using their data, we build and assess various analytics models and highlight some of those models which help derive quantifiable insights that operations managers can incorporate to improve their existing processes.

INTRODUCTION

Assembly Line has become omnipresent in the modern era. With a growing population and exponential demand for almost everything around us, assembly lines are a quick means to meet this requirement. It speeds up the manufacturing process dramatically. Also, the assembly line has come a long way since Ford's original conception, and the drive to optimize continues.

The need of the hour is to make integrated data-oriented platforms which can be leveraged for real-time decision making in the manufacturing industry. The assembly line, Henry Ford's innovation that revolutionized the automotive industry in the early 20th century by reducing the cycle time of building a car substantially, has been adopted extensively in many manufacturing processes [1].

An assembly line plays a significant role in any production process as it involves a considerable portion of the cost of a product. If the efficiency of the assembly line can be enhanced, productivity can be increased, and total processing time can be reduced. Optimization of the production line efficiency and reduction of downtime is always in the mind of an operations manager as reduced production time, setup time, and efficient use of workforce translates directly into profits.

The workforce is the backbone of manufacturing processes, and workforce planning should take center stage to make sure that the money spent on labor is used efficiently. An article on workforce planning by Gartner claims that most industries are don't have workforce planning robust enough to cope with ever-evolving work environments [2]. It is mainly attributable to the fact that most firms do not clearly understand the ever-changing supply-demand models. Forbes rightly states that using technologies to train and make the workforce competent by creating a real-time learning loop is the next big step for these industries.

This project aims to improve the efficiency of the line by developing analytical models by making use of the operational data provided by our industry partner to predict the completion times of tasks in the assembly line so that the workstations can be scheduled further in advance.

The Pipeline of all the assembly lines at the manufacturing unit, which manufactures trusses from which data for analysis was procured, contains gantry press, forklifts, and staple guns. A Gantry Press rides over the top of the assembly tables and does an initial pressing of the connector plates into the truss.

There is one gantry per assembly line, with lines sharing gantries in some instances. Some lines do not need this machinery. A forklift is required to move stacks of finished trusses away from the building once they are completed. All web and chord joints are pre-stapled before the installation of the connector plates. This is done for quality since unstapled joints tend to result in trusses with much more significant variance in their perimeters. Unlike most of the assembly lines today, which are automated and operated by robots, the assembly lines installed at our partner's location involve a higher human element.

Another objective is critiquing the data collection processes of the firm under study so that potential problems involving the workforce could be addressed through better and efficient data collection. Moreover, this project aims to assist our partner by highlighting essential variables impacting the performance of the lines. The collection of missing data, if any, that might have contributed to the modeling process, will also be suggested.

The remainder of this paper is organized as follows: A **Literature Review** describing various criteria and methods useful for the project is presented in the next section. The **Data** section throws light on the description of multiple variables that have been obtained or engineered from the derived variables. In the **Methodology** section, the adopted methodology is presented, and the criteria formulation is discussed. In the **Model** section, various models are evaluated and tested. The **Results** section compares the models that have been formulated based on the selected model metric and identifies the best model. The results of this model are then interpreted in detail. **Conclusion** section concludes the paper with a discussion of the implications of this study, future research directions, and concluding remarks.

LITERATURE REVIEW

A strategic goal for manufacturers using the assembly line is to schedule their orders in a manner that they can achieve maximum efficiency from each available resource. Identifying what factors impact the production time to manufacture a component is needed to ensure potential revenue maximization. If a large amount of assembly line data is available, the model is built to predict the completion time of a task allowing the management to schedule resources well in advance.

Prediction of the completion time of tasks in a manual assembly line is a complicated problem for several reasons. There is a considerable human intervention that makes it complicated to predict the time accurately, unlike in a fully automated assembly line where machines are programmed to complete a given task in a fixed period. Each human resource is different and has different productivity levels in each environment. Unless there is data that provides working patterns and times for all the people working on an assembly line, the model will always have some level of uncertainty. In an assembly line where each component built a custom one, further complications arise because the model has been trained on existing data and is being asked to evaluate the completion time of a task that has never been done. In this case, the model can only provide a close approximation of the actual result.

The findings related to understanding the impact of variables in an assembly line to predict completion times of task or improving efficiency and optimizing the process are mentioned below.

Studies	Motivation for Research	Algorithms Used	Results and Findings
(Meidan, Lerner, Rabinowitz & Hassoun, 2011)	Within the complex and competitive semiconductor manufacturing industry, lot cycle time (CT) remains one of the key performance indicators. Its reduction is of strategic importance as it contributes to cost decreasing, time-to-market shortening, faster fault detection, achieving throughput targets, and improving production-resource scheduling.	Decision Tree, Neural Network, Multinomial Logistic Regression Selective Bayes Classifier.	Compared with all the other models Selective Bayes Classifier demonstrates simplicity and interpretability as well as speedy and efficient model training.
(Tirkel, 2011)	Wafer fabrication is considered the most complex and costly challenge in the semiconductors industry. Cycle Time (CT), which denotes flow time, is one of its key performance measures. This work develops CT prediction models by applying Machine Learning (ML) and Data Mining (DM) methods. The models can assist in improving manufacturing and supply chain efficiency.	Decision Tree, Neural Network	The best fitted Decision Trees (DT) model achieves 76.5% accuracy, and the best Neural Network (NN) model (two hidden layers) achieves 87.6% accuracy.
(Lingitz, Gallina, Ansari, Gyulai, Pfeiffer, Sihn & Monostori, 2018)	The accurate prediction of manufacturing lead times significantly influences the efficiency of production planning and efficiency.	Linear Models, Tree Based models	Using Random Forest model, reducing the lead time from 20 seconds to 12.5 seconds.
(Apkinar & Bayhan, 2010)	There are three objectives to be achieved: to minimize the number of workstations, maximize the workload smoothness between workstations, and maximize the workload smoothness within workstations.	Genetic Algorithm with Ranked Postitional Weight Technique	Hybrid Genetic Algorithm when compared to Genetic algorithm outperforms by 1 workstation which would mean less equipment cost to purchase, payoff and maintain.
(Rane & Sunnapwar, 2017)	A methodology is proposed to reduce cycle time and time loss due to important factors like equipment failure, shortage of inventory, absenteeism, set-up, material	Branch and Bound Algorithm	Increase in the output by 11%

	handling, rejection and fatigue to improve output within given cost constraints.		
(Wang & Jiang, 2017)	In the traditional order completion time (OCT) prediction methods, some mutable and ideal production data (e.g., the arrival time of work in process (WIP), the planned processing time of all operations, and the expected waiting time per operation) are often used.	Deep Neural Network	From the experiment results, it can be observed that DNN can be successfully applied into the OCT prediction. DNN can not only solve the problems that NN applied in large-scale job shop would arise, but also decrease the information loss that PCA would result in during the dimensionality reduction.
(Nilakantan, Ponnambalam & Nielsen, 2017)	Minimizing cycle time for straight and U-shaped Robotic Assembly Line Balancing	Particle Swarm Optimization	In manufacturing systems, optimizing cycle time is an important problem to be solved. Since the problems addressed here is well known as non-deterministic polynomial-time-hard, particle swarm optimization is developed to solve the problem.

Table: Previous Research

We have used the following types of models:

Model Selection

1. Linear Models
2. Tree-based Models

Model evaluation metric

Root Mean Square Error (RMSE). Root mean squared error (RMSE). It is the standard deviation of the prediction errors of the model. The residual is calculated by measuring the distance of data points from the regression line. RMSE gives an idea of how spread out the residuals are. RMSE can paint a picture of how concentrated or spread out the data is around the best regression fit.

DATA

Data Description

The dataset was provided by an undisclosed industrial partner in the wood components manufacturing industry. The data contained various parameters for six assembly lines they run to manufacture the Roof Trusses. The data was available on daily level, i.e., how many units and the total specification of all the units built on each day.

Workstations: Each assembly line (such as Line 1, Line 2, or the Red Line) is a workstation. Each assembly line builds only one truss at a time.

Data Dictionary

Variable	Type	Description
Line Name	Text	Unique Name for each assembly station
Line ID	Numeric	Numeric ID value for each Line Name
Date	Date	Date of the observation
People	Continuous	The average number of employees declared to be working at the Line on the observation date. This is an ESTIMATED value, as we do not track all employee movements by the moment that they occur.
Hours Ran	Continuous	The number of hours that the Line was ran on the date of the observation.
Tot EE Hours	Continuous	The People value multiplied by the Hours Ran value.
Units Blt	Continuous	The number of trusses built during the observation date.
Ft Blt	Continuous	The number of board feet built on the observation date. Board footage is a function of the area and length of the boards that make up a truss.
Span Count	Continuous	The number of different truss spans, or lengths, built on the observation date. This provides an estimate of required jig / setup changes that were done during the interval.
Chord Count	Continuous	The number of chord members installed on the observation date. 'Chords' refer to the boards that make up the exterior of each truss.
Web Count	Continuous	The number of web members installed on the observation date. 'Webs' refer to the boards that make up the interior of each truss. Webs are basically the supports for the chords.
Plate Count	Continuous	The number of connector plates installed on the observation date. Plates connect chords to chords, chords to webs, and webs to webs within trusses. Connector plates are installed on each face of each truss.
TC Pitch Count	Continuous	The number of top chord pitches built on the observation date. This gives an estimate of required jig / setup changes that were done on that date.

BC Pitch Count	Continuous	The number of bottom chord pitches built on the observation date. This gives an estimate of required jig / setup changes that were done on that date.
WG Count	Continuous	The number of heel wedges (WG) installed on the observation date. When the ends of trusses (aka, heels) are raised slightly but the top and bottom chords still touch at the heel, a wedge is generally required to form a strong heel connection.
SD Count	Continuous	The number of heel sliders (SD) installed on the observation date. When the ends of trusses (aka, heels) are raised slightly and the top and bottom chords no longer touch one another at the heel, a slider is installed to form a strong heel connection.
GB Count	Continuous	The number of gable vertical members installed on the observation date. Gable trusses are used on the ends of buildings and their verticals basically form a wall that is the shape of the roof.
VS Count	Continuous	The number of vertical studs installed on the observation date. Vertical studs are like gable verticals but are installed between the webs in the truss.
Units/ EE Hour	Continuous	The number of units built on the observation date divided by the total number of employee hours.
Ft/ EE Hour	Continuous	The number of feet built on the observation date divided by the total number of employee hours.
Estimated Full Setups	Continuous	An estimate of how many new setups were done on the observation date. A setup would be done when the next truss being built is entirely different than the truss that was just built.
Estimated Heel and Span Adjs	Continuous	An estimate of how many heel and span adjustments were done on the observation date.
Estimated TC and BC Adjs	Continuous	An estimate of how many Top Chord and Bottom Chord adjustments were done on the observation date.

Table: Data Definitions

METHODOLOGY

Our study is divided into 5 major components:

1. Data Exploration
2. Data Preparation
3. Model Building
4. Model Evaluation and Comparison
5. Final Model Selection

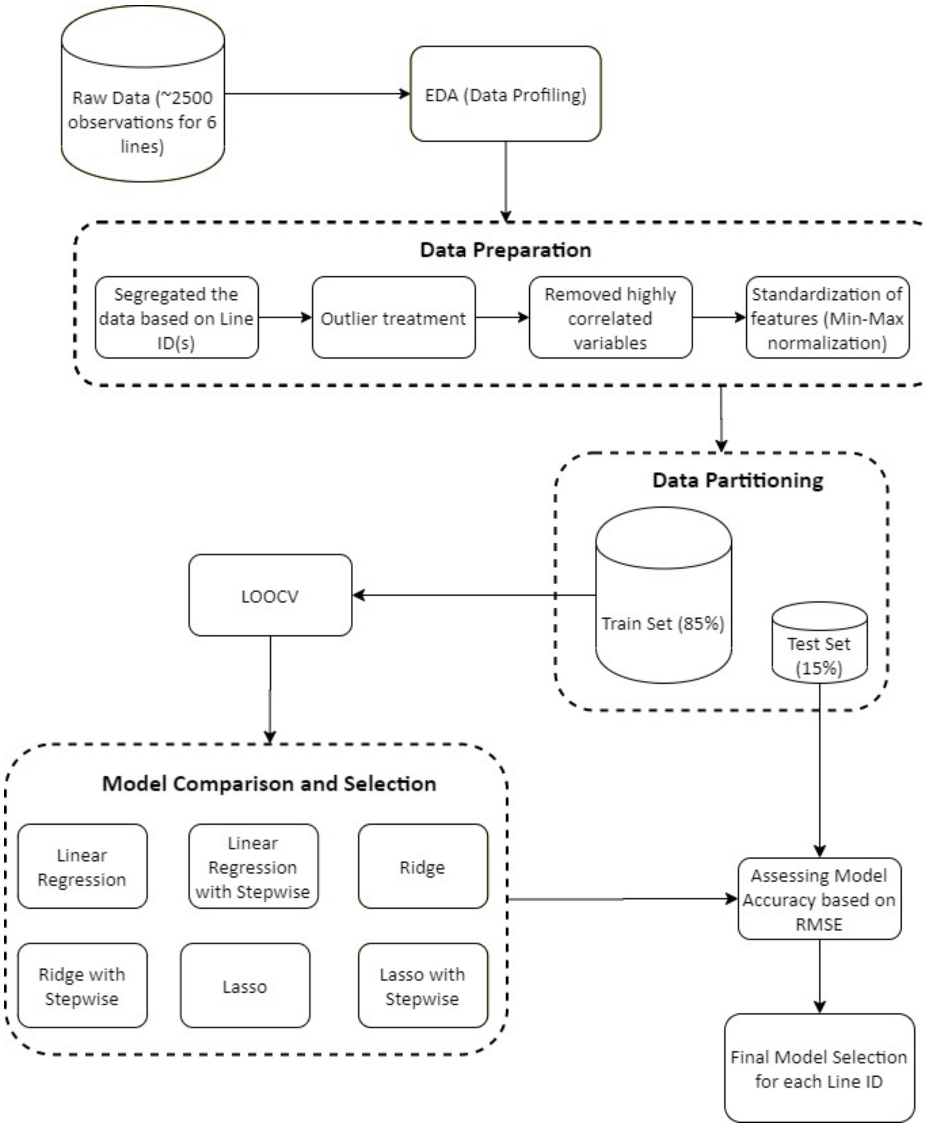


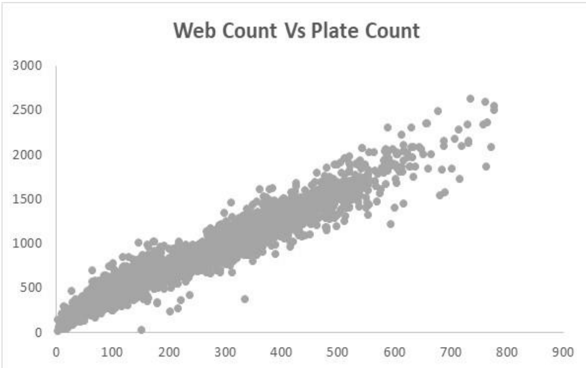
Figure: Study Design

1. DATA EXPLORATION

Exploratory Data Analysis

Understanding the data is a crucial prerequisite of building a machine learning model. The raw data provided was first analyzed using Python's Pandas Profiling package. This package is a one-stop-shop for exploratory data analysis as it allows for all the necessary information and graphs required to understand every feature of the data under investigation. This package also provides the distribution of every feature and correlation of every variable with all the other variables. All this information helps much not only to understand the problem at hand but also to gain astute business knowledge.

The data set is homogeneous in nature with highly correlated variables. Also, there were significant outliers present in the data.



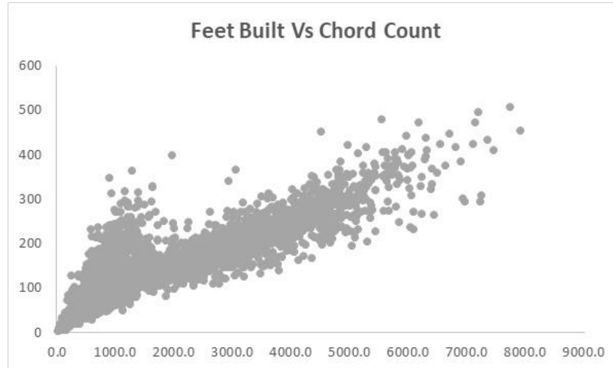


Figure: High Correlation between input variables

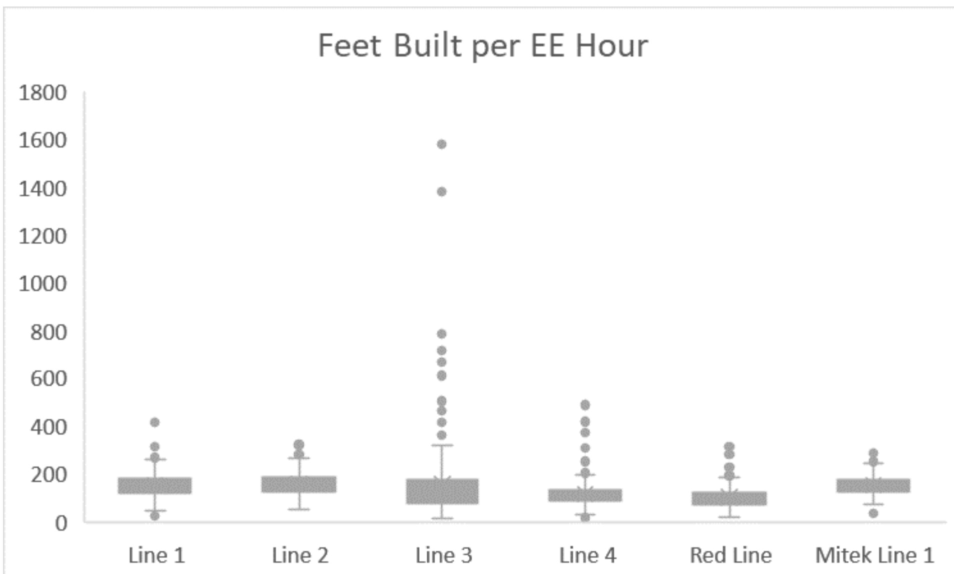


Figure: Presence of significant outliers in the data

Constraints

Setup times, Adjustment times and Idle times are fixed and hence cannot be optimized. Also, the efficiency of the workers employed in the assembly lines is assumed to be the same since we predict the number of employee hours required to complete a set of tasks in the assembly on a given day.

2. DATA PREPARATION

Data Cleaning

The data were inspected for any missing values and null values. Upon analysis, the data set was clean and was free from missing values.

Data Slicing

The dataset given contained information about all the assembly lines under study. To analyze the assembly lines, the data was sliced to obtain individual datasets containing data about respective assembly lines.

Feature Scaling

There were certain features like feet built, units built whose value ranges were higher than the other features. Using these features directly will place an unwanted importance on them thereby introducing a bias in the model. So, the input features were scaled to make sure that all the independent features are normalized.

Outlier Treatment

Outliers were identified for the dependent variable 'Total EE Hours' and all the rows where the value was an outlier and value's occurrence were rare was removed since imputing or treating outliers did not make sense in this case. Outliers were identified for the independent variable 'Feet Blt' and the rows containing outliers whose occurrences were rare were removed since it is assumed as the KPI of the lines under study.

Feature Generation

During this stage, features were generated, which could be critical in determining the business solution and could be used as direct inputs for our models.

Some of the features created were:

1. **Tot EE Hours**

Tot EE hours, which are total Employee hours, were calculated by multiplying total employee employed on a given day and the total number of hours the line ran for that day. This variable is later used as the dependent variable in our model.

2. **Units/ EE Hour**

Units/EE Hour, as the name suggests was created and used for exploratory data analysis by dividing the total units built on a given day by the total employee hours for that day.

3. **Ft/ EE Hour**

Ft/EE Hour as the name suggests was created and used for exploratory data analysis by dividing the total feet built on a given day by the total employee hours for that day.

3. MODEL BUILDING

Data Partition and features

The six datasets corresponding to the six assembly lines were partitioned into 85% training data and 15% test data. 'Total EE Hours' is used as the dependent variable for all the models, with all the other features becoming the dependent variable.

Algorithms Used

Linear Regression

A linear regression takes the form of $Y = a + bX$. Y is the dependent variable which is dependent on the independent variable X. The equation depicts a linear equation with slope b and intercept a. Ordinary Least-squares method is most common method of fitting a linear regression where we try to minimize the cost function, i.e. sum of the distance of all the datapoints is least from the linear regression equation is the least.

Because the deviations are first squared, then summed, there are no cancellations between positive and negative values.

Random Forest

Random forest is a learning technique that consists of bagging un-pruned decision trees with a randomized selection of features at each split. Initially, it draws n_tree bootstrap samples from the original data. For each bootstrap sample, it grows an un-pruned classification or regression tree. 16 Finally, the class, which has the highest number of votes across all trees in the forest, is used to classify the case (Breiman, 2001).

Regularization models

To improve the accuracy of the linear/OLS model, penalty terms can be added to the complex model equation. The addition of such penalty terms results in the regularization models: Lasso regression (L2) and Ridge regression (L1). These models are used, especially if there is a strong multicollinearity in the data. These models not only improve the prediction accuracy and interpretability of the model but also select only a subset of the independent features for the model by imposing a penalty on those features that overfit the model.

Models Developed

Model development started with Random Forest regression with a grid search hyperparameter tuning to select model parameters. Leave One Out Cross Validation (LOOCV) technique was used as the cross-validation technique to verify if the model developed was overfitting the data. However, after fitting one of the assembly line's data with this model, the results showed that the model overfits the data because the number of data points available for the line was minimal. Further, research about random forest revealed non-linear ensemble tree models such as this requires enormous volumes of data to make a correct prediction. Hence this model was discarded. The following are the models developed for all the assembly lines. LOOCV cross-validation technique has been used for all the models given the small size of the datasets.

- **Model 1:** A basic linear regression model with all the possible variables included with a LOOCV cross validation technique.
- **Model 2:** Ridge regression or L1 regularization model with all the possible input features with a LOOCV cross validation technique.
- **Model 3:** Ridge regression or L1 regularization model with all the possible input features with a LOOCV cross validation technique.

The following models were run on those features obtained for every line from the step wise variable selection used on the linear model (Model 1)

- **Model 4:** Linear regression with features obtained from stepwise selection with a LOOCV cross validation technique.
- **Model 5:** Ridge Regression with features obtained from stepwise selection with a LOOCV cross validation technique.
- **Model 6:** Lasso Regression with features obtained from stepwise selection with a LOOCV cross validation technique.

4. MODEL EVALUATION

Model Evaluation Criteria

Root mean squared error (RMSE). It is the standard deviation of the prediction errors of the model. The residual is calculated by measuring the distance of data points from the regression line. RMSE gives an idea of how spread out the residuals are. RMSE can paint a picture of how concentrated or spread out the data is around the best regression fit.

$$\boldsymbol{RMSE_{f_0}} = \left(\sum_{1=}^{N}, \left(z_{f_i} - z_{o_i}\right)^2 / N\right)^{0\cdot5}$$

Where:

- Σ = summation (“add up”)
- (zfi – Zoi) Sup>2 = differences, squared
- N = sample size.

5. FINAL MODEL SELECTION

Model Performance and Best Model Selection

The six models developed for the six lines were compared based on their RMSE values, and the model having the lowest test RMSE score was chosen as the best model for the line under study. While comparing RMSE for the five models, RMSE for train data was compared among the five models. RMSE for test and train data within each model was examined. Less the variation between the test and train RMSE for the same model better the fit. The following are the best models and their parameters for every line.

Line 1: Ridge Regression with LOOCV

Variables: Span Count, Chord Count, Web Count, Plate Count, TC Pitch Count, BC Pitch Count, WG Count, SD Count, GB Count, VS Count, Units Blt, Ft Blt

Equation

10.96268016 – 3.28894693 * UnitsBlt **+ 2.87110952 *** FeetBlt **- 0.16817168 *** SpanCount **+ 2.22435033 *** ChordCount **+ 4.94832272 *** WebCount **+ 2.31124533 *** PlateCount **+ 2.67437690 *** TCPitchCount **+ 1. 01406915 *** BCPitchCount **- 0.36001219 *** WGCount **– 0.78044722 *** SDCount **– 0.06263338 *** GBCount **– 0.33581068 *** VSCount

Line 2: Ridge Regression with stepwise selection and LOOCV

Variables: Units Blt, Ft Blt, Span Count, Chord Count, Web Count, TC Pitch Count, GB Count

Equation

17.541971 – 10.551363 * UnitsBlt **+ 5.190971 *** FeetBlt + **1.365980 *** SpanCount **+ 9.035058 *** ChordCount **+ 8.253371 *** WebCount **+ 2.754457 *** TCPitchCount **– 4.123254 * GBCount**

Line 3: Lasso Regression with LOOCV

Variables: Span Count, Chord Count, Web Count, TC Pitch Count, BC Pitch Count, WG Count, SD Count, GB Count, VS Count, Units Blt, Ft Blt

Equation

1.9932553 – 8. 8073995 * UnitsBlt **+ 7. 1229030 *** FeetBlt **- 8.4922719 *** SpanCount **+ 8.7715709 *** ChordCount **+ 2.8250068 *** WebCount **+ 6.2493507 *** TCPitchCount **+ 4.9596739 *** BCPitchCount + **2.7530905 *** WGCount **+ 1.4055556 *** SDCount **– 0.8399247 * GBCount – 0.3751220 * VSCount**

Line 4: Ridge Regression with LOOCV

Variables: Span Count, Chord Count, Web Count, Plate Count, TC Pitch Count, BC Pitch Count, WG Count, SD Count, GB Count, VS Count, Units Blt, Ft Blt

Equation

4.355114259 + 0.647495608 * UnitsBlt **+ 1.395045465 *** FeetBlt + **0.093877378 *** SpanCount **+ 0.752414333 *** ChordCount **+ 1.419421653 *** WebCount **+ 1.565981188 *** PlateCount **+ 0.453486163 *** TCPitchCount **+ 0.406288046 *** BCPitchCount + **0.064801017 *** WGCount **– 0.006098387 *** SDCount **+ 0.090250061 * GBCount – 0.483799165 * VSCount**

Red Line: Lasso Regression with stepwise selection and LOOCV

Variables: Chord Count, PlateCount**,** TC Pitch Count, BC Pitch Count, SD Count, Ft Blt

Equation

4.954741 + 5.928707 * FeetBlt **- 2.976127 *** ChordCount **+ 9.195355 *** PlateCount **+ 8.272399 *** TCPitchCount **+ 2.181818 *** SDCount

Mitek Line: Lasso Regression

Variables: Span Count, Chord Count, Web Count, Plate Count, TC Pitch Count, BC Pitch Count, WG Count, SD Count, GB Count, VS Count, Units Blt, Ft Blt

Equation

18.5509 – 4.8423 * UnitsBlt **+ 6.7883 *** FeetBlt **+ 2.0328 *** SpanCount **+ 7.6339 *** PlateCount **+ 4.9911 *** TCPitchCount **– 2.5803 *** SDCount

Line Name	Linear Regression (all variables)	Linear Regression (Feature Selection)	Ridge	Ridge (Feature Selection)	Lasso	Lasso (Feature Selection)
Line 1 RMSE	2.60	2.56	2.56	2.57	2.58	2.62
Line 2 RMSE	4.0	3.89	3.96	3.99	3.82	3.88
Line 3 RMSE	3.32	3.65	3.33	3.81	3.30	3.67
Line 4 RMSE	1.34	1.38	1.30	1.37	1.32	1.38
Red Line RMSE	3.09	2.84	3.08	2.92	3.08	2.84
Mitek Line RMSE	3.34	3.10	3.35	3.09	3.35	3.10

Table: RMSE scores of the all the models

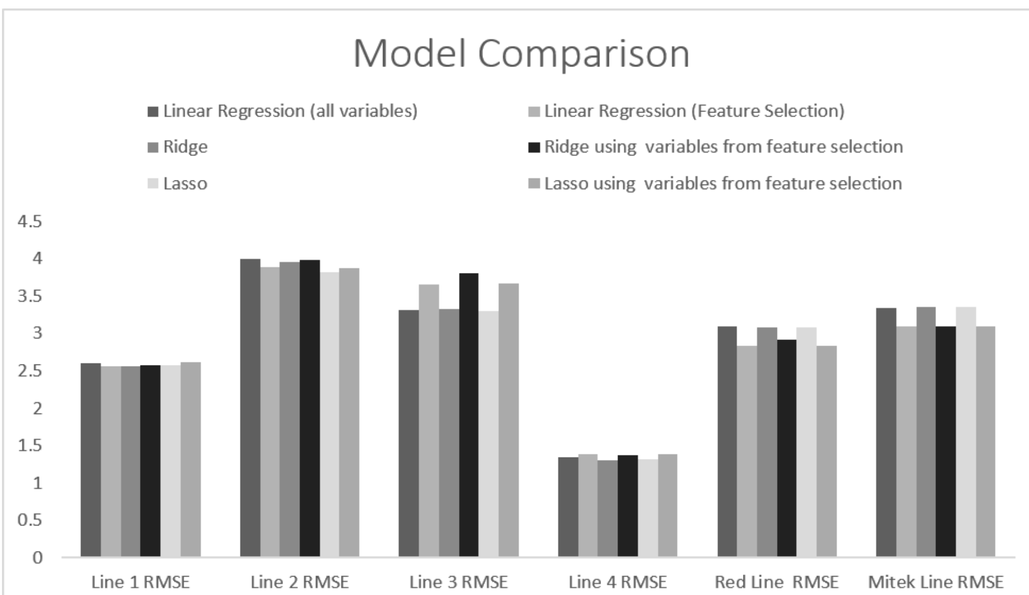


Figure: Model Comparison

CONCLUSION AND RESULTS

The final models predict a 2.1%-time reduction in the overall completion times of tasks for Line 4, Red Line and Mitek Line. Using the data from the month of Oct 2019 we captured the Sold $$ amount for Line 4, Red Line and Mitek Line. This amounted to $466,073. Assuming the time reduction is directly proportional to increase in feet built and the revenue increases by the same proportion, 2.1% increase in month of October for the 3 lines amounts to $9,718 or $116,621 annually. The total Sold $$ amount for all the lines in October 2019 amounts to $944,238. Extrapolating the same amount over the year gives the current revenue of $11,330,856. The increase in production capacity at Line 4, Red Line and Mitek Line adds $117K+ to the existing revenue of $11.33M.

The final models predict a 2.1%-time reduction in the overall completion times of tasks for Line 4, Red Line and Mitek Line. The 2.1% reduction in completion time amounts to 18 min reduction in completion time on average for all the 3 lines. Assuming, the 3 lines runs 5 days a week for all 52 weeks of the year, the total hours saved amounts to 242 hours.

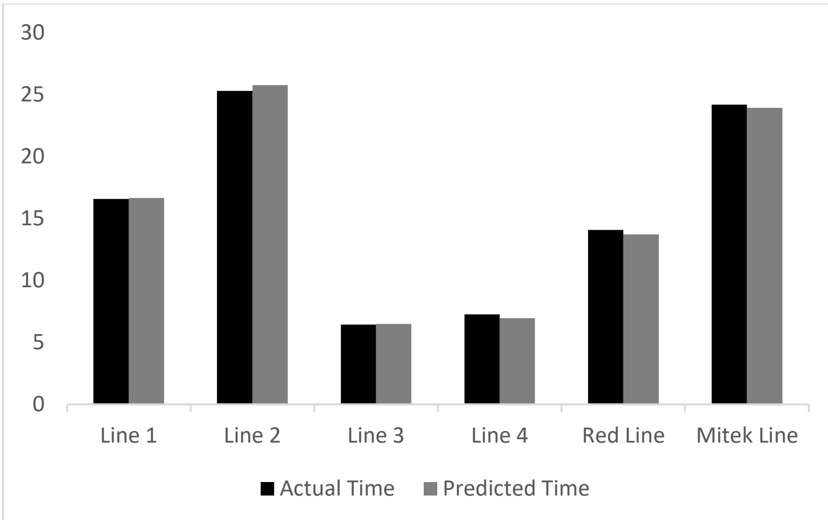


Figure: Actual Time vs Predicted Time for a task

FUTURE SCOPE

Data Collection Recommendation

Data is the cornerstone of any analysis, and procuring relevant data helps better understand the problem in hand and, in turn, helps to achieve actionable results. Data needs to be on the desired granular level. The issue at hand could have been analyzed better by collecting the data at a task level. Also, the volume of data available for each assembly line was not high enough to take up a nonlinear modeling approach.

Also, since the data under analysis is obtained from a labor-intensive process, recording the idle time will not make a lot of sense, but recording the setup times of every task could be used as KPI of the efficiency of the labor. Also, collecting operator centric data would help assess whether human intervention makes a significant impact on the completion times of tasks. It would also help in predicting the total employee hours required to complete the task since binning workers can exploit such data about workers into groups based on their efficiency. This could further aid in the allocation of responsibilities to operators based on their strengths. Also, it would lead the way to developing a robust training process for new workers just by deriving insights from the present worker's experience.

Prioritization

The brief explanation is that the delivery date for the orders determines the priority for production. If the data is collected at a unit level and the order level, various scheduling algorithms can be used to design a more robust scheduling policy at the partner's manufacturing firm. Order Prioritization, which is currently being determined by delivery date manually, can be automated using the scheduling algorithms.

Determine points of inefficiencies

Multiple points of inefficiencies could be determined along the whole supply chain process. For instance, currently the setup times and adjustment times of the equipment are fixed and cannot be changed. Hence, there was little scope to optimize these times in the Assembly Line. Other data pertaining to transportation of lumber and final product, breakdown time of the machinery and inventory can be collected so that inefficient processes can be identified and acted upon.

REFERENCES

1. A&E Television Networks (2009, November 13) Ford's Assembly Line starts rolling. Retrieved from https://bit.ly/2GZJ7Sd
2. Jackie Wiles (2020, January 8) From Workforce Planning to Planning work. Retrieved from https://gtnr.it/2GMNe3F
3. Y. Meidan, B. Lerner, G. Rabinowitz and M. Hassoun, "Cycle-Time Key Factor Identification and Prediction in Semiconductor Manufacturing Using Machine Learning and Data Mining," in IEEE Transactions on Semiconductor Manufacturing, vol. 24, no. 2, pp. 237-248, May 2011.
4. I. Tirkel, "Cycle time prediction in wafer fabrication line by applying data mining methods," 2011 IEEE/SEMI Advanced Semiconductor Manufacturing Conference, Saratoga Springs, NY, 2011, pp. 1-5.
5. Wang, Chuang & Jiang, Pingyu. (2017). Deep neural networks-based order completion time prediction by using real-time job shop RFID data. Journal of Intelligent Manufacturing. 10.1007/s10845-017-1325-3.
6. Lingitz, Lukas & Gallina, Viola & Ansari, Fazel & Gyulai, Dávid & Pfeiffer, András & Sihn, Wilfried & Monostori, Laszlo. (2018). Lead Time Prediction using Machine Learning Algorithms: A Case Study by a Semiconductor Manufacturer. Procedia CIRP. 72. 1051–1056. 10.1016/j.procir.2018.03.148.
7. Dao, Son & Abhary, Kazem & Marian, Romeo. (2017). Optimisation of Assembly Scheduling in VCIM Systems Using Genetic Algorithm. Journal of Industrial Engineering International.
8. Akpinar, Sener & Bayhan, Mirac. (2011). A hybrid genetic algorithm for mixed model assembly line balancing problem with parallel workstations and zoning constraints. Eng. Appl. of AI. 24. 449-457. 10.1016/j.engappai.2010.08.006.
9. Rane, A.B., Sunnapwar, V.K. Assembly line performance and modeling. *J Ind Eng Int* **13,** 347–355 (2017). https://doi.org/10.1007/s40092-017-0189-7