# Advances in Data Mining - Assignment 2

Predict future sales

**Group 75**

*Jesper Jäghagen (3649695), Sarp Önal (3296962), Rohan Pasricha (3599876)*

September 25, 2025

## 1 Introduction

In this assignment, we used various time series forecasting algorithms and data visualization techniques to accurately predict sales of a software company. We have implemented Prophet, Holt Winter's Exponential Smoothing and XGB regressor algorithms and used the results of the XGB for the Kaggle competition. We registered on the Kaggle platform with the team name as: *"Group75"*. Aim of this project is to predict total sales for every product and store in the next month by using the sales data of 34 months.

## 2 Data

Exploratory data analysis is performed as a first step to investigate the quality of the data and to find any interesting patterns in the data. The training set consists of approximately 2.9 million entries of sales records. One entry contains of the date of sale, shop id, item id, item price and sold quantity. The test set on the other hand holds 210,000 combinations of shop id's and item id's that will be predicted during November 2015. Records are available from January 2013 to October 2015, which corresponds to 34 months. There are a total of 60 shops, 84 item categories and 22,170 unique items.

The quality of the data is good because of three reasons. Firstly, there are no Nan-values to be handled. Secondly, the sales records are consistent, since there are transactions happening every single day during the entire training period. Thirdly, there is only one item price that is negative of the 22,170 items. The item with the negative price does not occur in the test set, it is therefore safe to remove this item from the training data. One confusing aspect of the data is that the records contain approximately 7,000 negative sold product counts. These will be interpreted as product returns from customers and will therefore be allowed in the data.

Lets now investigate how the sales pattern looks like during the period. We see in figures 1 and 2 that the total sales of the company follow a seasonal behavior, since there is a peek in sales that occur once every year. This observation is nothing unusual, because many gifts are being purchased around the Christmas holidays that are celebrated in Russia. We can also see that the trend of sold products in figure 2 declines with time, while the trend of total sales stays the same in figure 1. There are two possible explanations. The first is that a shift in frequently sold items has occurred, since more expensive merchandise require less sales to give the same return as cheaper products. The second explanation is that high inflation pushes the prices up, but makes people want to buy less goods. These two effects cancel out and create a zero trend in the total sales.

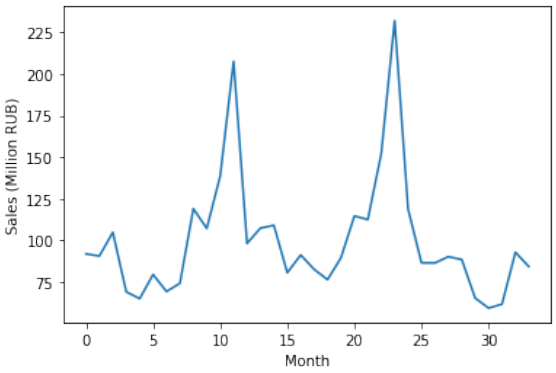



**Figure 1:** Total sales per month for 1C Company.

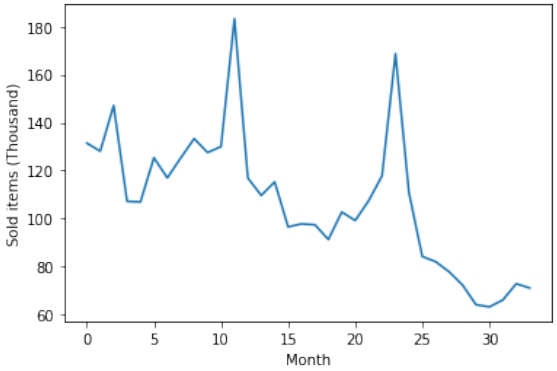



**Figure 2:** Total sold quantities per month for 1C Company.

The distribution of sales per shop and category are shown in figures 3 and 4. The shown id tags in the bar charts correspond to the shops and categories which sells more than the average shop respectively categories. We see that the sales of shops are quite evenly distributed with some exceptions. The categories on the other hand show that the majority of the sales are generated from a few high performing item groups. We can see the 5 best performing shops, categories and also most sold products in table 1. We see that C1 Company gets the most of its income from sales of video game related products. We see that the top product is the video game Grand Theft Auto V, which was released in September 2013. This could be a big contributing factor to the first peek in figure 1, since both occur at the same time. In other words are big video game releases a contributing factor to the total sales of the company. A final comment is that the best performing shops are located in densely populated areas in Russia such as Moscow and St. Petersbug.

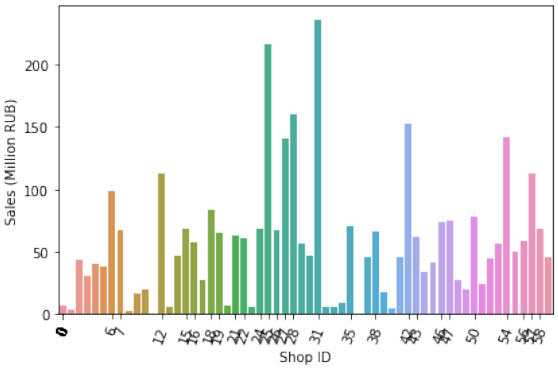



**Figure 3:** Total sales per 1C Company shop.

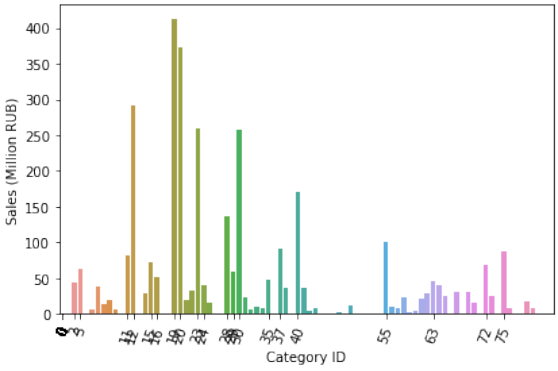



**Figure 4:** Total sales per category.

| | Rank | ID | Name | Sales (M RUB) |
|---|---|---|---|---|
| **Shops** | 1 | 31 | Moscow shopping center "Semenovsky" | 235 |
| | 2 | 25 | Moscow SEC "Atrium" | 216 |
| | 3 | 28 | Moscow shopping center "MEGA Teply Stan" II | 160 |
| | 4 | 42 | St. Petersburg TC "Nevsky Center" | 152 |
| | 5 | 54 | Khimki shopping center "Mega" | 142 |
| **Categories** | 1 | 19 | Games - PS3 | 412 |
| | 2 | 20 | Games - PS4 | 373 |
| | 3 | 12 | Game consoles - PS4 | 292 |
| | 4 | 23 | Games - XBOX 360 | 260 |
| | 5 | 30 | PC Games - Standard Editions | 258 |
| **Items** | 1 | 6675 | Sony PlayStation 4 (500 Gb) | 219 |
| | 2 | 3732 | Grand Theft Auto V [PS3] | 43.6 |
| | 3 | 13443 | Bundle Sony PS4 (500 Gb) + GTA V game | 34.3 |
| | 4 | 3734 | Grand Theft Auto V [Xbox 360] | 31.1 |
| | 5 | 3733 | Grand Theft Auto V [PS4] | 22.3 |

**Table 1:** The top 5 selling shops, categories and items. Each row shows the ranking, numeric id, English name and sales.

As the last part of the data analysis, we see how the prices and sell frequencies are distributed, in figures 5 and 6. 1C Company has a wide range of products, but only a few percentage are top sellers. We see this in figure 5 in form of a power-law distribution. We see that the the most common pricing is below 4000 RUB in figure 6 and that there exists very few items with high pricing. The most expensive item costs 531,434 RUB.

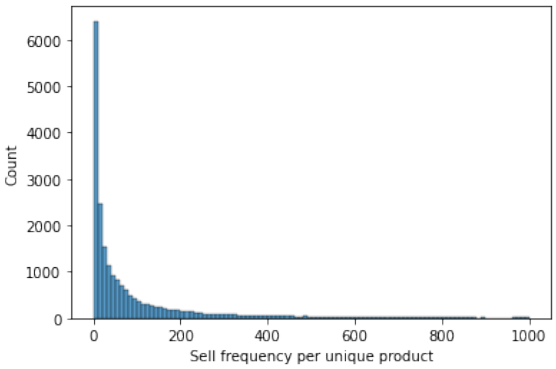



**Figure 5:** Total sell frequency for different products.

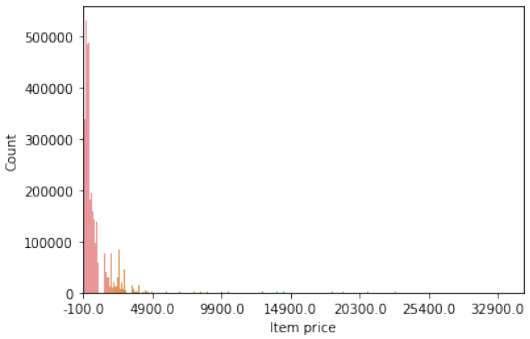



**Figure 6:** Distribution of prices.

# 3 Approaches

Three different machine learning approaches are used to predict the sales of the month of November 2015. Namely, the Prophet algorithm, Holt Winter's exponential smoothing and XGBoost, which are further described below. XGB regressor algorithm is used for the Kaggle competition.

## 3.1 Prophet Algorithm

The Prophet algorithm is a time series prediction model created by Meta. Prophet is tailor made to forecast data with strong seasonal behavior. For example can prophet take holidays and other special events into account when making a prediction. Prophet also automatically detect trend change points, points where the time series abruptly changes behavior. One negative aspect of Prophet

is that it does not support item-by-item predictions, which makes it a sub-optimal choice for the Kaggle challenge. To get a Kaggle prediction, a Prophet model must be trained for each item. One fit and predict takes approximately 1 second, which adds up to a total of 58 hours of run time for all test set entries.

However, Prophet is good when it predicts the total amount of sold products. Figure 7 shows the original time series data (solid line) and the fitted and predicted time series (dashed line). The time series was split 80-20 into training and test data. The Root Mean Squared Error was calculated for the test data which resulted in $RMSE = 16928.1$, which is an okay performance considering the relative axis scale.

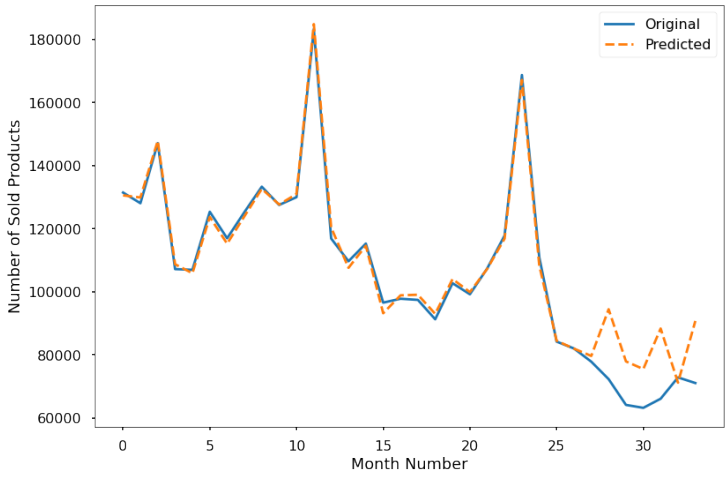



**Figure 7:** Monthly forecast by of total number of sold products by the Prophet method.

## 3.2 Holt Winter's Exponential Smoothing

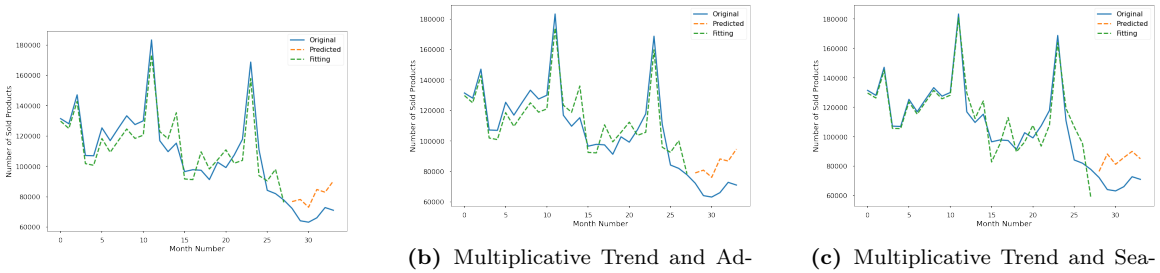
We have implemented the Holt Winter's exponential smoothing (HWES) algorithm since it predicts the next time step as exponentially weighted linear function of prior time steps and also considers the trends and seasons. By using exponentially decreasing weights, it increases the importance of recent data compared to older data. And it is important that the algorithm takes seasons into account since there might be more sales in some months (like an increase sale in Christmas). *Seasonal_periods* is chosen to be 12 as there are 12 months in a yearly seasonal structure. While having the advantages of detecting trends and seasonal behaviors; HWES doesn't support item by item predictions, which causes it to be a non-optimal choice for the Kaggle competition.

However, HWES is good for predicting the total amount of sold products. Original data is divided into 80-20 train-test split. Original data consists of 34 months, we have used first 28 months for training and last 6 months for test. Experiments were conducted on 3 different HWES architecture: additive trend and seasoning, multiplicative trend additive seasoning, multiplicative trend and seasoning. Figure 8 shows the original data, fitted and predictions for all of the HWES models experimented with.

It is possible to observe from the table 2 that, HWES with additive trend and seasoning has the best performance.

| Add and Add | Mul and Add | Mul and Mul |
|---|---|---|
| 13781.3 | 17006.6 | 17376.9 |

**Table 2:** RMSE of 3 HWES models




**(a)** Additive Trend and Season

**(b)** Multiplicative Trend and Additive Season

**(c)** Multiplicative Trend and Season

**Figure 8:** Monthly forecast of total number of sold products by HWES

## 3.3 XGB Regressor

For our 3rd algorithm, we have implemented the eXtreme Gradient Boosting Regressor (XGBRegressor).

It is an ensemble learning method that will create a final model by combining several weaker models. XGBRegressor uses gradient descent and boosting. The Boosting method adopts the iterative procedure to change the distribution of training data by focusing more on previously misclassified records when building the base learners. The models are added sequentially until no further improvement can be made. A gradient descent algorithm is used to minimize the loss. In gradient boosting, where the predictions of multiple models are combined, the gradient is used to optimize the boosted model prediction in each round.

The data is split into training and testing data (0.8, 0.2).

We did some hyperparameter tuning to obtain the final results displayed in the figure below. When using the default parameters of XGBRegressor, we were able to get a test accuracy of 0.68 and an MSE score of 47.19 however when we change the parameters (objective='reg:linear', max_depth=10, n_estimators=1000,min_child_weight=0.5, colsample_bytree=0.8, subsample=0.8, eta=0.1) we were able to achieve a test accuracy of 0.886, MSE score of 18.71 and RMSE score of 4.325.

Figure 9 compares the predictions made by the final model and the actual values of monthly sales of each item grouped by month and shop ID. We can see that some of the predicted values are negative even though we have removed the negative values from the training set so this was unexpected and further work may be able to fix this however we are satisfied with the overall evaluation metrics of the created final model.

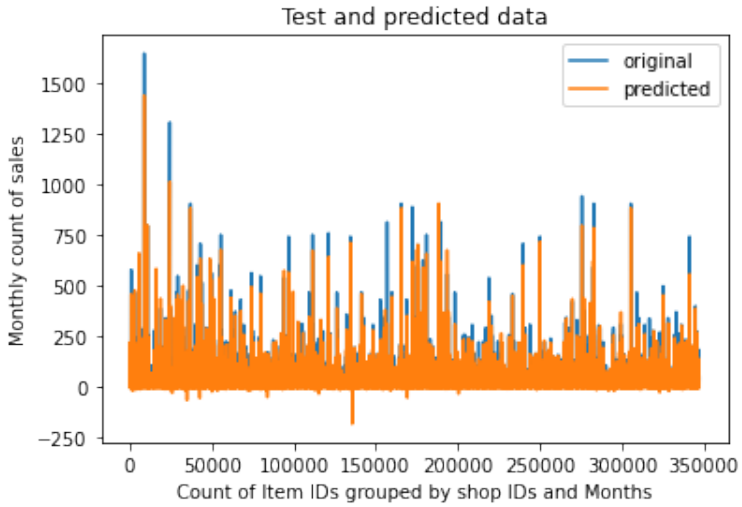



**Figure 9:** Forecast of total number of each product sold grouped by month and shopID

# 4 Discussion

We noticed from our experiments that, while Prophet and HWES are good for predicting the total amount of sold products, they are not the optimal choice for the Kaggle challenge since they need to be trained for each item separately which causes a really high run time for the entire dataset. On the otherhand, XGboost can be used optimally for the Kaggle challenge as it can run over the entire dataset very fast. We also observed that Prophet algorithm and HWES with multiplicative trend have similar RMSE score, while HWES with additive trend and seasoning out-performs them. We can also observe from figures 8 and 7 that Prophet makes better predictions in the training set but has lesser performance in test set compared to HWES with additive trend and seasons. This may imply Prophet is overfitting the data and HWES has better generalization abilities.

We can observe from the figure 7 that total number of sales increases a lot seasonally at the end of each year. This might be related to Christmas and black friday sales. We see that both the Prophet and HWES can learn this seasonal change in the training set, and it looks like they predict an increase at the end of the year 2015 as we can see from the November 2015. Also the Russian economy was not stable in 2015, which may interfere with the results and makes it harder to predict well in the year 2015.

# 5 Conclusions

XGBRegressor seems to be the best performing model out of the 3 we have tried to implement on this dataset as it performs fast and produces a relatively high test accuracy and low RMSE score. We can infer from the visualization in figure 9 that the final model does seem to perform quite well on the test data however there is room for some improvement as the model occasionally predicts negative monthly count of sales which was unexpected. We were able to successfully predict the count of sales grouped by Month and shopID of each itemID in the given dataset.

We used the XGboost algorithm for the Kaggle competition and got RMSE score **1,05171** with the team name of **Group75**.