TIME PREDICTION MODEL FOR AUTOMOTIVE PRODUCTION LINE

for,

Dr. Alvis Fong

Associate Professor

Machine Learning

Western Michigan University

Kalamazoo, MI

By,

CS 5821 Group-16 Students

(Md Abiruzzaman Palok, Tarin Nurany, Ananta Poudel, Jiadong Gui)

Submitted April 20, 2023

TABLE OF CONTENT

Abstract

This research paper focuses on improving the efficiency of production lines in the automotive industry by creating a time prediction model to estimate the time taken by workers to complete tasks at specific stations. The dataset used in the study was collected from Freedom Motors USA and underwent rigorous data cleaning and validation. Multiple regression algorithms were utilized to identify the most appropriate model. All the models resulted in promising results with 90% accuracy but finally, the Polynomial Regression model was selected based on its high R-Squared value of 0.923 and low Mean Squared Error (MSE) value of 5127.105, indicating a high degree of accuracy in the predictions. The paper presents the research's methodology, findings, and suggestions for future research.

INTRODUCTION

Background

The automotive industry is constantly looking for ways to improve the efficiency and quality of production processes. In most production lines, companies usually have multiple stations and each of them performs a specific task before passing on to the next station. Inefficient time management at any station can cause delays and errors on the production line, leading to decreased productivity and profitability. In this research paper, we address the problem of inefficient time management in the production line by creating a time prediction model. Our model aims to estimate how long a worker will take to complete a task at a given station, providing a useful tool for the workers of the next station to prepare for their tasks based on the predicted remaining time. The objective of this project is to improve production line efficiency and potentially reduce delays and errors. We use a dataset collected from an automotive company, Freedom Motors USA, to train and evaluate our model. In this paper, we present our methodology and findings, along with a discussion of the implications and potential future research directions.

Audience

The primary audience for this research paper is expected to be professionals and academics working in the automotive industry, particularly those involved in production line management, process optimization, and efficiency improvement. This research paper presents a novel approach to addressing the problem of inefficient time management in production lines, which is a common challenge faced by many companies in the automotive industry. As such, professionals and academics who are interested in improving the efficiency and quality of production processes are likely to be interested in the methodology and findings of this research.

Additionally, this research paper can also be of interest to researchers in the field of machine learning, as it provides an example of applying machine learning techniques to solve real-world problems in the automotive industry.

Working Definitions

Some terms appear in this paper that may need to be clarified for the average reader. Knowing these will help further comprehension of the discussed data. These terms include:

1. MSE – MSE is a statistical measure of the difference between the actual values and the predicted values of a given dataset. It is a commonly used metric for evaluating the accuracy of regression models and is calculated as the average of the squared differences between the predicted and actual values. A lower MSE value indicates better performance of the regression model (Khanna, 2018).
2. R^2 Score – R-squared is a statistical measure of how well the regression line approximates the data points. Specifically, it represents the proportion of the variance in the dependent variable that is accounted for by the independent variable(s) in the model. R-squared ranges from 0 to 1, with higher values indicating a better fit of the regression line to the data (Hair, Black, Babin, Anderson, & Tatham, 2019).
3. One Hot Encoding – One-hot encoding is a technique used to transform categorical variables into numerical variables that can be used in statistical models. It involves creating a binary column for each category in the categorical variable, where each column indicates whether the category is present (1) or absent (0) in the data. This technique is useful for machine learning algorithms, as it allows them to process categorical data in a meaningful way (Torgo, 2021).

4. Label Encoding – Label encoding is a technique used to transform categorical variables into numerical variables by assigning each category a unique integer value. This technique is useful for machine learning algorithms that require numeric input, as it allows the algorithm to process categorical data in a meaningful way. However, it is important to note that label encoding can introduce an arbitrary order to the categories, which may not be appropriate for some models (Singh, & Singh, 2018).
5. Curse of Dimensionality – The curse of dimensionality is a phenomenon that occurs in high-dimensional spaces where the volume of the space increases exponentially with the number of dimensions. This leads to several problems for machine learning algorithms, including overfitting, increased computational complexity, and sparsity of the data. As the number of dimensions increases, the amount of data required to accurately represent space increases exponentially, making it difficult to train models on high-dimensional data (Bhat, & Kulkarni, 2020).

Model Limitations

Generalization to new employees: One limitation of our time prediction model is that it may not perform well when predicting the time required for tasks performed by employees that were not present in the training data. This is because the model may not have learned the unique characteristics and working styles of these new employees, which could impact their efficiency and productivity.

Generalization to new vehicles: Another limitation of our time prediction model is that it may not perform well when predicting the time required for tasks performed on vehicles that were not present in the training data. This is because the model may not have learned the unique

features and conditions of these new vehicles, which could impact the difficulty and duration of the tasks.

Variation in vehicle conditions: Our time prediction model assumes that the task duration is solely determined by the task itself and does not consider variations in vehicle conditions that could impact task difficulty and duration. As such, the model may not be able to accurately predict task duration when faced with different vehicle conditions such as damaged or modified vehicles.

Multiple workers on the same task: Our time prediction model assumes that each task is performed by a single worker, and therefore may not perform well when predicting the time required for tasks performed by multiple workers. This is because the model may not be able to account for the coordination and communication required between workers, which could impact task efficiency and productivity.

METHODOLOGY

Data Collection

The dataset used in this project was collected over a period of six months, from October 2022 to March 2023, at Freedom Motors USA. The company has check-in and out software that allows workers to check in to a specific car at a specific station. The raw data collected by the software was handed over to us for analysis.

The dataset included information on 47 workers, 789 distinct vehicles, and 29 different stations. The variables in the dataset were:

- EmployeeID – A unique identifier for each worker in the production line.
- EmployeeName – The name of the worker.

- VehicleStock – The stock number of the vehicle.
- VehicleYear – The year of the vehicle.
- VehicleMake – The make of the vehicle.
- VehicleModel – The model of the vehicle.
- Check-In DateTime – The date and time the worker checked in to a specific vehicle
- Check-Out DateTime – The date and time the worker checked out from a previously checked-in vehicle.
- Check-In Station – The station at which the worker checked in.
- WorkingTime (in minutes) – The time difference between the check-out and check-in time.
- Notes – Any additional notes the worker added about their work.

Data Cleanup and Validation

Data clean-up is a crucial step in the data preparation process for machine learning and data analysis. This process ensures that the data is of high quality and suitable for the intended purpose. The following are the essential steps that were taken to ensure that the data used for analysis was of high quality and suitable for the intended purpose:

Identify and Remove Duplicate Records: Duplicate records can distort analysis results, and hence it's crucial to identify and remove them. In this step, all the duplicate records in the dataset were identified and removed to avoid the distortion of analysis results. This process ensures that each record in the dataset is unique, and only relevant data is used for analysis.

Handle Missing or Null Values: Missing or null values can have a significant impact on the accuracy of analysis results. Therefore, it's essential to handle these values appropriately. In this step, missing or null values were input using appropriate techniques. Different imputation

techniques were used based on the data type and the amount of missing data. If a large portion of the data was missing, the record or feature was removed.

Remove Outliers: Outliers can be caused by measurement errors or other anomalies and can have a significant impact on analysis results. Therefore, it's crucial to identify and remove them. In this step, outliers were identified using statistical methods and were removed or replaced with more appropriate values. This process ensures that the analysis results are not affected by the presence of outliers in the dataset.

Standardized Data Formats: Inconsistent data formats can create confusion and distorted results. Therefore, it's crucial to standardize data formats to ensure consistency and eliminate any discrepancies that may have arisen from inconsistent data formats. In this step, the data formats were standardized, such as date formats or units of measurement, to make the data more uniform and easier to analyze.

Identify and Remove Irrelevant Data: Some data may not be relevant to the analysis or may be redundant with other data. Identifying and removing this data can simplify the analysis and reduce computational complexity. In this step, we have removed Employee Name: as our goal was to work with Employee ID, Stock No: which is the ID of each of every vehicle and our objective was to focus on the model, Check-in time, and Check-out time: we aimed to work with the Working Time that is the difference of the Check-in and Check-out time, Notes: This field is a self-defined where employees add any special notes. We also removed some stations which are outside of our main objective – Queue: queue station is where a vehicle sits for entering the next station, which means no employee works on that station, Detail: detail station is where employees clean the vehicle, which is dependent on the condition of the vehicle, not on the model of the vehicle, and so on.

Check for Data Integrity: Data integrity ensures that the data is accurate, complete, and consistent and conforms to any required data constraints or rules. In this step, we merged the before lunch working time and after lunch working time together. Let's say, we have an employee working at a station and they took a break and came back to work at the same station. Our initial data showed two different times for this case. Therefore, we summed up the working times to ensure that the data was accurate, complete, and consistent and conformed to any required data constraints or rules.

Validate Data Quality: Data quality ensures that the data conforms to any applicable quality standards, such as those established by industry or regulatory bodies. In this step, the data quality was validated to check that the data conformed to any applicable quality standards, such as those established by industry or regulatory bodies.

These data clean-up steps were taken to ensure that the data used for analysis was of high quality and suitable for the intended purpose. These steps are not exhaustive, and the data clean-up process may vary depending on the specific data and analysis requirements. However, the above-mentioned steps provide a basic framework that was used to ensure that the data is clean and suitable for analysis.

Final Dataset

After cleaning up the data using the aforementioned steps, the dataset was reduced to 5136 observations. The dataset was then randomly split into two subsets: a training set consisting of 80% of the observations and a test set consisting of the remaining 20% of the observations. The training set was used to train the machine learning model, while the test set was used to evaluate the model's performance.

This process of splitting the dataset into training and test sets is a common practice in machine learning to ensure that the model is not overfitting to the training data and can generalize well to new data. The use of 80% for training and 20% for testing is a commonly used split ratio, but other ratios can also be used depending on the size and complexity of the dataset.

It is important to note that the random splitting of the dataset into training and test sets ensures that the two sets are representative of the original dataset and that there is no bias in the selection of the samples for either set. Additionally, the split ensures that the evaluation of the model's performance on the test set is independent of the model's training, which further ensures the reliability of the evaluation results.

Model Training

As the objective of this project is to predict the time taken by workers to complete a task at a given station, it falls under the regression problem category. To find the best model for this problem, we used multiple popular regression algorithms, including Multiple Linear Regression, Polynomial Regression, Decision Tree Regression, Random Forest Regression, and Support Vector Regression. We used Python as the primary programming language for this project and implemented the aforementioned algorithms using the NumPy, pandas, scikit-learn, and matplotlib libraries.

To handle the categorical variables in the dataset, we applied encoding techniques. The Station variable was easily encoded using One Hot Encoding since it only had **[X]** distinct stations. However, the Vehicle variable had 789 unique entries, which means applying One Hot Encoding would create 789 new variables, leading to the curse of dimensionality problem. Therefore, we opted for Label Encoding for the Vehicle variable, which replaces the categories with their corresponding numeric values. This may introduce some bias in the model because of

the numeric order, but we had to use it to avoid the curse of dimensionality problem. Here is a code snippet on how we achieved it:

```
from sklearn.compose import ColumnTransformer
from sklearn.preprocessing import LabelEncoder, OneHotEncoder
# Encoding
# Using Label Encoder on the Vehicle Variable
vehicle_le = LabelEncoder()
X[:, 1] = vehicle_le.fit_transform(X[:, 1])

# Using One Hot Encoding on Station Variable
station_ct = ColumnTransformer(transformers=[('encoder',
OneHotEncoder(sparse_output=False), [2])],
                               remainder='passthrough')
X = np.array(station_ct.fit_transform(X))
```

Multiple Linear Regression

Multiple linear regression is a statistical method for predicting or explaining the relationship between a dependent variable and two or more independent variables. It estimates coefficients that represent the slope of the line between the dependent variable and each independent variable, and the intercept represents the expected value of the dependent variable when all independent variables are zero (Mukaka, 2012). We first trained the Multiple Linear Regression model on the dataset. After training, the model was evaluated using the train and test datasets. The following are the results we obtained:

- R- Squared Score (Train Data): 0.9157
- R- Squared Score (Test Data): 0.9067
- Mean Squared Error (Train Data): 5602.7883
- Mean Squared Error (Test Data): 5344.4440

The R-squared score for the train data indicates that our model explains around 91.57% of the variance in the data. Similarly, the R-squared score for the test data shows that our model generalizes well to new, unseen data with an explanation of around 90.67% of the variance. The mean squared error (MSE) is a measure of the average squared difference between the predicted and actual values. Our model has an MSE of 5602.7883 for the train data and 5344.4440 for the test data, indicating a good fit for the model. However, we need to evaluate the performance of other algorithms before selecting the best one for this problem.

Polynomial Regression

Polynomial regression is a statistical technique used to model nonlinear relationships between a dependent variable and an independent variable. It involves fitting a polynomial function to the data, which allows for more flexible and accurate modeling than linear regression (Yadav, Yadav, & Yadav, 2018). We used the Polynomial Features function from the scikit-learn library to generate polynomial features from the independent variables to fit a polynomial regression model.

After fitting polynomial regression models of degrees 2, 3, and 4, we obtained the following results from the test data:

- The R-squared score for degree 2: 0.923
- Mean Squared Error (MSE) for degree 2: 5127.105
- The R-squared score for degree 3: 0.910
- Mean Squared Error (MSE) for degree 3: 5997.196
- The R-squared score for degree 4: 0.913
- Mean Squared Error (MSE) for degree 4: 5755.820

These results indicate that the degree 2 polynomial regression model had the highest R-squared score and the lowest MSE, indicating it was the best fit for the data.

Support Vector Regression

Support vector regression (SVR) is a machine learning algorithm used for regression analysis. It uses a similar approach to support vector machines (SVMs) to find an optimal hyperplane but for regression tasks rather than classification. SVR is particularly useful for dealing with non-linear relationships and high-dimensional data (Tuncer, & Dogan, 2019). In our project, we applied SVR with different kernels, including Linear, Polynomial, RBF, and Sigmoid kernels, to find the best model for our dataset. However, the results were not as good as we expected, and all kernels failed to provide a good fit for our data. The R_2 scores for all kernels were negative, which indicates that our model performed worse than a horizontal line. The following are the R_2 scores obtained from our SVR models:

- Linear Kernel (Train Data): 0.15665772841495806
- Linear Kernel (Test Data): 0.145185440422538
- Polynomial Kernel (Train Data): -0.09666209486115429
- Polynomial Kernel (Test Data): -0.11852457140158124
- RBF Kernel (Train Data): -0.09787225657951093
- RBF Kernel (Test Data): -0.12018944126203235
- Sigmoid Kernel (Train Data): -0.10219550499897734
- Sigmoid Kernel (Test Data): -0.12562779156279125

These results suggest that SVR is not a suitable regression algorithm for our dataset.

Decision Tree Regression

A decision tree is a tree-like model used for making decisions in a hierarchical structure. It is a popular machine learning algorithm used for classification and regression tasks. In a decision tree, each internal node represents a test on an attribute, each branch represents an outcome of the test, and each leaf node represents a class label or a numerical value (Kumar, Singh, & Chauhan, 2020). For our dataset, we trained a Decision Tree Regression model and got the following results:

- R-squared Score (Train Data): 0.941
- R-squared Score (Test Data): 0.901
- MSE (Train Data): 3851.13
- MSE (Test Data): 6079.26

The high R-squared score on the train data indicates that the model fits the train data well, while the R-squared score on the test data suggests that the model generalizes well to unseen data. However, we observed a higher MSE on the test data, which means that the model is not perfect and could benefit from further improvements.

Random Forest Regression

Random forest regression is an ensemble learning method used for regression tasks. It involves constructing multiple decision trees at training time and outputting the mean prediction of the individual trees as the final prediction. The individual trees are trained on different subsets of the data and different subsets of the features, making random forest regression less susceptible to overfitting compared to a single decision tree (Cakir, & Taylan, 2020). In this project, we trained Random Forest Regression models with different numbers of trees and evaluated their

performance using R-Squared Score and Mean Squared Error (MSE). The following are the findings of our analysis:

Random Forest Regression:

# of Trees = 5 R-squared Score (Train Data) 0.938
# of Trees = 5 R-squared Score (Test Data) 0.896
# of Trees = 5 MSE (Train Data) 4118.76
# of Trees = 5 MSE (Test Data) 5927.90

# of Trees = 10 R-squared Score (Train Data) 0.939
# of Trees = 10 R-squared Score (Test Data) 0.898
# of Trees = 10 MSE (Train Data) 4031.77
# of Trees = 10 MSE (Test Data) 5814.49

# of Trees = 50 R-squared Score (Train Data) 0.941
# of Trees = 50 R-squared Score (Test Data) 0.901
# of Trees = 50 MSE (Train Data) 3945.25
# of Trees = 50 MSE (Test Data) 5668.16

# of Trees = 100 R-squared Score (Train Data) 0.941
# of Trees = 100 R-squared Score (Test Data) 0.903
# of Trees = 100 MSE (Train Data) 3938.36
# of Trees = 100 MSE (Test Data) 5561.13

# of Trees = 250 R-squared Score (Train Data) 0.941
# of Trees = 250 R-squared Score (Test Data) 0.903
# of Trees = 250 MSE (Train Data) 3934.58
# of Trees = 250 MSE (Test Data) 5543.00

# of Trees = 500 R-squared Score (Train Data) 0.941
# of Trees = 500 R-squared Score (Test Data) 0.903
# of Trees = 500 MSE (Train Data) 3931.54
# of Trees = 500 MSE (Test Data) 5532.78

As we can see from the results, the model's performance improves as the number of trees increases. The model with 500 trees achieved the highest R-Squared score of 0.903 and the lowest MSE of 5532.78 on the test data. Therefore, the Random Forest Regression model with 500 trees is a good fit for this dataset.

Model Evaluation

Here's a summary of the R-squared score and MSE values for each model:

MODEL	R_SQUARED	MSE
MULTIPLE LINEAR REGRESSION	0.906	5344.44
SUPPORT VECTOR REGRESSION	0.145	N/A
POLYNOMIAL REGRESSION	0.923	5127.105
DECISION TREE REGRESSION	0.901	6079.26
RANDOM FOREST REGRESSION	0.903	5532.78

Based on the comparison of the R-Squared and MSE values, we have chosen to move forward with the Polynomial Regression model. The Polynomial Regression model had the highest R-Squared value of 0.923, indicating a good fit to the data, and a relatively low MSE value of 5127.105, indicating a good level of accuracy in the predictions.

FUTURE IMPROVEMENTS

To improve the accuracy of the model in predicting individual employee time, data can be collected on individual employees by including age, experience, skills, training, and other relevant factors that may affect their productivity. The collected employee data is then transformed into features that the model can use. For example, create a characteristic to capture an employee's years of experience, or a characteristic to measure their skill level in a specific task. By incorporating employee data and considering individual employee differences, you can

improve the accuracy of your model in predicting employee productivity and the time required to complete tasks.

To improve the accuracy of the model in predicting the time required for different vehicle conditions, this can be done by including vehicle data such as the age and condition of the vehicle, additional features, options, and other relevant factors that may affect its efficiency. Also, collect more data on how employees perform with the vehicle under different conditions. Then use the additional data to develop new capabilities that capture the impact of different vehicle conditions on employee performance. For example, create a feature that measures the age of a vehicle or a feature that captures the level of vehicle maintenance. By collecting additional data and developing new features to capture the impact of different vehicle conditions on employee performance, the accuracy of the model in predicting the time required to complete a task can be improved.

CONCLUSION

In our research paper, we tackled the issue of time management inefficiencies in the automotive industry's production line. We proposed a time prediction model that can estimate the duration a worker will need to complete a task at a specific station. Our model serves as a valuable tool for workers in the subsequent station, allowing them to prepare for their tasks based on the predicted remaining time, resulting in increased efficiency, fewer delays, and errors.

We collected data from Freedom Motors USA (a wheelchair-accessible autonomous company). We applied several popular regression algorithms such as Multiple Linear Regression, Polynomial Regression, Decision Tree Regression, Random Forest Regression, and Support

Vector Regression, and assessed their performance based on the R-squared score and mean squared error (MSE).

Our study found that the Polynomial Regression algorithm, with a degree of 2, had the highest R-squared score and lowest MSE, indicating it was the most suitable fit for our dataset. We also tested the Decision Tree Regression algorithm, which had a higher R-squared score for the training data but slightly lower for the test data. Overall, our model achieved an R-squared score of around 92.30% for the test data, with an MSE of 5127.105, indicating a good fit.

However, we recognize that there is scope for further research to improve our model. For example, we only explored a limited number of regression algorithms and could investigate other algorithms or ensemble methods. We also used Label Encoding for the Vehicle variable, which may introduce bias in the model due to the numeric order. Future studies could explore more advanced encoding techniques or feature engineering to improve the model's performance.

In summary, our research demonstrates how machine learning can be utilized to tackle inefficiencies in production line management. By providing accurate time predictions, our model can help workers plan and execute their tasks more efficiently, leading to increased productivity and profitability. We hope that our study inspires further research and practical applications in this field.

REFERENCES

Bhat, P. V., & Kulkarni, S. S. (2020). Anomaly Detection in High-Dimensional Data: The Curse and Blessing of Dimensionality, *IEEE Transactions on Knowledge and Data Engineering*, *32*(8), 1516-1529. http://doi.org/10.1109/TKDE.2019.2947092.

Cakir, M., & Taylan, P. (2020). A comparison of machine learning algorithms for predicting the performance of photovoltaic cells. *International Journal of Energy Research, 44*(2), 1047-1061. https://doi.org/10.1002/er.4966.

Hair, J. F., Black, W. C., Babin, B. J., Anderson, R. E., & Tatham, R. L. (2019). *Multivariate data analysis, Cengage Learning, 8*, 189-193.

Khanna, N. (2018). Performance Evaluation of Prediction Models using Error Measures, *International Journal of Computer Applications, 182*(20), 10-14. http://doi.org/10.5120/ijca2018917875.

Kumar, N., Singh, A., & Chauhan, D. S. (2020). A novel approach for detection and diagnosis of diabetes using data mining techniques. *Computers in Biology and Medicine, 126,* https://doi.org/10.1016/j.compbiomed.2020.104026https://doi.org/10.1016/j.compbiomed.2020.104026.

Mukaka, M. M. (2012). A guide to appropriate use of correlation coefficient in medical research, *Malawi Medical Journal, 24*(3), 69-71. https://doi.org/10.4314/mmj.v24i3.12.

Singh, S., & Singh, V. (2018). A Comparative Study of Feature Encoding Techniques for Machine Learning. *Journal of Computer Science and Information Technology, 6*(1), 1-6. http://doi.org/10.12691/jcsit-6-1-1.

Torgo, M. (2021). Data Preprocessing for Machine Learning, *International Handbook of Data Science, 2*, 35-50. http://doi.org/10.1007/978-3-030-65960-0_3.

Tuncer, T., & Dogan, E. (2019). Predicting compressive strength of lightweight concrete using artificial intelligence. *Construction and Building Materials, 229*, 116893. https://doi.org/10.1016/j.conbuildmat.2019.116893.

Yadav, S., Yadav, R., & Yadav, M. (2018). Polynomial regression analysis of the impact of water quality parameters on the abundance of phytoplankton in an urban lake. *Journal of Environmental Management, 212*, 238-247. https://doi.org/10.1016/j.jenvman.2018.01.029.