Forecasting Product Sales Using Machine Learning

Dylan J Parker

Western Governors University

Table of Contents

Project Overview

A. Project Highlights

A1. Research Question / Business Case:

- My project addressed a common and important Business Case. Sales forecasting is an integral part of the inner workings of a retail business. Businesses need to ensure they have enough product inventory on hand to meet consumer demand, while ideally not overstocking their inventory. Overstocking results in waste, both in procurement costs, and logistic costs associated with shipping and distribution. An accurate sales forecasting model allows these decisions to be better informed and minimizes these inefficiencies. My project addressed this by developing a robust forecasting model and delivered the predictions to downstream stakeholders.
- There was also a research question I answered using statistical testing, which was the following:
  - **Null Hypothesis**: Exogenous Features to the target variables' time-series data will not produce a meaningful increase in forecasting accuracy.
  - **Alternative Hypothesis**: Exogenous Features to the target variables' time-series data will produce a meaningful increase in forecasting accuracy.

A2. Project Scope:

- The scope of this project was to use the dataset provided by Corporacion Favorita to create a robust sales forecasting model for daily product sales grouped by product family. I first developed a model using only the Endogenous, or time-based features of the target variable. Then a model was developed that also incorporated Exogenous features. The best-performing model was determined using Root Mean Squared Error and the Diebold-Mariano Test to evaluate the statistical significance of the difference between the accuracy of two forecasts. This model was then used to develop the final deliverable, an interactive Tableau Dashboard that displayed the forecasts in the form of a 95% confidence interval. This was all outlined in the project proposal, and I did not deviate from the proposed scope.

A3. Solution Overview - Tools and Methodologies:

- To complete this project, I used the following tools:
  - Github for version control during development
  - Python code running inside a Jupyter Notebook developed in a Linux-based VSCodium environment.
  - I used the pandas library to facilitate the incorporation and manipulation of data in my project.
  - I primarily used the sktime, sci-kit learn, and statsmodels python libraries to incorporate high-level functionality that simplified my workflow when possible.
  - I used Tableau Desktop to create the Dashboard that served as the final deliverable and published it for deployment on Tableau Public.
- The following methodologies were employed throughout the project's lifecycle:
  - CRISP-DM project planning and execution
  - Best practices for version control during development
  - Descriptive Analytics for Exploratory Data Analysis
  - Predictive Analytics to Forecast product sales
  - Best practices for cross-validation and optimization of trained models

- Statistical testing to prove necessary assumptions for provided analytics solutions as well as to evaluate the results.
  - Augmented Dickey-Fuller Test for stationarity of a time-series
  - KPSS test for stationarity of a time-series
  - Anderson-Darling Test of a Normal Distribution
  - Diebold-Mariano Test for the difference in the accuracy of two forecasts against a ground truth.

Project Plan

B. Project Execution

Summarize how the execution of each of these elements differed from the plan presented in Task 2.

B1. Project Plan:

- The proposed project plan was executed without any changes. The stated goal of the project was to create a Tableau dashboard that contains accurate forecasts for a 15-day forecast horizon for product sales of each of the 33 different product families sold by Corporacion Favorita. This was successfully delivered by completing the following three objectives as stated in the proposal:
  - Perform an Exploratory Data Analysis of the provided Dataset.
  - Develop a robust predictive model capable of generating forecasts on a 15-day horizon.
  - Provide downstream stakeholders with a clear representation of our model's forecasts.

B2. Project Planning Methodology:

- The project was planned and executed using the Cross Industry Standard Process for Data Mining (CRISP-DM). CRISP-DM contains the following 6 phases, which were executed using an iterative agile approach:
  - **Business Understanding** – Here I evaluated the business need and the relevant stakeholders. I needed to produce a forecasting model to provide actionable insight into future sales for product procurement and logistics teams. I determined that I should be sure to include a confidence interval for the final forecasts to ensure that the uncertainty surrounding the predictions is highlighted, and that point forecasts are mistaken for certainty.
  - **Data Understanding** – Here I examined the data for inconsistencies and potential errors. I made note of missing values and outliers and used descriptive statistics and visualization to explore the data.
  - **Data Preparation –** Here I addressed some of the missing values by selecting a subsection of the data of sufficient size for effective modeling. The remaining missing values were imputed using a centered mean imputation strategy, with a zero-forecasting approach to making predictions for those dates (if needed) I analyzed outliers by looking at sales promotion data, and if there was an inconsistency, I used a local maximum strategy to reduce the impact of the outlier. If I could communicate with the customer I could choose a better solution, but this was done to simulate the domain knowledge required to effectively address these issues. Preprocessing was done to prepare the data for modeling, including a log1p transformation to stabilize the variance.

- **Modeling** – A Naive forecasting model was used to provide a baseline to test further approaches. A 50-fold expanding window cross-validation strategy was employed to evaluate models using Root Mean Squared Error as the error metric. A random search using the same cross-validation strategy was performed to tune the hyper-parameters of the models.
- **Evaluation** – The models were evaluated on their cross-validation performance and compared to each other using the Diebold-Mariano statistical test. The modeling and evaluation processes were iterated through to obtain a sufficient RMSE on cross-validation as per the proposed project success criteria.
- **Deployment** – Once sufficient performance was obtained, the Tableau Dashboard deliverable was deployed into production. After the initial deployment, successive iterations back to the modeling process were performed to improve the results, providing the ability to receive feedback from stakeholders to inform further development.

B3. Project Timeline and milestones:

- The project's timeline and milestones weren't changed from the proposed plan. Here is a reiteration of the timeline:

Milestone	Projected Start Date	Projected End Date	Duration (days/hours)
Initial EDA	2/08/2023	2/10/2023	2 days
Initial Model	2/10/2023	2/12/2023	2 days
Initial Dashboard	2/12/2023	2/13/2023	1 day
Iterative Deployments	2/13/2023	2/19/2023	6 days
Final Deployment	2/19/2023	2/20/2023	1 day

Methodology

C. Data Collection Process

Discuss these elements; offer examples.

Actual data selection vs. planned collection process:

- Regarding the actual selection of the dataset for the project, I came up with the project idea after finding the dataset on Kaggle, so the actual collection process was simply downloading the dataset from Kaggle.
- Before exploring the data, I assumed I would use all the data provided within the dataset for modeling. I realized that there were many product families with large chunks of missing data at the beginning of the dataset. This could be because the product families were not carried by Corporacion Favorita stores at the time. I found that by selecting roughly the last 2 years of the data, I was able to have a sufficient size dataset to train my models while also eliminating this problem for the most part.

Obstacles to data collection:

- There were no major obstacles to the data collection process. All files were provided for download on Kaggle and are easily accessible.

Unplanned data governance handling:

- There was no sales data provided for Christmas, and almost no data for New Year's Day throughout the entire dataset. I assumed this was due to the Corporacion Favorita stores being closed on these days. There were also some extreme outliers in the data that were likely errors in data collection. If I could communicate with the customer I could obtain more accurate information, but I made judgment calls on how to handle these situations to simulate this domain-specific knowledge.
- The fact that the dataset was provided to Kaggle by Corporacion Favorita eliminated governance issues that could occur from using proprietary or sensitive information.

C1. Advantages and Limitations of Data Set

- Some advantages of the Dataset were that it was easily accessible, decently clean, and formatted. It was straightforward to import it using pandas and explore it. Also, there was more than enough data provided to develop an effective model.
- Some disadvantages of the dataset were that there was complex cyclical behavior and multiple seasonal periods present, which made it relatively difficult to model using simple straightforward approaches. Also, there were underlying questions such as those previously mentioned that would most effectively be answered through communication with the customer, but this was not possible in this case.

D. Data Extraction and Preparation Processes

The Data Extraction process was done using the Pandas Python package inside my Jupyter notebook. I loaded the training dataset into a Pandas DataFrame and ensured that the index was properly parsed as a DateTime object. Although the dataset provided daily product family sales for each Corporacion Favorita store, I aggregated these sales across all stores as previously discussed in my project proposal. I also loaded all of the exogenous feature datasets into separate DataFrames to be explored and analyzed.

The Data preparation process involved identifying the previously mentioned issues with missing values and outliers and cleaning the data. I also had to address missing values in the oil price dataset, which I handled with a simple linear interpolation. I formatted the data by using the timestamp as the index and making each product family's sales data its own column. This was the format specified for use with the sktime library. After finishing the data selection and cleaning process, I exported the endogenous training data and exogenous features to CSV files for use in the modeling notebook.

E. Data Analysis Process

E1. Data Analysis Methods

During the EDA process, descriptive analytics were used to learn more about the structure of the data and gain some insight that could inform our model-building process. It was observed that there was very strong weekly seasonality present in the dataset, and complex cyclical behavior. It was determined that many of the product families were non-stationary time series. This meant I needed to apply some transformation to make the processes stationary or use a modeling technique where stationarity was not an underlying assumption. It was also observed that the variance of the data was highly unstable, which indicated the modeling process could benefit from a transformation of the data to stabilize the variance.

During the modeling process, predictive analytics were used to develop a model that could generate forecasts informed by the historical data, and potentially by incorporating exogenous features into the models. These models were evaluated both using a hold-out test set, as well as a 50-fold expanding window cross-validation process. The model hyper-parameters were tuned using a random search with cross-validation.

The specific models that were tested were as follows:

- Naive Forecaster that used a simple moving average based on the last 7 values.
- Exponential Smoothing
- STL Decomposition based forecaster.
- Light Gradient Boosted Machine Random Forest model trained on a tabularized reduction of the data.

The exogenous features that I incorporated into the modeling were the product families' promotion data which indicated a representation of the extent to which a particular product family was under some promotion for a given day. I would have liked to include more exogenous features but the time for tuning the hyper-parameters was around 8 hours and for the scope of this project I decided to limit it there.

Whenever I obtained a new best-performing model in cross-validation, I generated forecasts for the hold-out test set and used that to update the Dashboard. Before determining whether a model was performing better the forecasts of the new model were compared to that of the old model using a statistical test for significance.

After evaluating and obtaining the best model, a 95% confidence interval was calculated using Root Mean Square Forecast Error and this was used to build the final Dashboard deliverable.

These analytical methods were suitable for this project because predictive analytics is required to produce the forecasts necessary to satisfy the customer's business needs.

E2. Advantages and Limitations of Tools/Techniques

Some of the advantages of the tools/techniques I chose:

- The sktime library has built-in methods for industry-standard cross-validation methods for time-series data, as well as a random hyper-parameter search that uses cross-validation and allows for nested parameter tuning. This allowed me for example to tune the value of the window used to make the tabularized reduction of the data passed to the LGBM model while tuning its hyper-parameters at the same time.
- Using an LGBM random forest model meant that the stationarity of the data was not an underlying assumption that needed to be satisfied.
- The statsmodels library had built-in methods for many statistical tests which simplified my workflow.
- The sktime library's plotting functions for time plots, autocorrelation plots, and partial autocorrelation plots were also extremely useful in simplifying my workflow when describing the data.
- Using Root Mean Squared Error as my error metric was useful because it is easily interpretable by non-technical stakeholders. The units of this metric are the same as the

target variable in question, so if I forecast unit sales of products, the error metric is also reported in unit sales of products.

Some of the limitations of the tools/techniques I chose:

- The random hyper-parameter search method of the sktime library is designed to only handle one univariate time series, so I had to iterate through each of the 33 product families which greatly impacted the time it took to optimize the models.
- The sktime library does not natively support GPU acceleration, which makes the scalability of my method questionable. Even after simplifying aggregating the individual store data, I ran into performance bottlenecks.
- A major limitation of the Root Mean Square Error as an error metric is that it is sensitive to outliers, which in retrospect could have been a problem with this dataset.

E3. Application of Analytical Methods

Descriptive Analytics:

- Checked time series for stationarity using the Augmented Dickey-Fuller test and KPSS Statistical Tests. I built a custom wrapper function to perform both of these tests from the statsmodels library and display the relevant test statistics, p-values, and test statistic thresholds. An alpha= .05 was used to evaluate these tests.
    - The Augmented Dickey-Fuller test has a null hypothesis that the time series contains a unit root and thus is not stationary. The Alternative hypothesis is that there is no unit root present and thus the time series is stationary.
    - The KPSS test has a null hypothesis that the time series is trend-stationary and an alternative hypothesis that the time series contains a unit root and thus is not stationary.
- The results of these two statistical tests together can help confirm if a time series is stationary and in cases where they do not agree, potentially give more information about how one could make the process stationary. In this case, most of our product family time series were difference stationary, meaning that the ADF test's p-value was sufficient to reject its null hypothesis, and the KPSS test's p-value was also sufficient to reject the null hypothesis. I applied a first-order differencing operation to make all of the time series stationary, but when I tried to integrate the differenced data into the models, I had issues with negative values being generated at predictions even after inverting the differencing. Ultimately, I decided to use a predictive solution that did not rely on stationarity as an underlying assumption (G.V.K, 2021).
- STL decomposition was used from the sktime library to decompose the time series into their trend, seasonal and residual components and analyze their autocorrelation plots and partial autocorrelation plots. It should be noted that this approach combines both the linear trend component and non-linear cyclical behavior.
    - Visual analysis of these plots was done as follows:
        - I adjusted the hyperparameter that adjusts the seasonal period and the smoothness of the trend component to look for clean separations of the 3 components, which can be identified by a trend component that has an autocorrelation plot that slowly decreases over time, and a partial autocorrelation plot that has a strong value at lag 1, and then goes abruptly to almost no autocorrelation at successive lag values. For the seasonal component, you would like to see strong autocorrelation at multiples of lag values of the seasonal period, and partial autocorrelation at these values that decreases somewhat over time. For the residual component, you would like to see almost no autocorrelation or partial

autocorrelation, which indicates that you have captured most of the trend and seasonality with the first two components.

- By looking at these correlation plots the seasonal periods with strong signals become easily apparent. There was an obvious seasonal period of 7 days present in almost all of the time series. Some other less prevalent periods appeared to be present, but this was the biggest takeaway for me that could inform my modeling process.
- Before beginning the modeling process, the skewness and kurtosis values of the dataset were checked using built-in pandas functions. Their values indicated that the variance of the data was unstable and suggested that a transformation to stabilize the variance might help in the modeling process. It should be noted that this transformation would need to be reversed when forecasts are made to return to the scaling of the original data. A log1p transformation was chosen with a conditional application for the non-zero values of the dataset. Although the log1p transformation can handle zero values, I chose to keep the zero values because, in the context of predicting product sales, zero values in the dataset are potentially an important signal. After applying this transformation, the skewness and kurtosis were reduced to within an acceptable range, indicating that the variance had stabilized substantially.

Predictive Analytics:

- A Naive Forecaster model was used as a baseline, looking at a moving average of the last 7 values, corresponding to the seasonal period observed in the STL decomposition. The performance on the hold-out test set was measured at a Root Mean Square Error of 85.35.
- The cross-validation strategy started with an initial window of 42 days and used the forecast horizon of 15 days that we wanted our model to predict into the future. Each fold took the 15 days used as a test set from the previous fold and incorporated it into training, then used the next 15 days as a holdout for testing. This means that after 50 folds of cross-validation, the model has predicted every 15-day forecast horizon in the entire training set after the initial window. The RMSE was measured for each fold and averaged to generate the cross-validation score. This process helps build confidence that our evaluation is determining how well a model generalizes to the future, and not just how well it performs on prediction of any one specific time interval.
- I built a custom wrapper function to efficiently iterate through the 50-fold, expanding window cross-validation process and tried out some other models with default hyper-parameters and with some light tuning for the seasonal period of 7 days. The best-performing models were the STLForecaster and the LGBMRegressor models.
- Next, I implemented the ForecastRandomSearchCV method from the sktime library to perform hyper-parameter tuning for the models. The best-tuned model was an LGBMRegressor model that achieved a cross-validation score of 69.33.
- In the next iteration I implemented exogenous features into the LGBMRegressor model, which drastically increased the time to run the hyper-parameter tuning. Even when only using the promotion data for the product families, the optimization run time was around 8 hours. This process resulted in my best-performing model in cross-validation with a score of 68.35.
- When comparing these two models forecasting performed on the holdout test set, the model without exogenous features had an RMSE of 68.62, and the model with exogenous features included for the dates in the forecast horizon (which in the case of product promotion values, would be available for the short-term future) had an RMSE of 62.92.

- This seemed like a meaningful increase in accuracy, but to verify this I used the Diebold-Mariano statistical test. I will include more details about this in the following section, but it compares two forecasts against ground truth, to determine if the predictive accuracy of the two forecasts is meaningfully different. An alpha=.05 was used to evaluate this test.
- After determining that the model with exogenous features was objectively the best-performing model both in cross-validation and on the hold-out test set, I used Root Mean Square Forecast Error to compute a 95% confidence interval. This process relies on the assumption that the residuals of the forecasts are normally distributed.
- To assert this assumption, the Anderson-Darling statistical test was used with alpha = .01. This test has a null hypothesis that the data is normally distributed and an alternative hypothesis that it is not normally distributed. From evaluating my final model's forecast residuals, every product family's forecast residuals failed to reject the null hypothesis except for one. (The CLEANING product family)
- The process to compute the confidence interval was to use the square root of the mean of the squares of the forecast residuals and multiply this by 1.96 standard deviations to obtain a 95% confidence band. In the dashboard, this value would be added and subtracted from the point forecasts generated by the model to obtain the upper and lower bounds of the confidence interval (Artley, 2022).
- When incorporated into the visual dashboard, these confidence intervals contained the ground truth of the hold-out test set for almost all data points.

Results

F. Project Success

F1. Statistical Significance

As previously mentioned, to evaluate the statistical significance of our predictive analytics solution, the Diebold-Mariano test was utilized. The research presenting this test was summarized in the project proposal. It is a good fit because, unlike many such tests that came before it, it does not rely on any underlying assumptions about the data such as stationarity. It compares two sets of forecasts against ground truth values and evaluates if the accuracy of the forecasts is meaningfully different.

This test is a two-tailed test that has the following hypotheses:

- Null Hypothesis: The accuracy of the forecasts is not meaningfully different from one to another.
- Alternative Hypothesis: The accuracy of the forecasts is meaningfully different.

I implemented this statistical test using a custom-built function in my modeling notebook. The test is performed as follows:

- Compute the residuals of each forecast from the ground truth by subtracting the forecast from the actual values.
- Compute the loss differential series by taking the difference of the squares of the residuals.
- Calculate the mean, variance, and standard error of the loss differential series.
- Compute the auto-covariance by taking the sum of the difference between the loss differential series and its mean.
- Compute the Diebold-Mariano test statistic by subtracting the mean of the loss differential series from the auto-covariance and dividing this result by the standard error.

- Perform a two-tailed t-test using the cumulative distribution function to obtain the p-value.

The results from this test are demonstrated in the HTML export of my Jupyter notebook. I determined that the model without exogenous features and the model with exogenous features forecasts on the hold-out test set were meaningfully different from one another with alpha=.05, with a p-value of .036. This means we can reject the null hypothesis that the forecast accuracy is not meaningfully different between the two sets of forecasts. This answered my research question, which was that the introduction of exogenous features would not generate meaningful predictive accuracy. In this case, this evidence is sufficient to reject my null hypothesis.

F2. Practical Significance

The practical significance is that I successfully developed a model that can provide Corporacion Favorita with valuable insights into the short-term future of their product sales. This satisfies their business need and will allow them to generate value by minimizing their excess inventories while still meeting product demand. This also allows them to minimize shipping and distribution costs and increase their short-term liquidity.

The discovery that the exogenous features do provide meaningful predictive accuracy to the model has practical significance in that perhaps there are other features we could use to gain even higher accuracy in our forecasts. We would need to make sure that these features are available at prediction time and consider the increase in computational cost associated with making the model more complex.

F3. Overall Success

I believe this project was a success because I completed the proposed goal and all its objectives. I met all the proposed KPIs, which I will re-iterate here:

Criterion/Metric	Required Data	Cut Score for Success
RMSE of cross-validation	Forecast of each fold of the CV	< mean 75 across all families
Diebold-Mariano Test	Forecasts of 2 different models	P-value < 0.05 (95% CI)
Final Confidence interval	Final model predictions and ground truth of test set	Confidence interval must visually include the ground truth.

These metrics were all met successfully, as I have already detailed in the above sections.

G. Key Takeaways

G1. Summary of Conclusions

- The best-performing model that made the final deployment was an LGBM model trained on a tabularized reduction of the training data.
- The incorporation of exogenous features into the model did objectively increase the predictive accuracy of the forecasts. The sales promotion data for the product families were used.
- The introduction of the exogenous features drastically increased the computational expense and training time of the model optimization. I think moving forward a more scalable model should be developed using a framework that supports GPU acceleration such as TensorFlow or PyTorch. This could more easily handle iterations of model optimization and incorporate more exogenous features.
- Another possible improvement could be to use a truly multivariate modeling framework such as an LSTM that can handle all the time series at once instead of having to iterate through each product family and develop a separate model for each.
- From analyzing the dashboard and the visual representations of the forecasts' confidence interval vs the ground truth, it is evident there are some product families with much higher variance in the forecasts than others.
- This could be due in some part to the selection of Root Mean Square Error as the error metric, which does not handle outliers well. Looking closer at the visualization it also seems many of the product families with high degrees of uncertainty are ones that have low sales daily sales volumes. Another improvement moving forward would be to compare and test different error metrics and see their effect on the outcome.

G2. Effective Storytelling

The tools and graphical visualizations provided in the final deliverable Dashboard provide effective storytelling. The dashboard clearly shows a representation of the historical data, and also overlays the model's forecasts on the hold-out test set against the ground truth. This makes it obvious that we are not providing an exact answer with our forecasts, but rather giving a range that we are confident is likely.

The dashboard is also interactive which allows it to contain the data from all 33 product families without becoming cluttered, as the user can cycle through the product families and view their respective forecasts.

The visual component of the dashboard is designed to be obvious in what it presents, with very little explanation required to conclude. This is a hallmark of effective data storytelling.

I have also included the actual values of the point forecasts and cutoffs for the confidence interval below the visualization in case a higher level of granularity than the visual provides is desired.

G3. Findings-based Recommendations

- Although the business need was met by the completion of this project, the accuracy of this model will likely deteriorate over time. Thus, one recommendation is to periodically

evaluate the performance of the model to look for data drift and potentially revisit the iterative process to improve the model and keep it up to date.
- The findings of the research question also present an obvious recommendation to obtain and test more exogenous features to try and further increase the accuracy of the forecasts.

H. Panopto Presentation

Here is a link to my Panopto presentation:
https://wgu.hosted.panopto.com/Panopto/Pages/Viewer.aspx?id=0297e01e-a1a0-494c-a21e-afaf01648a97

Appendices

I. Evidence of Completion

All code and associated files are contained in my GitHub repository and available for viewing.
https://github.com/Dylan-Parker/store_sales_prediction

The final deliverable of the project is hosted on Tableau Public here:
https://public.tableau.com/app/profile/dylan.parker/viz/sales_prediction_dashboard/Dashboard1

I have exported both Jupyter notebooks containing my python code and its output from running into HTML files, which I printed into PDF format. They will be submitted alongside this document. They are also available on my GitHub.

NOTE: my modeling notebook's hyper-parameter tuning function's output contains a warning that I was not able to suppress. It has been reported as an issue to the maintainers of the codebase and has not yet been solved. It does not affect the output in any way, but it does generate a large amount of text that you have to sift through to see the output. I have edited the version of the HTML file that I am uploading to remove these warnings because it can overload the memory of some computers. I searched through all of the parameters of the file with error output in my Panopto presentation and you can see that nothing is altered in this file it is just done for ease of viewing. The original file is on my github.

Sources

Artley, B. (2022, June 27). *Time series forecasting: Prediction intervals*. Medium. Retrieved February 22, 2023, from https://towardsdatascience.com/time-series-forecasting-prediction-intervals-360b1bf4b085

G, V. K. (2021, June 16). *Statistical tests to check stationarity in time series*. Analytics Vidhya. Retrieved February 22, 2023, from https://www.analyticsvidhya.com/blog/2021/06/statistical-tests-to-check-stationarity-in-time-series-part-1/