**Using machine learning to predict demand for weather-sensitive products at Walmart Stores**

1. **Introduction & Background**
   1. **Describe the relationship between sales of product and weather.**

It seems obvious that the sales volume of products are influenced by many factors, such as holidays, prices, average per capita income, weathers and so on. Then, how can retailers matching their inventory levels with uncertain market demands to optimize their profits? Our group is interested in one certain factor which will affect the order quantity, the weather.

Non-catastrophic weather was proved to be an important determinant of demand for weather sensitive products, such as umbrella, heating oil, ice cream by many researchers [1]. In 2005, the coldest May in 22 years curbed spending on seasonal goods, such as garden supplies, outdoor furniture and air conditioners [2]. In 1998, William Daley [3], who is a former Secretary of Commerce of US, submitted a statement to Congress in which pointed out that at least $1 trillion in economic activities and 70% of the total amount of business activities in the United States are closely related to weather change.

* 1. **Previous researches of prediction for product sales based on weather conditions.**

In order to help retailers to optimize their inventory to avoid being out- of - stock or overstock during and after extreme weather, we need to predict the demand for commodities in certain weather condition as accurate as possible. However, how will weather affect retail business is very difficult to estimate and predict. Decision Tree algorithms is a classification/ regression algorithms that has been utilized on many previous study to predict the sales in real-life application under different conditions. It is considered one of the most competitive and efficient algorithms in terms of interpretability in prediction applications [7]. Thomassay and Fiordaliso [8] uses decision tree clustering to forecast the sale in apparel industry. In a study of forecasting supermarket sales, CART algorithms of decision tree have been used to successfully 70% of the variation in sales values [9]. Decision tree has become the base algorithms for may researchers to implement their experiments. However, since gradient boosting is another algorithms that is based on decision tree [10], [11], we are curious how will these two algorithms performs differently on our data set.

The previous teams that studied the sales behaviour using gradient boosting shows promising results. Korolev and Ruegg [12] uses XGBoost implementation of gradient boosting trees, which is the most popular framework in gradient boosting [13],  and Bayesian Optimization to further reduce the prediction error [12] in store sales prediction. Another team treats gradient boosting algorithms as a refinement of traditional classification tree algorithms. In a study that classifies remotely sensed data, by using gradient boosting algorithms, the accuracy was improved from 84% - 95% [14].

In our project, we take advantages of some prediction models studied by former researchers, trying to perform two different machine learning methods, Decision Tree and Gradient Boosting, to our dataset, comparing these two algorithms to find the optimal prediction method. And as result, we find that generally gradient boosting performs better than decision tree method.

* 1. **Data Source**

The dataset that we are investigating comes from a Kaggle competition for Walmart weather-sensitive products’ sales prediction. The datasets was in csv formats for us to download. There are three datasets was given: a training dataset to train your algorithms, a testing dataset to test the accuracy of your prediction result, a key data set as a complementary information and a weather dataset as a record of the weather information during the period of time of our interest. In these datasets, the units sold of 110 items across 45 Walmart stores and covered by 20 weather station was given. The detailed weather measurements across the day in 2 years were recorded along with the sale of the item at that day.

* 1. **Our team’s approach**

First of all we need to clean the data before we do further analysis. After that we performs decision tree algorithms and gradient boosting algorithms. One thing need to be noticed is that the item in our dataset is encoded differently between each store. So in order to keep our analysis result interpretable, we need to narrow our analysis into each store and uses different weather condition as features for machine learning. After calculating the accuracy for each algorithms, we investigate which algorithms performs better and dives into deep analysis to find out why.

1. **Data preprocessing & exploratory analysis**

The “train” dataset contains the units of a product sold in a store, the item number of the product, date it was sold and the store number in which this product was sold. The weather dataset contains the specific weather information in the above dates in the training dataset. “Station number”, 17 weather features and the date of recording these features was recorded in weather dataset. The corresponding station number for each store was recorded in the key dataset. Since we are here to investigate the relationship between weathers and the sale of the products, we need to match the weather record with the sale record together to analyze the sale on the day that a particular weather condition occur. First we merge the dataset “key” and the dataset “weather” together by station number to link the weather condition features and the store by station number. Second we merge this dataset with the train dataset by store number so we can match the sales and weather condition.

In order to predict the sales of weather sensitive products more accurately, several preprocessing steps were conducted on our original data. We observed that a lot of our data features contains “M” and “T” values. “M” indicates missing values and “T” indicates the value that is slightly bigger than 0 [15].Thus, we replace “M” with NAN and “T” with 0.001. Next, we replaced any “-” with a space and changed all numeric columns into float64 for further analysis. Besides, we removed attributes with missing values more than 50 percent.

After cleaning the dataset, we can first perform an exploratory analysis to have a general understanding about the cleaned data. Due to the different item number for same product in different stores, we separated the dataset by store numbers to study the sale under specific weather condition for each store. The three variables “tmin”, “tmax”, “tavg” are the temperature measurements. They are increasing drastically and then decrease in a much mild trend. The average dewpoint shows two peaks next to each other and wet bulb shows drastically increase at the beginning and stays high after a certain level.  In “cool” and “preciptotal”, 90% of the entries are 0. Resultant direction starts and ends with a high peak but the highest peak is at the middle of the distribution. Resultant wind speed and average wind speed are right-skewed bell-shaped curves. “Heat” and the number of units sold is decreasing overall but has a small peak right after the initial high value. Station pressure is left skewed and Sea level has a distribution that is a relatively symmetric bell-shaped curve. The sunset and sunrise time contains mostly 0 in it so we deleted these two features. The column “codesum” is the artificial encoding corresponding to the day’s weather features. There are 31 weather codes in total and the codesum of the day could be any combination of any numbers of codes from the 31 weather codes above. To study whether the complexity of the weather could change the sales, we created a new numeric column “weather” representing how many weather conditions occurs in a recorded day. From the histogram we can see that “units”, “resultspeed” and “heat” all follows a heavily right skewed distribution. This is a possible indication that there is an internal relationship between the units sold and the rest of the two weather factors. We need further analysis to know more about the underlying trend.

1. **Prediction analysis**
   1. **Using Decision Tree method to train dataset and test the prediction accuracy**

Decision tree is one of the most popular and fundamental methods in the field of machine learning for classification and regression, because of its easiness to understand the detailed procedure in each layer of the results by conducting a tree-shape visualization. The rough prototype model of decision tree analysis was developed by J. Ross Quinlan at the University of Sydney, which was based on the Concept Learning System (CLS) algorithm, also named as the Iterative Dichotomiser 3 (ID3). [16] Gradually, many scholars modified and improved the decision tree algorithm repeatedly. Finally, the decision tree algorithm we are using right now has two most essential advantages, which are high legibility and high efficiency. The model built by decision tree algorithm is a tree-like flowchart consists of many nodes, edges and leaves. Each node represents a test on a specific attribute, each edge represents an output from a specific range, and each leaf represents the classifier or prediction. The testing result for the entire process is from the root node through a certain number of edges and nodes, and finally ended at one of the leaves. The nodes can be of any combination of types (continuous, discrete but ordered, categorical, etc.). It is worth mentioning that to a multi-tree can always be accommodated as a larger binary tree by splitting the same attributes more than one time[17].

Instead of using the information gain as a dividing guide, which is calculated by the Decision tree classifier, the Decision tree regressor considered the mean squared error (MSE) reduction as its splitting rule (node impurity), which is equal to variance reduction. Suppose the dataset has a shape of rows and features, the MSE is defined as:

To get the best split in every step, after normalization, the attribute with the biggest variance from features, and all possible splitting value are transversely selected to be plugged in the criterion function (variance) for searching its minimum value below:

where are the subsets with a lower node impurity in its compared to original mean squared error, and is the updated corresponding output value of all input rows in each subset. [18] Here are the instructions how to generate the as:

,

, where is the number of in subset , and is the mean value of in .

So, the first splitting function is:

Then, we can calculate value the loss function (variance, sum of MSE) of this step as:

Then select the next pair dividing attribute and splitter (can be the same as last one), and repeat above procedures to generate , and

Repeat above steps until all stopping conditions at each nodes were met. For example, when finding a split which has a subset with the smallest variance (lowest impurity), this subset will not be further split in this tree model. Finally, after splits, will get the minimum value of the loss function of:

The following function is the last step to the leave nodes for the decision tree:

, where is all subsets after splits, contains all attributes used in splits.

After understanding the algorithm of Decision tree method, we use a regression package from scikit learn in python to train and predict product sales. And the results show that when changing the maximum depth of the tree, the prediction performance becomes better first, reaching the optimum value, which is 10, then after maximum depth with 20, it becomes worse.

* 1. **Using Gradient Boosting method to train dataset and test the prediction accuracy**

Introduction

Gradient boosting method is a machine learning technique which combines many weak models together from a single weak learner. In this case, we will use decision tree algorithm as the weak learner in an iterative loop.

Suppose we have a set of data , and an initial model to predict . Then we will have residuals

Then if we want to improve the prediction model, we can define a model such that

Equivalently,

It is generally infeasible to find a model which exactly fits the previous equations. However, we can fit a decision tree regression to the residuals, which means the new training data set used to build a model for will be  
Then, corrects by the pseudo residuals predicted by decision tree regression. So is a better model than . If the new model is not perfect, then we need add another decision tree regression.

How does this relate to gradient descent?

Gradient descent is an iterative method to minimize an objective function by moving to the opposite direction of the gradient with respect to the current point.

Specifically, if we have a loss function

Then we can get an optimization program

Then we take derivatives with respect to

which is the negative residual. Thus, we can interpret residuals as negative gradients. Also, we can interpret as

Thus, we actually update based on negative gradients. According to Cheng Li and other scholars who have done research about this, the concept of negative gradients is more general and useful than the concept of residuals. So, we will then use gradient descent to optimize our program [19].

Algorithm

The general goal for many prediction algorithms is to find an approximation to a function which minimize some specific loss function. As for the gradient boosting, it seeks an approximation which is a weighted sum of for , and each is from a decision tree regression:

In general, we will start with a model and update the model every time using greedy algorithm and gradient descent:

Also,

Then we will choose close to the negative gradient so that the calculation of coefficient is optimized based on the line search strategy.

Finally, the gradient boosting algorithm [20]:

|  |  |
| --- | --- |
| 1 |  |
| 2 | For k = 1 to N do: |
| 3 | where |
| 4 | Fit a decision tree to Residual\_ki, which means train it with data set |
| 5 |  |
| 6 |  |
| 7 | End for loop |
| 8 | Output . |

In summary, the main idea of the gradient boosting algorithm with the weak learner decision tree regression is that we optimize the coefficient each time in order to make the approximation of reduce as much deviation as possible, compared to . And the predicted deviation is based on decision tree regression fitting the current negative gradients.

We know that gradient boosting method combines multiple decision tree functions into a single strong learner function in an iterative backtracking procedure. And after every iteration, the loss function, mean squared error, is optimized to be the minimum. Therefore, we expect the prediction result to be better than a simple decision tree method. And the results show that when changing the maximum depth of the tree, the prediction performance becomes better first, reaching the optimum value, which is 5, then becomes worse.

1. **Results**

First, we need to understand that every time, the prediction test will ends up with slightly different result, causing by the random procedure in decision tree and gradient boosting methods. The following analysis was based on one of the tests.

In our project, we select product sales data in a certain store with nonzero item number to perform a decision tree machine learning, and a gradient boosting method based on decision tree. And we use several indicators to show prediction accuracy. First, R-squared is a statistical measure of how close the data are to the fitted regression line. It compares the fitness of the chosen model with the null hypothesis, a horizontal straight line. And the R2 can be negative, when the chosen model fits worse than the horizontal line. And generally the higher the R-squared, the better the model fits your data. Then, we use Spearman's rank correlation coefficient and Pearson correlation coefficient to measure the correlation of true product sales and predict ones. Spearman’s rank usually used as assessment of monotonic relationships whether linear or not, while Pearson's correlation assesses linear relationships.

In our results, for item 5 and 45, their R-squared, Spearman's rank and Pearson's correlation coefficient are both high in both decision tree and gradient boosting methods. We can assume these two product are weather sensitive product which have strong relationship with weather conditions and can be predict precisely. In the meanwhile, as shown in the prediction figure, the blue line and green line fit pretty well, further confirming our assumption.

Then, for item 86, when using decision tree method, its R-squared score is only 0.009. It seems that the fitted regression result is terrible. Then check the Spearman rank and Pearson’s correlation coefficient, which is 0.657 and 0.393, with p value 0.156 and 0.441. Because the p value is too large to reject the null hypothesis, we cannot be sure that this product is weather sensitive. In the meanwhile, in gradient boosting result, although these three indicator scores are different from decision tree. However, they are still higher than other items’ scores. Therefore, we conjecture this item to be potential weather sensitive product.

As for other 5 items, all three indicator scores are low both in decision tree and gradient boosting models. Therefore, they are weather non-sensitive product which cannot be predicted accurately by machine learning.

1. **Discussion**

Comparing with simple decision tree algorithm, gradient boosting method performs better when the maximum depth of the tree is less than 5. Then decision tree method performs better with increasing maximum depth. However, with further increasing the maximum depth, gradient boosting shows pretty well reliability, the accuracy does not change much, and it reaches the optimal result much faster than decision tree. The reason for these result may be that gradient boosting uses weighted averaging, and, therefore, gives a more reliable prediction if the overfitting is not the case. And, the gradient boosting most depends on many decision tree functions to minimize the mean squared error of residual values in each step. Therefore, the maximum depth may not have as much effect on the prediction result as decision tree.

1. **References**

[1] Fu, H., Dan, B., & Sun, X. (2014). Joint optimal pricing and ordering decisions for seasonal products with weather-sensitive demand. Discrete Dynamics in Nature and Society, 2014.

[2] L. Eaenest, “Same-store sales rise 2. 9% in May,” Los Angeles Times, June 2005, http://articles.latimes.com/2005/jun/03/ business/fi-retail3.

[3] S. Tiantian and G. Junjun, L. Vue, "The connotation and management on the weather sensitive demand," Business Research, pp. 70-7l, 2013.

[4] Taghizadeh, E. (2017). Utilizing artificial neural networks to predict demand for weather-sensitive products at retail stores. arXiv preprint arXiv:1711.08325.

[5] T. Choi, Y. Yu and K. Au, "A hybrid SARIMA wavelet transform method for sales forecasting," Decision Support Systems, vo1.51, pp. 130-140, 20 I I.

[6] Liu, Yue, Jianguo Zhao, and Junjun Gao. "Weather sensitive demand forecasting method based on SVR for shoes products." Knowledge and Smart Technology (KST), 2014 6th International Conference on. IEEE, 2014.

[7] Thomassey, S. (2010). Sales forecasts in clothing industry: The key success factor of the supply chain management. International Journal of Production Economics, 128(2), 470-483. doi:10.1016/j.ijpe.2010.07.018

[8] Thomassey, S., & Fiordaliso, A. (2006). A hybrid sales forecasting system based on clustering and decision trees. Decision Support Systems, 42(1), 408-421. doi:10.1016/j.dss.2005.01.008

[9] Silva, A. L., & Cardoso, M. G. (2005). Predicting supermarket sales: The use of regression trees. Journal of Targeting, Measurement and Analysis for Marketing, 13(3), 239-249. doi:10.1057/palgrave.jt.5740150

[10] Friedman, J. H. (2002). Stochastic gradient boosting. Computational Statistics & Data Analysis, 38(4), 367-378. doi:10.1016/s0167-9473(01)00065-2

[11] A Kaggle Master Explains Gradient Boosting. (2017, January 24). Retrieved December 08, 2017, from <http://blog.kaggle.com/2017/01/23/a-kaggle-master-explains-gradient-boosting/>

[12] Gradient Boosted Trees to Predict Store Sales. (n.d.). Retrieved December 8, 2017, from <http://www.bing.com/cr?IG=D4C135E0CE91428E94E58F408BE3A4C7&CID=1FB55BB2F5DD683229FF50E1F47269EE&rd=1&h=q5-c3QQd3o-NmPVtZPh4MhpArw91OIRrqmHG9Ba5Mfw&v=1&r=http%3a%2f%2fcs229.stanford.edu%2fproj2015%2f193_report.pdf&p=DevEx,5065.1>  (<https://pdfs.semanticscholar.org/41f9/754bb9cfda188066e84f1bc6929df88a4f96.pdf>)

[13] A Gentle Introduction to the Gradient Boosting Algorithm for Machine Learning. (2016, September 21). Retrieved December 08, 2017, from <https://machinelearningmastery.com/gentle-introduction-gradient-boosting-algorithm-machine-learning/>

[14] Lawrence, R. (2004). Classification of remotely sensed imagery using stochastic gradient boosting as a refinement of classification tree analysis. Remote Sensing of Environment, 90(3), 331-336. doi:10.1016/j.rse.2004.01.007

[15] (n.d.). Retrieved December 09, 2017, from <http://www.nws.noaa.gov/om/csd/info/NOWdata/FAQ.php>

[16] DMS Tutorial – Decision Trees. http://dms1.irb.hr/tutorial/tut\_dtrees.php

[17] Loh, Wei-Yin. 01/01/2011. Classification and regression trees, Wiley interdisciplinary reviews. Data mining and knowledge discovery, 1(1), 14 - 23-23. (ISSN: 1942-4787)

[18] Lee, Y. (1999). Data mining criteria for tree-based regression and classification (Order No. 9937579). Available from ProQuest Dissertations & Theses Global. (304532383). Retrieved from <http://proxy.library.georgetown.edu/login?url=https://search.proquest.com/docview/304532383?accountid=11091>

[19] Cheng Li. "A Gentle Introduction to Gradient Boosting". <http://www.chengli.io/tutorials/gradient_boosting.pdf>

[20] Hastie, T.; Tibshirani, R.; Friedman, J. H. (2009). "10. Boosting and Additive Trees". The Elements of Statistical Learning (2nd ed.). New York: Springer. pp. 337–384. ISBN 0-387-84857-6. <http://statweb.stanford.edu/~jhf/ftp/trebst.pdf>

1. **Appendix**

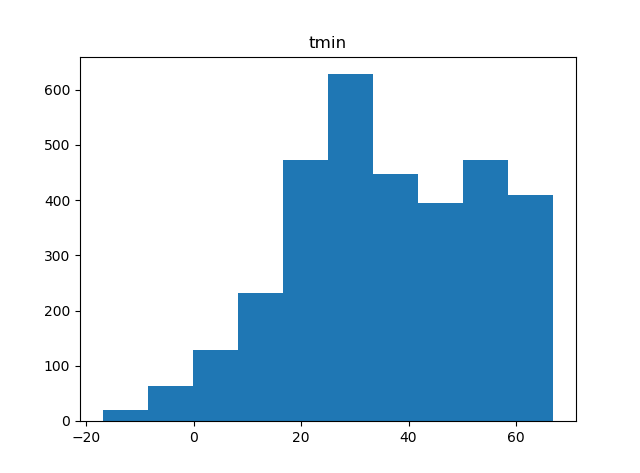
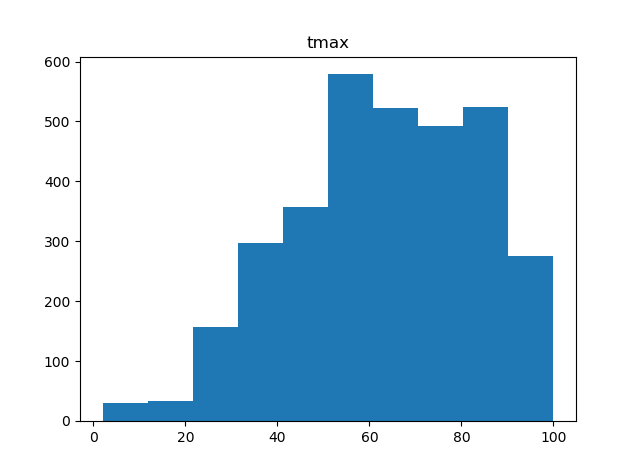
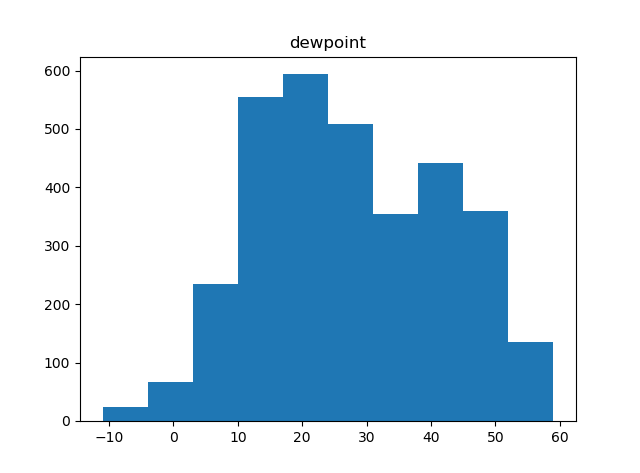
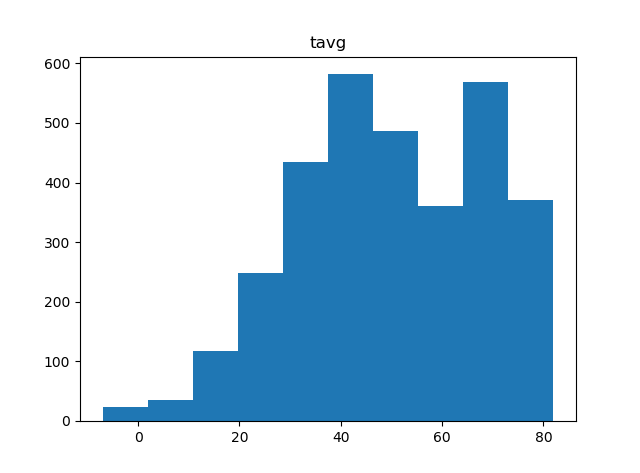
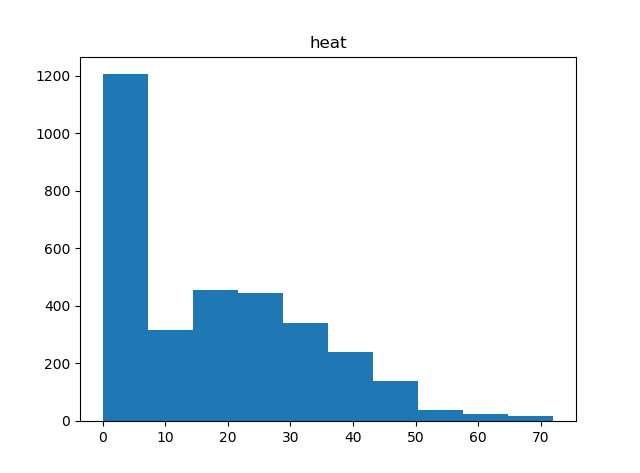
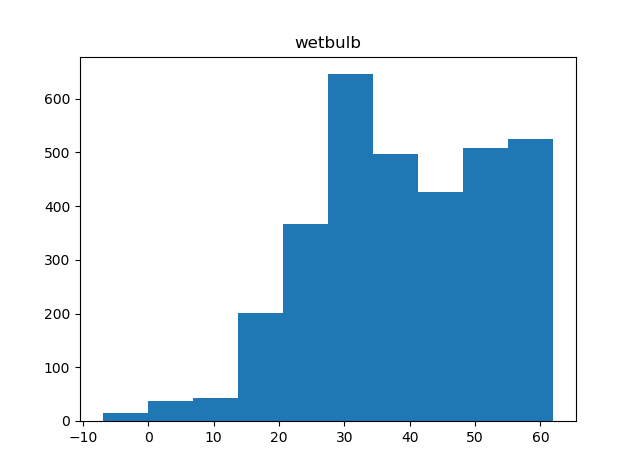
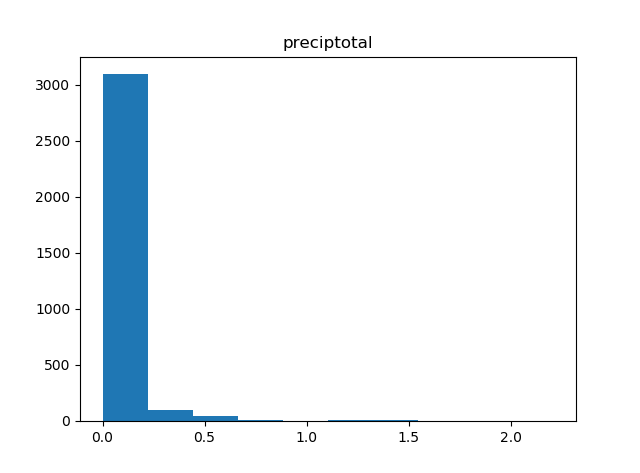
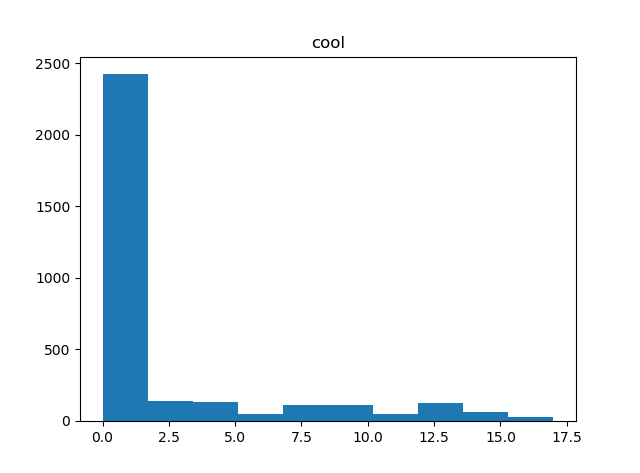
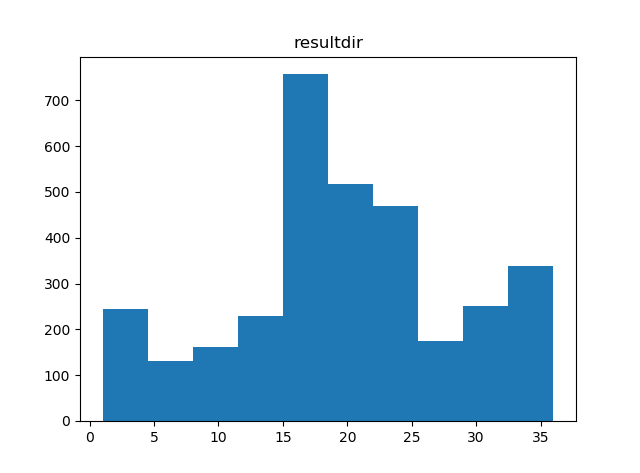
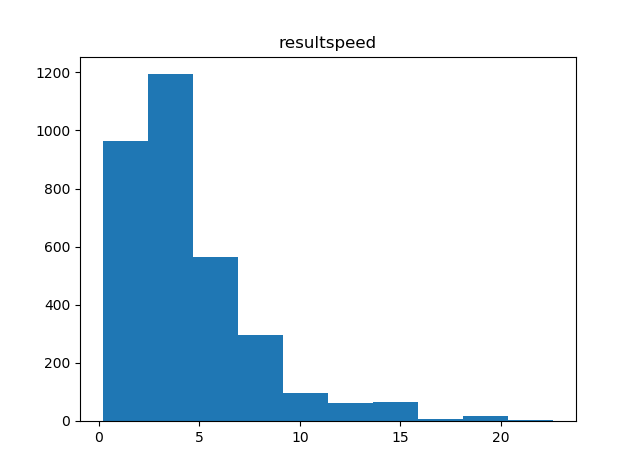
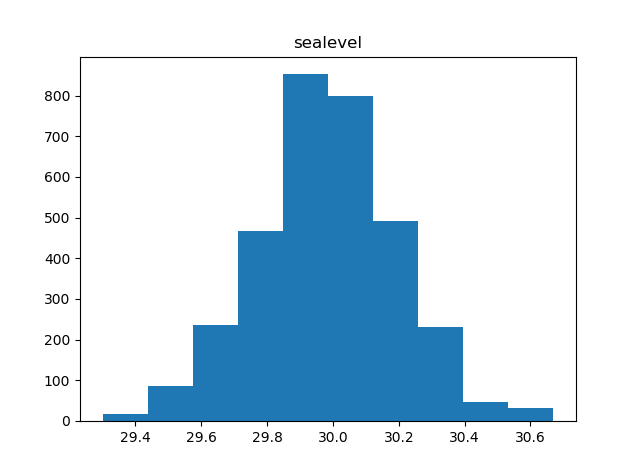
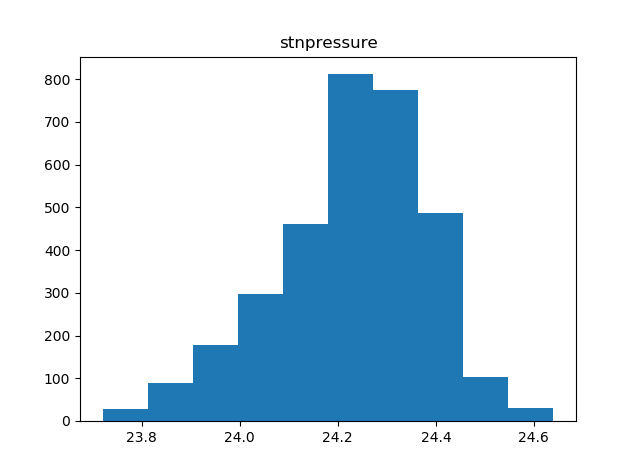
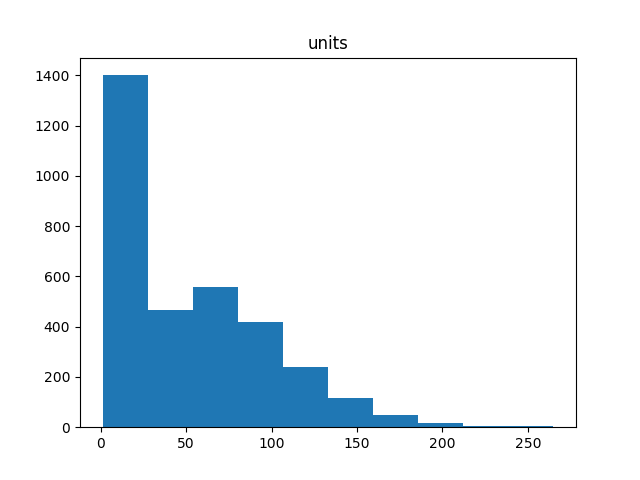
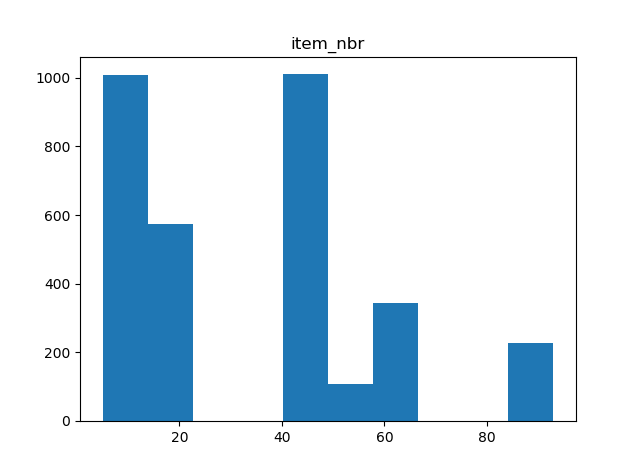
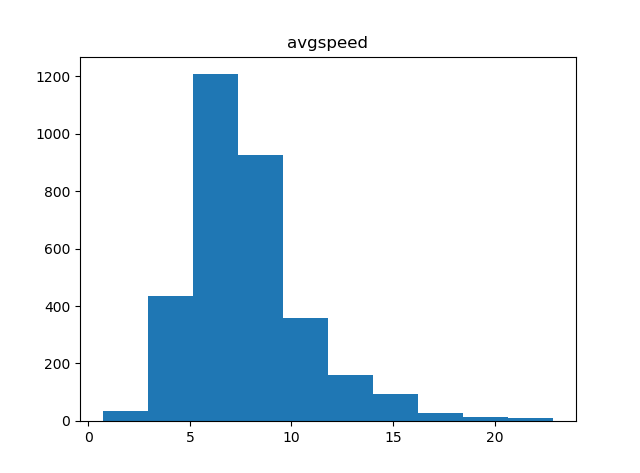
     

Figure 1. Distributions of numerical variables in cleaned dataset.

|  |  |  |  |
| --- | --- | --- | --- |
| Item ID | R-squared | Spearman rank | Pearson |
| 5 | 0.815 | 0.843 | 0.915 |
| 17 | -0.524 | 0.017 | 0.216 |
| 45 | 0.867 | 0.921 | 0.934 |
| 49 | -0.302 | 0.219 | 0.215 |
| 61 | -0. 161 | 0.030 | 0.079 |
| 93 | -1.427 | -0.233 | -0.314 |
| 86 | 0.009 | 0.657 | 0.393 |
| 15 | -2.297 | 0.201 | 0.190 |

Table 1. R-squared, Spearman's rank and Pearson's correlation coefficient score of prediction product sales for decision tree.

|  |  |  |  |
| --- | --- | --- | --- |
| Item ID | R-squared | Spearman rank | Pearson |
| 5 | 0.983 | 0.981 | 0.992 |
| 17 | -0.192 | 0.093 | 0.147 |
| 45 | 0.976 | 0.995 | 0.990 |
| 49 | -0.930 | -0.090 | 0.040 |
| 61 | 0.002 | 0.144 | 0.206 |
| 93 | -0.761 | -0.355 | -0.224 |
| 86 | 0.226 | 0.778 | 0.596 |
| 15 | -0.410 | 0.173 | 0.192 |

Table 2. R-squared, Spearman's rank and Pearson's correlation coefficient score of prediction product sales for gradient boosting.

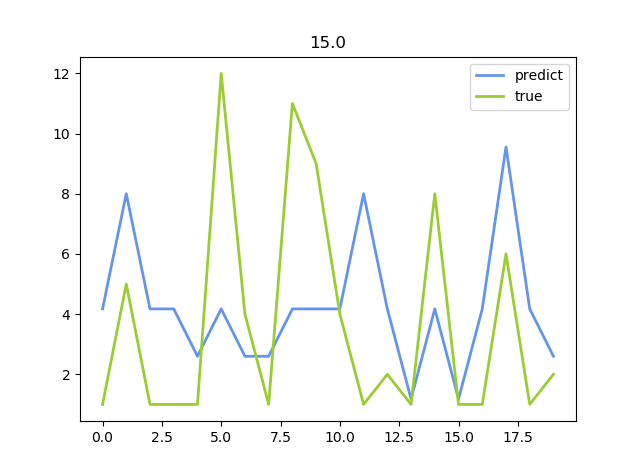
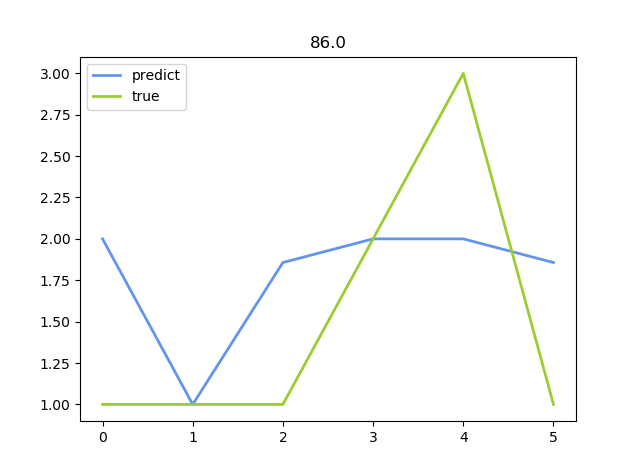
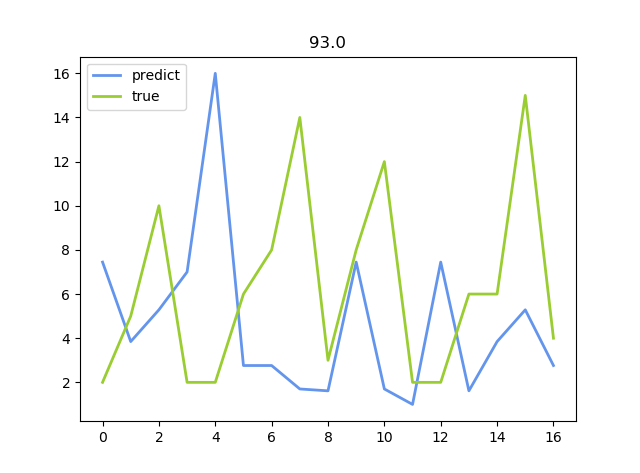
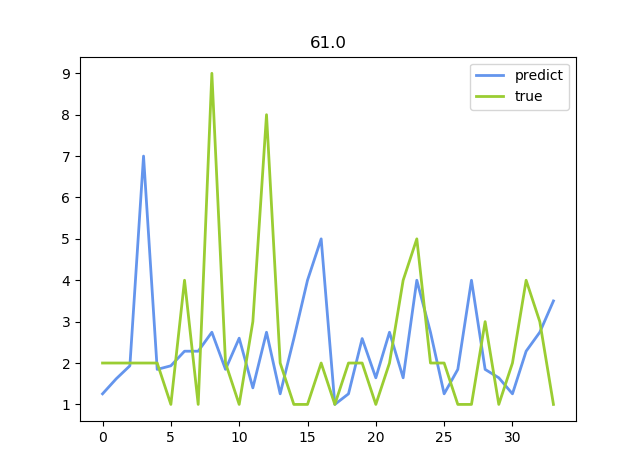
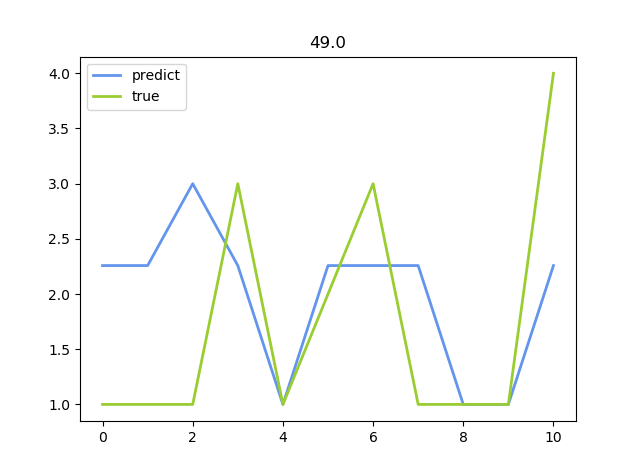
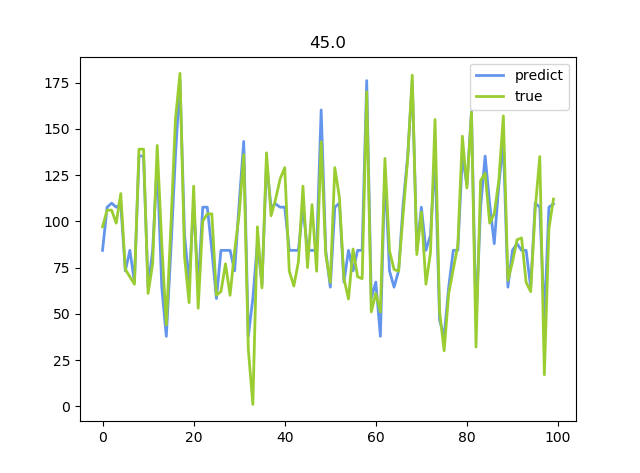
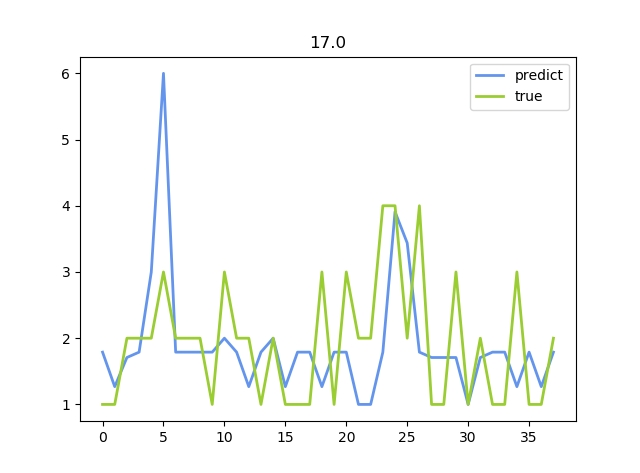
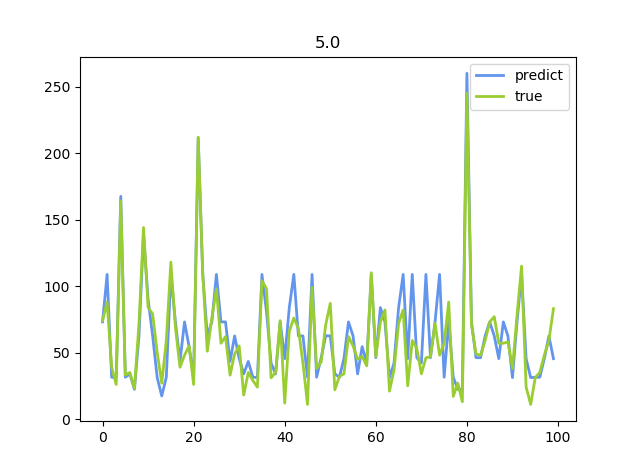


Figure 2. Predict and true product sales comparison in decision tree model.

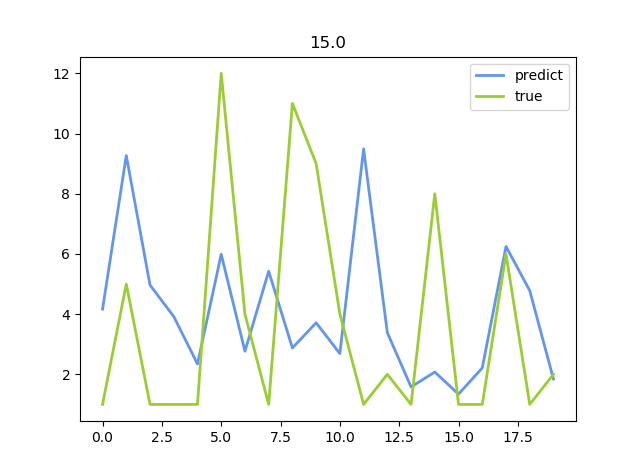
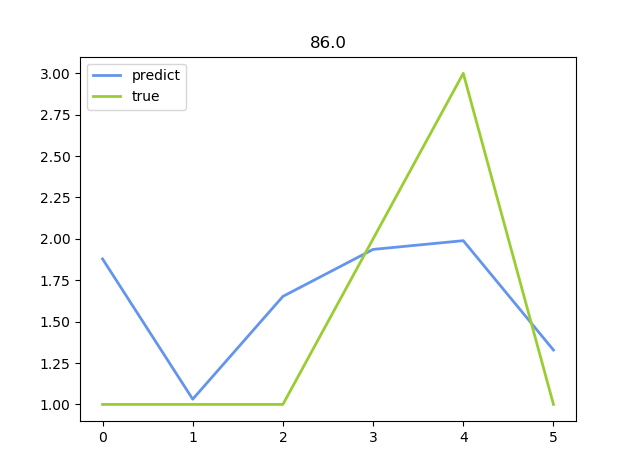
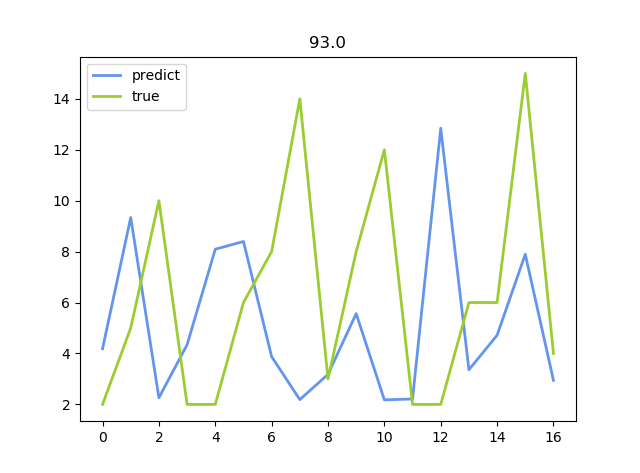
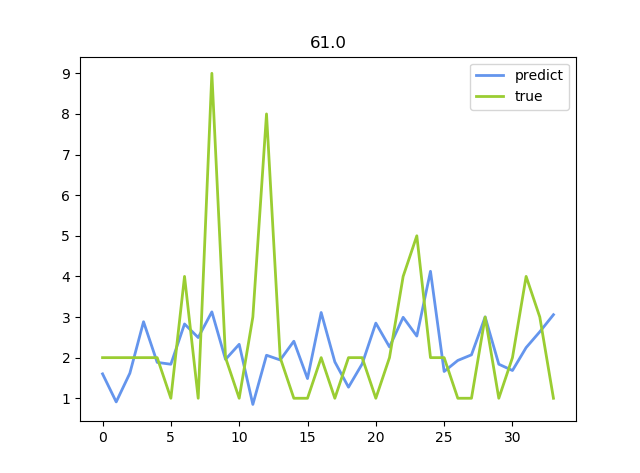
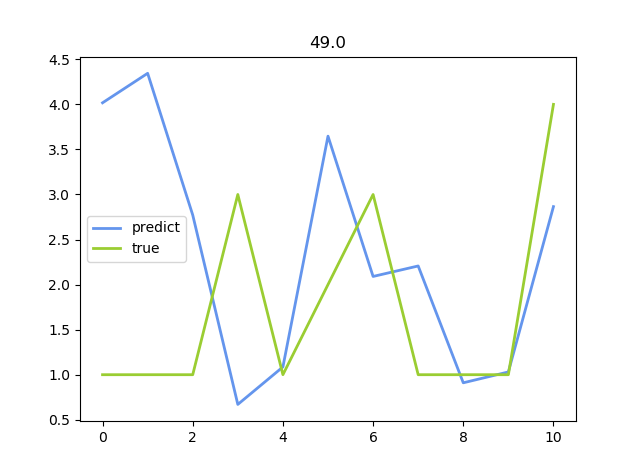
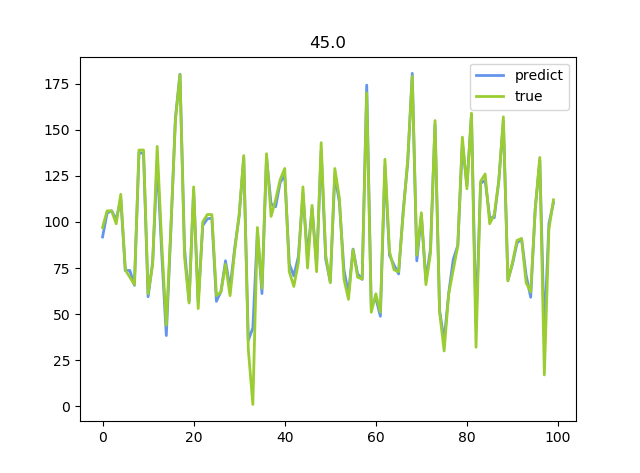
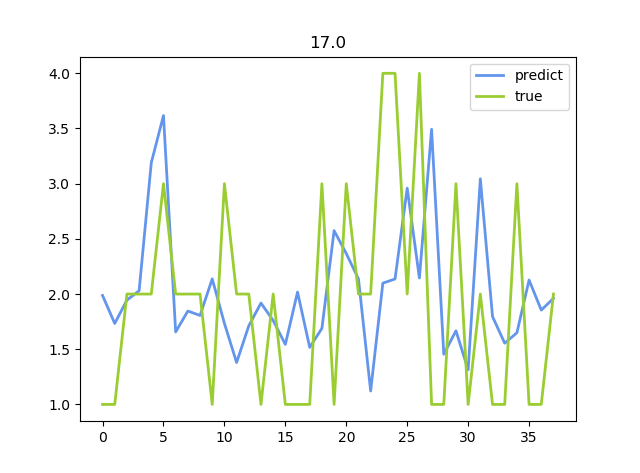
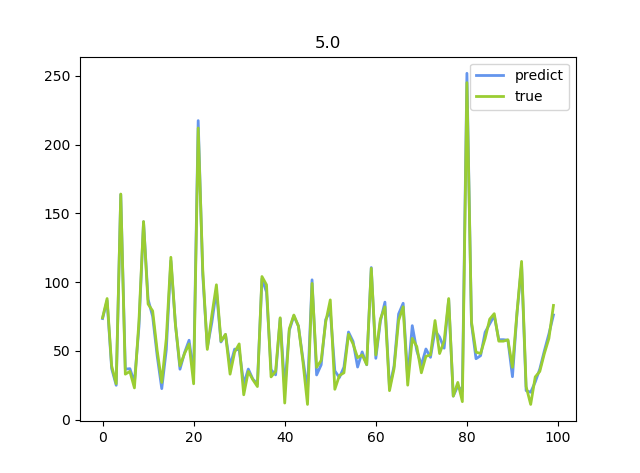


Figure 3. Predict and true product sales comparison in gradient boosting model.