Machine Learning Approaches to "Predict Future Sales" Kaggle Competition

Jocelyn McConnon

April 29th, 2020

Intro to Machine Learning Applications

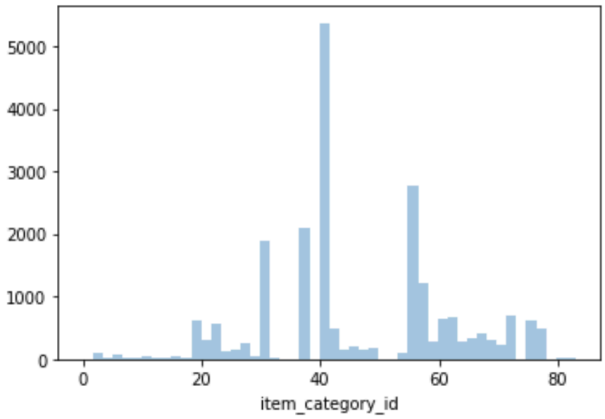
Lally School of Management
Rensselaer Polytechnic Institute
Troy, Ny 12180

Executive Summary

The following report will discuss the Kaggle prediction competition “Predict Future Sales”. This competition uses sales data from a Russian software firm named 1C Company. Key features of the data include when, where, and how many of their products are purchased. The objective is to predict the number of each product which will be sold by each store for the upcoming month. Submissions to this contest are scored using the root-mean-squared error. RMSE is computed by taking the square root of the average of the squared errors. Three public solutions will be discussed in this report, as well as an exploratory approach aimed at better understanding the features of the data. The modeling and methods of feature selection will be explained, and leaderboard scores are taken into consideration.

The dataset consists of 6 CSV files. All the files that compose the data contain no missing or repeating information. Therefore, there is no need to clean the data at this point. One file contains the 84 categories of items. The category name is given along with a unique ID number. It is worth noting here that the categories names are in Russian. One of the solutions discussed in this report translated the category names to English, and created a supercategory feature. Figure 1 contains a histogram which describes the distribution of items among the categories. The category named “Кино - DVD” which translates to English as “Movies - DVD” (ID number 40) contains close to 5,500 items. 1C Company has 22,170 items for sale and thus roughly 25% of their inventory is in DVD Movies.

Figure 1: Histogram of Number of Items vs. Item Category ID



Another file for the data used in this competition contains the item name, item ID number, and item category ID. There are 22,170 items, and like the category names, the item names are given in Russian. An example of an item name translate to English would be “100 Best romantic melodies (mp3-CD) (Digipack)”.

There is a file which contains the 60 shop names (In Russian), and corresponding shop ID number. The shop name inherently contains the city in which the shop is located. For example, “Krasnoyarsk TC "Vzletka Plaza”” is one of the shop names given (Krasnoyarsk is a city in Siberia, and Vzletka Plaza is where the shop is located). Figure 2 shows the distribution of sales amongst 1C Company’s 60 shops. Their Vzletka Plaza location, for example, is shop ID number 17 and experiences a moderate volume of sales.

Figure 2 is made using the sales data contained in the test file. The testing data file contains 6 columns and 10,34 rows. The features are date, consecutive month number of the date, shop ID, item ID, item price, and number of products sold. The number of products sold is the predicted value. In other words, given the

selected features, the objective is to predict the number of each item sold at each sore location in the upcoming month. Thus, the testing file contains three columns. One column contains shop IDs, another contains item IDs, and the last column contains a ID for this tuple.

In Figure 3, we can observe the distribution of sales among 22170 items. The range is relatively tight, meaning that 1C Company's products perform similarly. There are some products that sell better than others, but there are no products which do not sell at all.

Figure 2: Histogram of Number of Sales vs. Shop ID

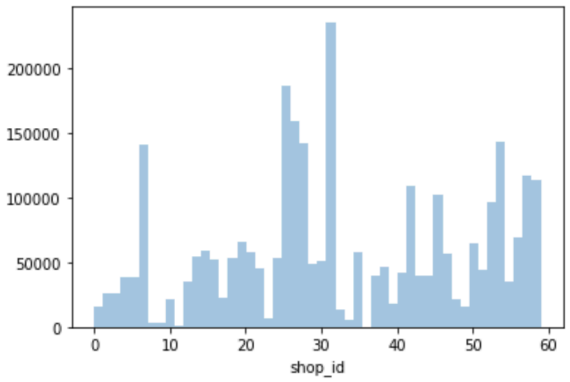
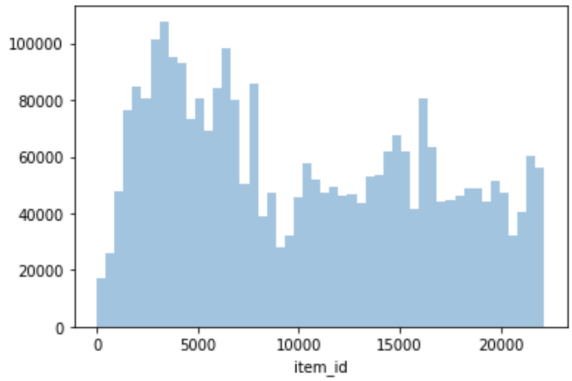


Figure 3: Histogram of Number of Sales vs. Item ID



Benchmarking of Other Solutions

The first solution considered is named "Feature Engineering/LightGBM/ Exploring Performance" and was submitted by Luke M. Specifically, version 135 will be discussed in the following section. This submission has a score of 0.85389, which is the lowest of the considered solutions and thus the most accurate method discussed in this report. The approach to building this model was predicated upon translating the Russian category, item, and shop names into English. The English translation of the data was one of the three inputs for this submission. Using the translated data, a supercategory of category names was added as a feature. For example,"Accessories" would become the supercategory of "Accessories - PS2". As part of the data preprocessing, outliers were removed from the training sales data. Shop IDs that do not appear in the testing set were also removed from the training data. Features such as day, month, and year were added using the date given in the training data. The training data was then grouped by month, shop ID, and item ID. K-means clustering was then performed. The best silhouette score was found using k = 7, and so shops were clustered into 7 clusters using their category sales as features. A feature was added representing the amount of time elapsed from the start of each month until an item was sold. This will be used in the model to determine the past performance of items. Other features added include (but are not limited to) sales of items by shop, sales within categories by shop, mean monthly sales by shop and by item,

average price of items and categories. Since so many features are added, the first two month blocks were discounted from the training data for the reason that the author felt some metrics may not be representative. This solution was unique in that it improved upon an initial model which was included in the inputs. A Light Gradient Boosting (LGB) model was used with a learning rate of 0.01. This model can be described as a series of trees in which the growth is determined by the best performing node. Light Gradient Boosting, which was designed by Microsoft, is known for its speed and is a popular predictive modeling algorithm in the word of machine learning (Serengil, 2018). The training RMSE was 0.62278, and the testing RSME was 0.740195.

The next solution to be considered is named “Feature engineering, xgboost” and was submitted by Denis Larionov. Specifically, discussed in this report is the 2658 version of the notebook, which has obtained a leaderboard score of 0.90684. To prepare the data, items with outlying prices and low sales were removed from the training dataset. Additional features were added by exploiting the naming structure of the shops and categories. Specifically, the city name contained in the name of the shop was added as a feature. Like in the first solution we examined, the supercategory of categories was also added as a feature. MeanEncoding was used on the categorical features to convert the names into numerical values. Trend features were added to describe the sales of items for the past 6 months, as well as the performance of shops in terns of sales. All features were used, including the newly created ones. A XGBoost model was used, and produced a training RMSE of 0.813137 and a testing RMSE of 0.905782. The XGBoost model is similar to the Light Gradient Boosting model which was previously discussed, and popular for its speed and performance. It is frequently referenced with respect to Kaggle Competitions since it is widely used on the platform (Brownlee, 2016).

The third solution I will consider is named “A beginner guide for sale data prediction” and was submitted by Triet

Figure 4: Performance summary

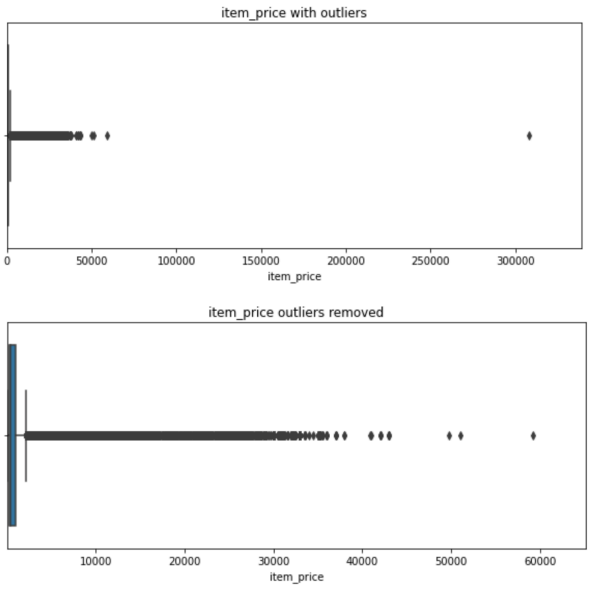
	Solution 1 Feature Engineering/ LightGBM/Exploring Performance	**Solution 2** Feature engineering, xgboost	**Solution 3** A beginner guide for sale data prediction
Score	0.85389	0.90684	1.25011
Feature selection	K-means clustering	Existing features were supplemented.	Existing feature were supplemented.
Model	LGB	XGBRegressor	LSTM
Runtime	6447.6 seconds	3976 seconds	3684.5 seconds
Training accuracy	0.622781	0.813137	3.85
Testing accuracy	0.740195	0.905782	4.29

Chau. Version 1171 will be referenced in this report, which had a Kaggle performance score of 1.25011. This score is the highest of the submissions we have discussed, meaning this method is the least accurate. This method uses a Long Short-Term Memory architecture. First, this method checks to see which shops are in the trainings and testing data. The model accounts for the spike in sales in the month of December, due to holiday shopping. The data is normalized, which is unique to this method. In fact, after normalizing the data there are several items for which the price is negative. The author uses LabelEncoder to transform the shop names into numerical values which are then vectorized. The model is built using sequence classification with Long Short Term Memory. This model is known for its ability to handle data which is usually technically difficult to train with, which stems from the nature of the memory blocks used in the model (Brownlee, 2017). After the model is evaluated, the training scores is 3.85 RMSE and the test score is 4.29 RMSE. These are high numbers and show that the model is very inaccurate. One possible explanation is that this solution did not add many features. The features added were minimal- only month and year. The other methods discussed went to great lengths to consider the addition of features that would strongly influence the prediction of items sold, and it will be shown in the exploratory approach that there were factors which had a high correlation with the prediction of sales, and had a significant impact on the model performance. This method also added missing item prices as the last price on record for the given store and item ID. This may have an had overall detrimental effect on the accuracy of the model.

Data Description and Initial Preprocessing

The remaining reading will cover an exploratory approach which utilizes feature engineering and compares the performance of the following three models: linear regression, random forest classifier, and decision tree classifier. The data preprocessing implemented in this approach is based on the most relevant features discussed in the previous solutions. Month, category, monthly sales, and average monthly sale price were the 4 features I chose to implement and analyze the importance for each model. Additionally, outlier data was removed. Figure 5 illustrates the importance of identifying and removing outlying data. One row was removed on the basis of outlying price, shown here in Figure 5. Two rows were removed from

Figure 5: Illustration of outlying item price



the training data on the basis of outlying number of products sold in a given day.

The month feature is a value from 1 to 12 which represents the month in which the item was sold. This was an important feature in the solutions discussed previously because the prediction is for the number of items sold in each store in the upcoming month. Therefore, by adding this feature the prediction models were able to more accurately predict the number of sales for the upcoming month.

Another important feature which was added is the category feature. This was done by splitting the item category name on "-" and adding the first element as the type feature. By exploiting the naming convention of the item categories, this feature adds more general information and acts as a supercategory. However, since this features is categorical and thus useless to the ML models, this exploratory approach used LabelEncoder to encode into a numerical value.

The monthly sales performance feature is computed by taking the sum of sales in a given date block number. Recall that the date block number is the sequential count of months. This is important since it enables the model to account for months in which there are more sales- like December.

The monthly sales performance was computed by taking the sum of the item prices for each date number block, and divided by the number of sales in the date block- which gives the average price for the sales in the given date block. This provides the model with some additional information regarding the frequency of large purchases, which would be associated with the item ID.

The sales performance indicators were crucial to the performance of the previous discussed solutions, and when added to the training data, had significantly impacted the performance of the models. The addition of these features reduced the RMSE of the models, as you can see in Figure 6. The training data was also standardized

Figure 6: Summary of Prediction Model Performance

Linear regression	Decision Tree	Random Forest	Monthly Performance Features	PCA	K-Means
2.35	2.37	2.37			
1.82	2.09	2.13	X		
2.36	2.37	2.37		X	
1.93	2.06	2.10	X	X	
1.84	2.09	2.13	X		X

during preprocessing. This did not have an impact on the RMSE, however it did decrease the magnitude of the importance of the features in the model. Finally, the training data was spilt into trying and testing data before the various models were fit.

PCA and K-Means were also performed and the results summarized in Figure 6. PCA was performed to see its impact on the models. Using 4 PCA components, the RMSE for linear regression increased .01 to 2.36. The

Figure 7: Feature Importance of Linear Regression without PCA and K-Means

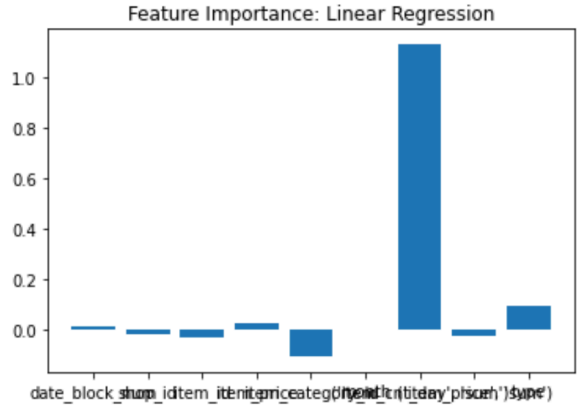
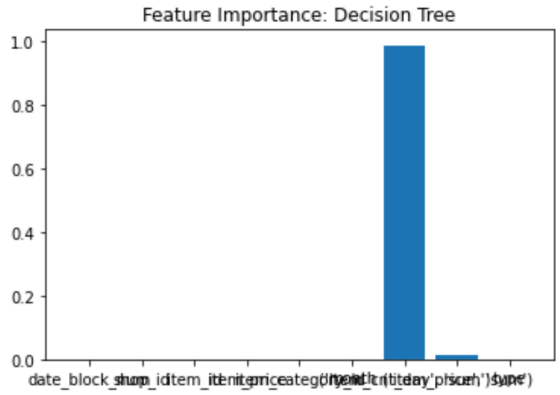


Figure 8: Feature Importance of Decision Tree without PCA and K-Means



RMSE did not change for decision tree or random forest. K-Means was performed using 7 clusters, and added as a feature to the data. This had a negligible effect of the RMSE.

Modeling

The summary of the three models RMSE is included in Figure 6. Random forest and decision tree had nearly identical performance, and both performed worse than the linear regression model. The feature importance, however, varied wildly.

For the linear regression model, category was the most important feature and type had a large negative importance before the monthly sales performance metrics were added. After they were added, the most important feature was the monthly sales count. The linear regression model had the lowest RMSE without PCA or K-Means, which was 1.82. This was the best performing prediction model developed in the exploratory approach, and the feature importance histogram is given in Figure 7.

The decision tree model used primarily the item price and item category as important features before adding the monthly sales metrics. After, the monthly sales count was by far the most important feature with the monthly average price having a slight influence as well- shown in Figure 8. Decision tree performed best with the monthly features

and PCA, and obtained an RMSE of 2.10.

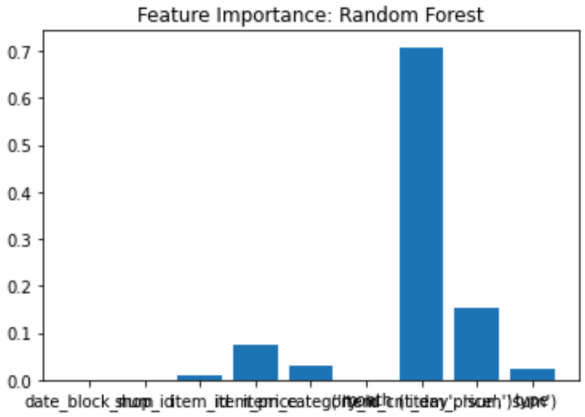
The random forest classifier used item id, item prices, and type as the most important features prior to adding the monthly sales performance metrics. After, monthly sales count was the predominant feature as seen in Figure 9, but many other features influenced the model slightly. Among all three models, the monthly sales count appears to be an important feature.

This exploratory approach utilizes the most important features to the other models discussed. These additional features had either a negative, negligible, or positive level of importance in the three models used. PCA and K-Means were performed in the data preprocessing to see their effects, but it seems like adding features has a more significant impact on the performance and reduces RMSE greatly.

The exploratory model was unable to outperform any of the previously discussed methods. However, it provided insight into the important aspects of the feature engineering, data preprocessing, and modeling. In this context, standardization, PCA, and K-Means did not significantly lower the RMSE but that does not mean that it's not worthwhile to do so.

The addition of the monthly sales performance metrics were crucial in reducing the RMSE since these features indicate which months are likely to experience a high volume of sales and

Figure 9: Feature Importance of Random Forest without PCA and K-Means



how well those months do are in terms of revenue. The previously discussed models corroborate this fact.

One reason random forest and decision tree may not be performing as well as linear regression is that the data in the testing set may fall outside of the range of the training data. As we saw when removing outliers, there was days which experienced a high number of sales. Perhaps several of the days in the upcoming month is unusually high and causes decision tree and random forest to become inaccurate (Mwiti, 2021).

To conclude, the exploratory approach provided insight into which features were most important for the Kaggle competition "Predict Future Sales" and why as well as how different prediction models utilize the same features to varying levels of success.

Reference List

"Predict Future Sales" Kaggle Competition. https://www.kaggle.com/c/competitive-data-science-predict-future-sales/overview

"Feature Engineering/LightGBM/Exploring Performance". Luke M. Version 135. https://www.kaggle.com/deepdivelm/feature-engineering-lightgbm-exploring-performance

"Feature engineering, xgboost". Denis Larionov. Version 2658.https://www.kaggle.com/dlarionov/feature-engineering-xgboost

"A beginner guide for sale data prediction". Triet Chau. Version 1171. https://www.kaggle.com/minhtriet/a-beginner-guide-for-sale-data-prediction

"A Gentle Introduction to LightGBM for Applied Machine Learning". Sefik Ilkin Serengil. October 13, 2018. https://sefiks.com/2018/10/13/a-gentle-introduction-to-lightgbm-for-applied-machine-learning/

"A Gentle Introduction to XGBoost for Applied Machine Learning". Jason Brownlee. August 17, 2016. https://machinelearningmastery.com/gentle-introduction-xgboost-applied-machine-learning/

"A Gentle Introduction to Long Short-Term Memory Networks by the Experts". Jason Brownlee. May 24, 2017. https://machinelearningmastery.com/gentle-introduction-long-short-term-memory-networks-experts/

"Sequence Classification with LSTM Recurrent Neural Networks in Python with Keras". Jason Brownlee. July 26, 2016. https://machinelearningmastery.com/sequence-classification-lstm-recurrent-neural-networks-python-keras/

"Random Forest Regression: When Does It Fail and Why?". Derrick Mwiti. April. 9th, 2021. https://neptune.ai/blog/random-forest-regression-when-does-it-fail-and-why