10 Academy Batch 3: Week 5

Pharmaceutical Sales prediction across multiple stores

Overview

Business Need

You work at **Rossmann Pharmaceuticals** as a data scientist. The finance team wants to forecast sales in all their stores across several cities six weeks ahead of time. Managers in individual stores rely on their years of experience as well as their personal judgement to forecast sales.

The data team identified factors such as promotions, competition, school and state holidays, seasonality, and locality as necessary for predicting the sales across the various stores.

Your job is to build and serve an end-to-end product that delivers this prediction to Analysts in the finance team.

Data and Features

The data and feature description for this challenge can be found here.

Data fields

Most of the fields are self-explanatory. The following are descriptions for those that aren't.

Id - an Id that represents a (Store, Date) duple within the test set
Store - a unique Id for each store
Sales - the turnover for any given day (this is what you are predicting)
Customers - the number of customers on a given day
Open - an indicator for whether the store was open: 0 = closed, 1 = open
StateHoliday - indicates a state holiday. Normally all stores, with few exceptions, are closed on state holidays. Note that all schools are closed on public holidays and weekends. a = public holiday, b = Easter holiday, c = Christmas, 0 = None
SchoolHoliday - indicates if the (Store, Date) was affected by the closure of public schools
StoreType - differentiates between 4 different store models: a, b, c, d
Assortment - describes an assortment level: a = basic, b = extra, c = extended. Read more about assortment here
CompetitionDistance - distance in meters to the nearest competitor store
CompetitionOpenSince[Month/Year] - gives the approximate year and month of the time the nearest competitor was opened
Promo - indicates whether a store is running a promo on that day
Promo2 - Promo2 is a continuing and consecutive promotion for some stores: 0 = store is not participating, 1 = store is participating
Promo2Since[Year/Week] - describes the year and calendar week when the store started participating in Promo2
PromoInterval - describes the consecutive intervals Promo2 is started, naming the months the promotion is started anew. E.g. "Feb,May,Aug,Nov" means each round starts in February, May, August, November of any given year for that store

Learning Outcomes

- Technical Skills: Pandas, Matplotlib, Numpy, HTML and CSS ,Flask. They will also learn how to write modular code.
- Creation of new features
- Predictive pipeline: Exploratory data analysis, data wrangling, building and fine-tuning models

- Deployment: Students will know how to serve predictions in a basic web app. The app will written with HTML/CSS and served with flask

Team

- Emmanuel Sekyi with Jean-Henock
- Usman, Abla, Moustapha, Sebastian

Key Dates

- Discussion on the case - 1130 Rwanda time on Monday 17 August 2020. Use #all-week 5 to pre-ask questions.
- Interim Solution - 2000 Rwanda time on Tuesday 18 August 2020.
- Final Submission - 2000 Rwanda time on Saturday 22 August 2020

Group Work Policy

You are expected to complete Tasks 2 and 3 with your assigned group. Task 1 is to be done individually. All members of the group can submit the same code and Heroku link for Tasks 2 & 3. We recommend that everyone keeps a copy of this code in their own GitHub repository.

The interim report must be done individually.

We expect all group members to contribute equally. We leave the assignment of roles within groups to the group members.

Grading for the week

There are 100 points available for the week.

20 points - community growth and peer support. This includes supporting other learners by answering questions (Slack), asking good questions (Slack), participating (not only attending) daily standups (GMeet) and sharing links and other learning resources with other learners.

25 points - presentation and reporting.

5 points - interim submission

5 - Requirements met, clear presentation

3 - Most requirements met, presentation acceptable

1 - Some effort made

20 points for the final submission. This is measured through:

- Clarity of graphs (5 points)
- Clarity of message (5 points)
- Professionalism/production value (free of spelling errors, use of same font, well produced) (5 points)
- Balance between being 'full of information' and 'easy to understand' (5 points)

55 points - data analysis and coding

10 points - interim submission

Validity of recommendations made (5 points)

Quality of code (including readability) (5 points)

45 points - final submission

Validity of recommendations made (25 points)

Quality of code (20 points)

Badges

Each week, one user will be awarded one of the badges below for the best performance in the category below.

In addition to being the badge holder for that badge, each badge winner will get +20 points to the overall score.

Visualization - quality of visualizations, understandability, skimmability, choice of visualization

Quality of code - reliability, maintainability, efficiency, commenting - in future this will be CICD

Innovative approach to analysis -using latest algorithms, adding in research paper content and other innovative approaches
Writing and presentation - clarity of written outputs, clarity of slides, overall production value
Most supportive in the community - helping others, adding links, tutoring those struggling

The goal of this approach is to support and reward expertise in different parts of the Data Scientist toolbox.

Late Submission Policy

Our goal is to prepare successful learners for the work and submitting late, when given enough notice, shouldn't be necessary.

For interim submissions, those submitted 1-6 hours late will receive a maximum of 50% of the total possible grade. Those submitted >6 hours late may receive feedback, but will not receive a grade.

For final submissions, those submitted 1-24 hours late, will receive a maximum of 50% of the total possible grade. Those submitted >24 hours late may receive feedback, but will not receive a grade.

When calculating the leaderboard score:

- From week 8 onwards, your two lowest weeks' scores will not be considered.

A.Instructions

The task is divided into the following objectives

- Exploration of customer purchasing behavior
- Prediction of store sales
- Serving predictions on a web interface

Task 1 - Exploration of customer purchasing behavior

Exploratory data analysis is the lifeblood of every meaningful machine learning project. It helps us unravel the nature of the data and sometimes informs how we go about modelling. A careful exploration of the data encapsulates checking all available features, checking their interactions and correlation as well as their variability with respect to the target.

In this task, we seek to explore the behaviour of customers in the various stores. Our goal is to check how some measures such as promos and opening of new stores affect purchasing behavior.

To achieve this goal, we need to first clean the data. The data cleaning process will involve building pipelines to detect and handle outlier and missing data. This is particularly important because we don't want to skew our analysis.

Visualizing various features and interactions is necessary for clearly communicating our findings. It is a powerful tool in the data science toolbox. Communicate the findings below via the necessary plots.

We can use the following questions as a guide during your analysis. It is important to come up with more questions to explore. This is part of our expectation for an excellent analysis.

- Check for seasonality in both training and test sets - are the seasons similar between these two groups?
- Check & compare sales behavior before, during and after holidays

- Find out any seasonal (Christmas, Easter etc) purchase behaviours,
- What can you say about the correlation between sales and number of customers?
- How does promo affect sales? Are the promos attracting more customers? How does it affect already existing customers?
- Could the promos be deployed in more effective ways? Which stores should promos be deployed in?
- Trends of customer behavior during store open and closing times
- Which stores are opened on all weekdays? How does that affect their sales on weekends?
- Check how the assortment type affects sales
- How does the distance to the next competitor affect sales? What if the store and its competitors all happen to be in city centres, does the distance matter in that case?
- How does the opening or reopening of new competitors affect stores? Check for stores with NA as competitor distance but later on has values for competitor distance

Deliver your exploratory analysis notebook - make sure you answer all the questions asked in task 1 using the appropriate plots or summary tables and give useful insights. A 3 - 5 slides presentation is enough for interim submission.

Task 2 - Prediction of store sales

Tasks 2 and 3 are to be done in groups. Groups are as follows

Prediction of sales is the central task in this challenge. We want to predict daily sales in various stores up to 6 weeks ahead of time. This will help the company plan ahead of time.

There following steps outline the various sub tasks needed to effectively do this:

Preprocessing

It is important to process the data into a format where it can be fed to a machine learning model. This typically means converting all non-numeric columns to numeric, handling NaN values and generating new features from already existing features.

In our case, we have a few datetime columns to preprocess. We can extract the following from them:

- weekdays
- weekends
- number of days to holidays
- Number of days after holiday
- Beginning of month, mid month and ending of month
- (think of more features to extract), extra marks for it

As a final thing, we have to scale the data. This helps with predictions especially when using machine learning algorithms that use Euclidean distances. We can use the standard scaler in sklearn for this.

Building models with sklearn pipelines

At this point, all our features are numeric. Since our problem is a regression problem, we can narrow down the list of algorithms we can use for modelling.

A reasonable starting point will be to use any of the tree based algorithms. Random forests Regressor will make for a good start.

Also, for the sake of this challenge, work with sklearn pipelines. This makes modeling modular and more reproducible. Working with pipelines will also significantly reduce your workload when you are moving your setup into files for the next part of the challenge. Extra marks will be awarded for doing this.

Choose a loss function

Loss functions indicate how well our model is performing. This means that the loss functions affect the overall output of sales prediction.

Different loss functions have different use cases.

In this challenge, you're allowed to choose your own loss function. We need to defend the loss function we choose for this challenge. Feel free to be creative with your choice. You might want to use loss functions that are easily interpretable.

Post Prediction analysis

Let's explore the feature importance from our modelling. Creatively deduce a way to estimate the confidence interval of your predictions. Extra marks will be give for this.

Serialize models

To serve the models we built above, we need to serialize them. Save the model with the timestamp(eg. 10-08-2020-16-32-31-00.pkl). This is necessary so that we can track predictions from various models.

Assume that you'll make daily predictions. This means you'll have various models for predictions hence the reason for serializing the models in the format above.

Task 3 - Serving predictions on a web interface

Imagine a hypothetical solution where you as the data scientist need to serve predictions daily. There are a few ways to approach this problem. One approach is to run predictions in the jupyter notebook and send a report to the managers.

Now imagine 100 different stores. All of a sudden, this method does not scale.

An alternative approach is to build a basic web app to automatically do everything from modelling to serving predictions. Your task is to build a solution at scale using the steps outlined below:

Basic webpage in html and css

This will serve as a dashboard for the managers. We can display predictions in a table. Please feel free to be as creative as possible here.

Introduction to flask

Flask is a python library for building server side applications. It is a micro framework. This means it is really stripped down and you can easily pick it up and work with.

Preparing scripts for data preprocessing

Write a script to automatically do the preprocessing for modelling. This script should be well documented with docstrings. For a guide on how to write excellent docstrings, check the sklearn documentation.

Using serialized model for prediction

Save your models in pickle format.

Logging

Log your steps using the logger library in python.

Can we give them 3-5 usage cases that their website should be able to serve?

- All stores
- Individual store
- Comparison of Christmas to pre-Christmas

Hosting

We will use heroku for hosting. Create a free heroku instance and deploy your code. Submit a link to your site.

References

1. Loss functions
2. Sklearn pipelines
3. Merging dataframes
4. Introduction to flask
5. HTML and CSS
6. Time series analysis
7. RandomForests

Kaggle kernels -

8. https://www.kaggle.com/c/rossmann-store-sales/notebooks
9. https://www.kaggle.com/thie1e/exploratory-analysis-rossmann
10. https://www.kaggle.com/elenapetrova/time-series-analysis-and-forecasts-with-prophet
11. https://www.kaggle.com/shearerp/interactive-sales-visualization
12. https://www.kaggle.com/michaelpawlus/obligatory-xgboost-example
13. https://www.kaggle.com/stefanozakher94/eda-and-forecasting-with-rfregressor-final-updated
14. https://www.kaggle.com/emehdad/time-series-linear-models-tslm
15. https://www.kaggle.com/sammyshen/exploratory-and-randomforest