

Predict store sales

Nausester Sebastian & Feierabend Jonas, 14.11.2023

DESCRIBE THE PROBLEM

SCOPE

The project aims to enhance store sales predictions for the near future by considering various factors, such as current oil prices and holidays. This predictive capability assists in planning the timing and quantities for store restocking. In the absence of machine learning, a traditional approach using Excel sheets with average sales over past years would be used.

Post-deployment, ongoing monitoring of the same metrics will allow for a comparison between predictions and actual data. The goal is to achieve estimates for the upcoming 7 days within a range of $\pm 10\%$. This range can't be accomplished for single days because even a single family can induce spikes of this size in small stores. Although it helps to have estimates for single days to get a rough estimate of how many people will come in one day.

The new data will be aggregated across different stores and collected through a centralized system. This data will then be provided through an API. Regular training of our neural network, facilitated by batch training every few days due to low resource requirements, allows for the deployment of updated models. These models become accessible to managers responsible for supply organization. The API is designed to separate data aggregation and storage from model training. Changes in data aggregation will not affect the API, and similarly, modifications in the model or training method won't impact the data aggregation process.

Stakeholders in this project include the heads of the stores and the parent company. The project milestones include

- analyzing influential factors
- recording data
- creating and testing the model
- using the model for predictions while continuously improving it with upcoming data.

The project incorporates APIs to obtain external data, such as oil prices and inflation, creating a dataset daily for each store.

METRICS

We don't require precise day-specific sales figures but rather need the ability to optimize stock levels across different stores. To achieve this, an estimated accuracy within the range of approximately $\pm 10\%$ is sufficient, aligning with the buffer capacity of our stores. The accuracy on one day won't be measured because of the earlier mentioned unforeseeable spikes.

The desired level of accuracy is crucial for determining the quantity of products to ship to each specific store. Additionally, we aim to provide sales estimates for a 7-day period in advance.

Therefore, our prediction horizon needs to extend beyond a single day, as forecasting 8 days upfront could potentially degrade the accuracy of predictions. Given our operational scale of 50 stores across 33 branches, throughput is not expected to be a limiting factor, and the computational load is well within manageable limits.

DATA

We are using the data from the Kaggle competition Store Sales - Time Series Forecasting. The data is labeled on a daily basis, with almost a tuple for every date. For future model accuracy, it's crucial to continuously update the model with the latest data, considering factors like inflation or potential increases in oil prices. The switch to renewable energies will probably greatly reduce the influence of oil prices in the long term, while electricity prices could gain in influence. It is therefore important to evaluate the (negative) influence of old data. Also holidays have an impact on the sales numbers, although we don't look at them individually and just look at the development over years or months.

Our sales predictions are made separately for each product type and each supermarket due to their significant impact on the overall forecasting results.

The dataset spans four years, providing more than enough historical data. However, it's estimated that two years should be sufficient to analyze cyclic patterns, such as increased purchases before Christmas. Older data may not accurately represent the current market conditions, and consideration is given to either discarding old data or assigning lower importance during model training.

Ground truth validation is performed using data from different stores reporting to the headquarters. If a store reports significantly higher or lower sales than usual, data analysts investigate to determine if it's a collection error or a valid number. This data is also leveraged to monitor the behavior of different customer groups. While it's not feasible to scrutinize individual behavior due to privacy and ethical concerns, the aggregated data still provides valuable insights.

Initially considering data scaling and various cleaning techniques, it appears that sktime, the chosen tool, may not require such preprocessing. Raw data provision is deemed sufficient. There is a suggestion to flatten out significant spikes in the data, as they may not necessarily be reliable indicators of future behavior. Instances of such spikes should be investigated, and reasons for these specific outbursts should be monitored. It's noted that sktime does not heavily consider spikes that are not in the last days of the dataset.

MODELING

Our initial approach involved treating the forecasting problem similarly to a regression problem, incorporating lag features to consider not only the current day but also preceding days. While this method exhibited success on the training data, it fell short in performance on Kaggle.

Subsequently, we explored alternative solutions and encountered challenges using the Darts library for the Kaggle competition. Despite its potential utility for real-life data analysts with its ability to produce insightful graphs and user-friendly interface, the obstacle for us lay in converting the data from different TimeSeries back to a .csv format.

Our final solution pivoted to sktime, a Python library akin to scikit-learn but designed specifically for time forecasting. The NaiveForecaster (class in sktime), despite relying solely on sales from previous days and not utilizing current day data, demonstrated effectiveness in predicting store sales. This method proves suitable for capturing peaks associated with holidays, weekdays, and seasons but may fall short in predicting peaks induced by other factors such as oil prices.

In retrospect, our extensive data investigation may have seemed redundant, as the NaiveForecaster outperformed our initial regression-based approach. Despite witnessing more advanced solutions with better performance in the Kaggle competition, often involving complex models, including deep neural networks, our thorough data investigation remains a valuable foundation for potential future performance improvements.

DEPLOYMENT

In the case of task 2, we have deployed the model on HuggingFaces. There you have the possibility to enter data and get a forecast for it. Thinking out of the box, meaning outside this assignment, we would proceed as described below.

The models should be bundled with a software package equipped with key functionalities. It should deliver predictions for forthcoming days, potentially extending to weeks and months, enabling the market leader to make informed procurement decisions. Additionally, the software should feature an automated process to gather real-time information, enhancing the model's continuous improvement. An automatic calibration mechanism is crucial to detect if the current model no longer aligns with the evolving data, prompting adaptability. The intention is to retrain the model daily, leveraging sktime's performance capabilities. Over the next few months, our goal is to support managers in using the model alongside their traditional methods, monitoring its impact. After this period, we plan to reassess whether the system has positively influenced outcomes, needs improvement, or should be discarded.

USAGE OF GENERATIVE AI

We utilized ChatGPT for our assignment in two primary ways. First, we explored its capabilities in code generation. Initially, we experimented by asking ChatGPT to provide a solution for our forecasting problem. However, the results were suboptimal, as it treated the forecasting task similarly to a regression task, resulting in poor performance. This highlighted the importance of evaluating solutions and exploring alternative approaches, leading us to choose sktime, a method not initially recommended by the AI but ultimately more effective.

ChatGPT has proven invaluable in streamlining simple yet repetitive tasks. For instance, when dealing with straightforward assignments like trying out different scikit-learn models or using

Matplotlib, we can swiftly request ChatGPT to generate the code, make necessary variable adjustments, and quickly complete the task. This eliminates the need for extensive searches for the right imports or debugging common errors, making the process more efficient.

In addition to code generation, we've found ChatGPT to be a useful tool for essay writing. Given our inclination towards technical subjects and a reluctance for extensive writing, we initially summarize key points in the document and later leverage ChatGPT to rephrase and refine the content into a more polished and coherent style.

Conclusion

This project has served as a valuable learning experience in the realm of machine learning forecasting. While our initial solutions didn't align with our initial plans, this divergence provided us with a comprehensive understanding of different methods, their workings, and inherent strengths and weaknesses.

The outcome of our machine learning model, while not perfect, demonstrates its effectiveness in providing robust results when evaluated over the average of multiple days. However, its performance tends to be less accurate when comparing true and predicted values for a single day. This discrepancy is acceptable given the inherent variability in daily sales, making the model's overall performance quite good.

REFERENCES

- Kaggle Time Series Course:
 - Link: [Kaggle Time Series Course](#)
 - Experience: Despite its comprehensive content, we encountered challenges in achieving satisfactory results. The complexity of the methods presented made implementation challenging.
- Darts:
 - Link: [Darts Library](#)
 - Experience: While Darts offered interesting functionalities and visualization features, it didn't seamlessly align with our requirements for the Kaggle competition, limiting its utility in that context.
- Medium Sktime Article:
 - Link: [Sktime: Exploring Basic Time Series Forecasting](#)
 - Experience: Sktime, as highlighted in the Medium article, provided us with very good results, and its user-friendly nature made the implementation process relatively easy and efficient. This became a preferred choice due to its effectiveness in our specific project context.