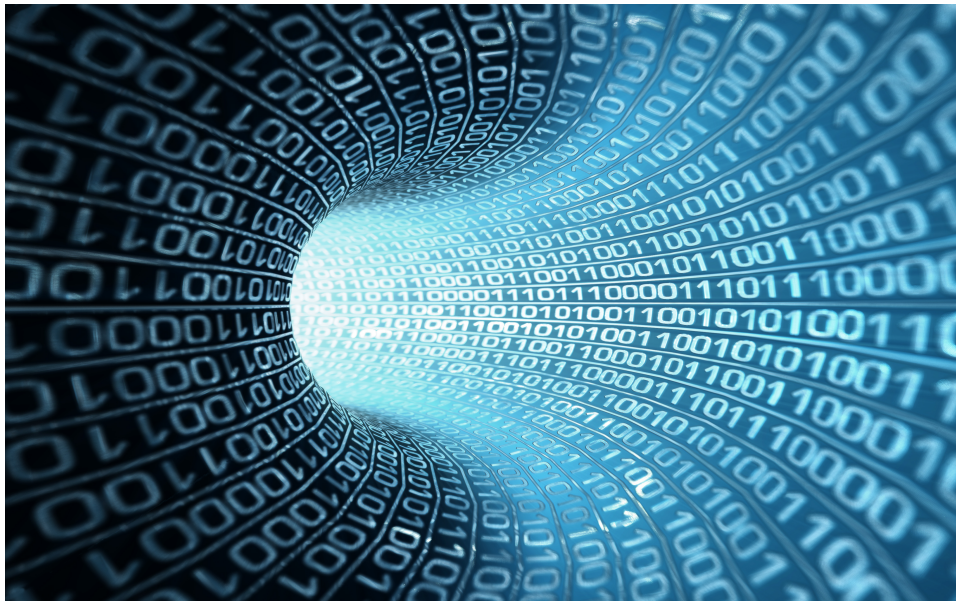
Business solutions powered by AI and data science

John H. Muller



Overview

Today I would like to do the following:

- Introduce myself and highlight my skills and experience,
- Review how I attacked a wide variety of business problems,
- Discuss how I can help with your challenges.

About me

- Computer Scientist (PhD), data scientist and finance quant,
- Twenty years of experience turning data challenges into competitive advantages through machine learning and AI solutions.
- Industry experience in
  - Financial Services
  - Quantitative Investing
  - Retail
  - Image and video compression
- Passion for solving business problems and making my work matter.

Skills, Experience and Tools

- Data Science
  - Supervised learning: classification, regression, tree based methods
  - Unsupervised learning: clustering, dimension reduction
  - Neural Networks, Deep Learning and Generative AI
- Coding: Python and the python ecosystem
  - pandas, numpy, scikit-learn, pytorch ...
- SQL
- Frameworks: LangChain, Langgraph, CrewAI, Hugging Face
- Computer Science: algorithms and data structures

Classification

A classification problem involves trying to predict a discrete class given given a set of predictor variables.

For example, we might try to predict a medical condition, e.g. whether a patient will have a stroke, given some observed features of the patient such as age and bmi.

It is often useful to try several methods on a given data set and choose the one that best solves the particular situation.

To decide which method to choose we might compare them on error rate, computational load, availability of data, interpretability of results, and more.

In this section I will discuss the pros and cons of two methods for a medical diagnosis data set: Logistic Regression and Random Forests

The dataset has 11 predictor variables and one target variable, **stroke**

1. id: unique identifier
2. gender: "Male", "Female" or "Other"
3. age: age of the patient
4. hypertension: 0 if the patient doesn't have hypertension, 1 if the patient has hypertension
5. heart_disease: 0 if the patient doesn't have any heart diseases, 1 if the patient has a heart disease
6. ever_married: "No" or "Yes"
7. work_type: "children", "Govt_jov", "Never_worked", "Private" or "Self-employed"
8. Residence_type: "Rural" or "Urban"
9. avg_glucose_level: average glucose level in blood
10. bmi: body mass index
11. smoking_status: "formerly smoked", "never smoked", "smokes" or "Unknown"*
12. stroke: 1 if the patient had a stroke or 0 if not

Logistic regression

- an old stand-by method
- assumes a linear relationship, but more complicated than OLS
  $\Longrightarrow$ can interpret variables impact
- very fast to fit and predict

Random Forests

- non-linear $\Longrightarrow$ not as easy to interpret
- predicts the average over many decision trees
- more computation to fit and predict.

A common loss function for classification problems is Cross-entropy. Below I show this loss for the two methods on the train and test data sets. The test loss is a good indication of out-of-sample performance.

Method	Training Loss	Test loss
Logistic	0.15	.135
Random Forest	0.095	.105

Criteria to consider when choosing the method

- loss (Random Forests)
- interpretability (Logistic Regression)
- required compute for train and fit (Logistic Regression)
- how it deals with ordinal or categorical input (Random Forests)

Which methods to choose depends on how much weight on each criteria.

The previous example involved *structured* data, that is, the training data is like a spreadsheet.

But there are also many applications of classification to unstructured data. For example,

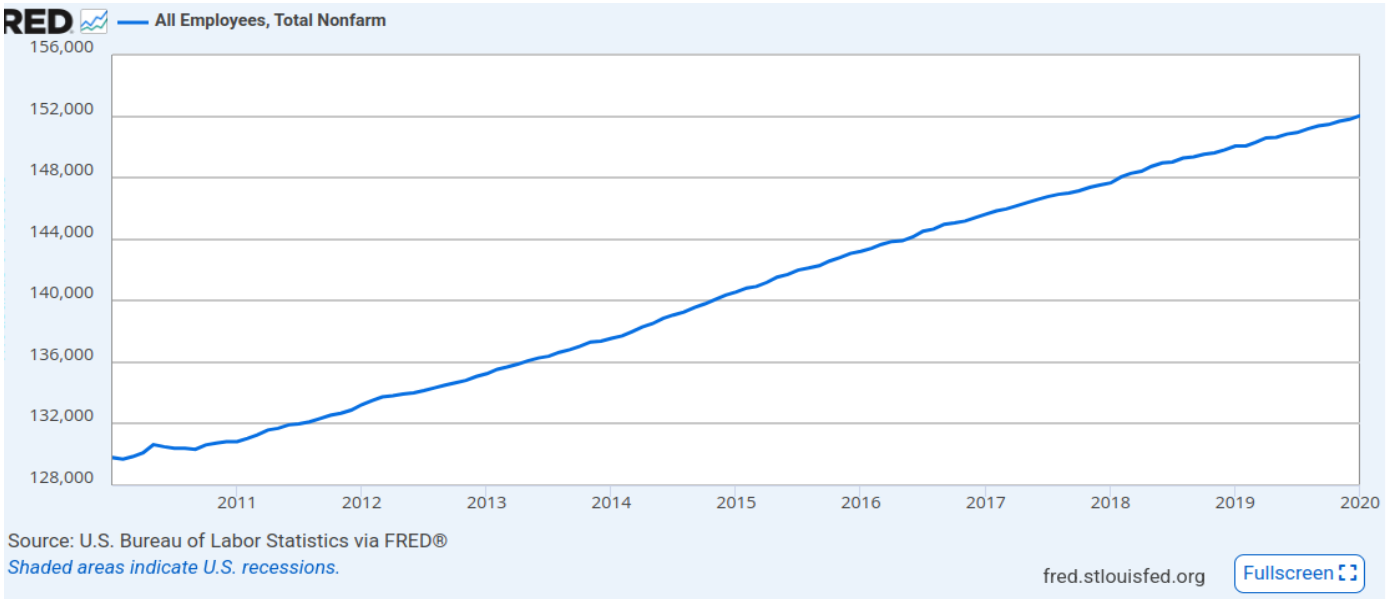
- Images: From a picture of a roof, determine if it is damaged,
- Text: classify sentiment of customer feedback,
- Audio: Determine the sentiment of a customer from their voice.

Regression

Regression is the other most common data science problem. Instead of predicting a *class*, regression tries to predict a numeric outcome. Examples include:

- What will be the price of Google stock tomorrow,
- How many TVs will be sold next month,
- How many new jobs were created in the US in a month.

We will examine the last one, that is, predict the monthly **change**, that is the month over month change, in US Payroll. See below for monthly levels.



From the previous plot, the *level* of the data looks fairly predictable. But our interest is in the monthly *changes* which are shown below and look much harder to predict.

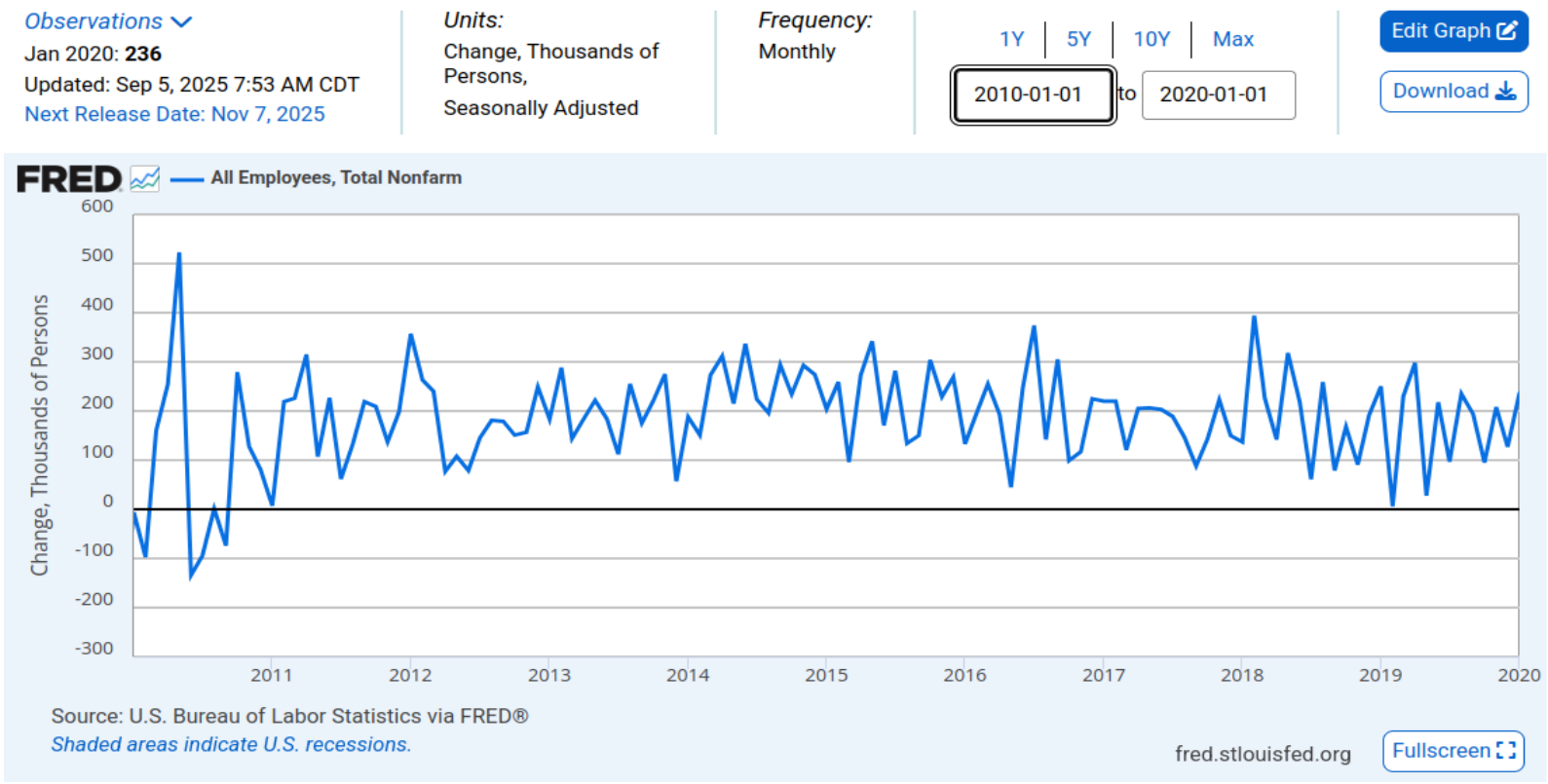


Figure 2.2: NonFarm Payroll, changes

Our predictor variables could be anything *job* related, including

1. One or more previous values,(monthly)
2. Unemployment rate. (monthly)
3. Civilian Labor force. (monthly)
4. Job Openings (monthly)
5. GDP (quarterly)
6. unemployment claims (weekly)
7. ADP payroll changes (monthly)
8. Google trend values for "job", "layoff', ... (hourly)

I will start with the just the first few predictor variables from above.

Levels of the predictor variables.

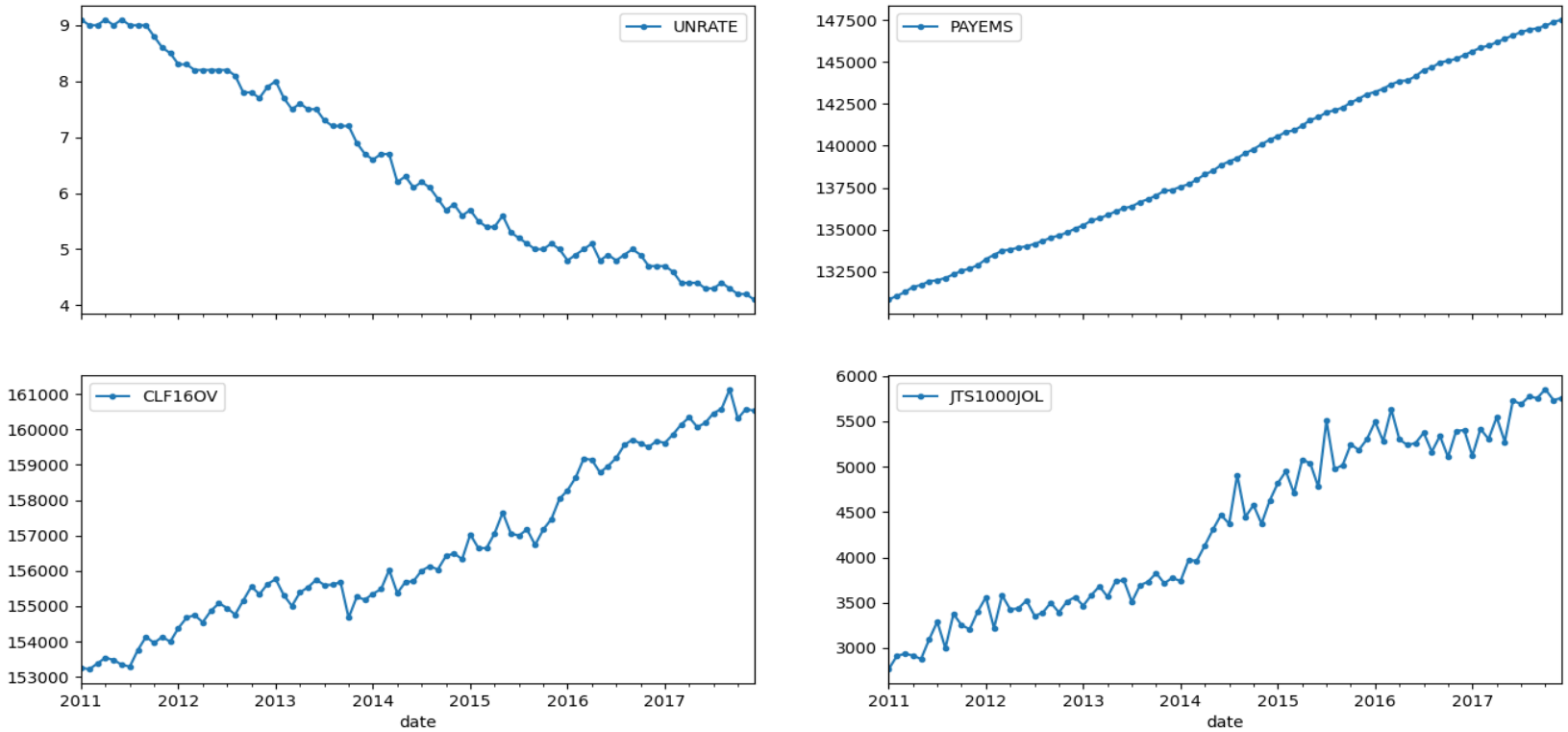


Figure 2.3: predictor variables, level

Changes in the predictor variables.

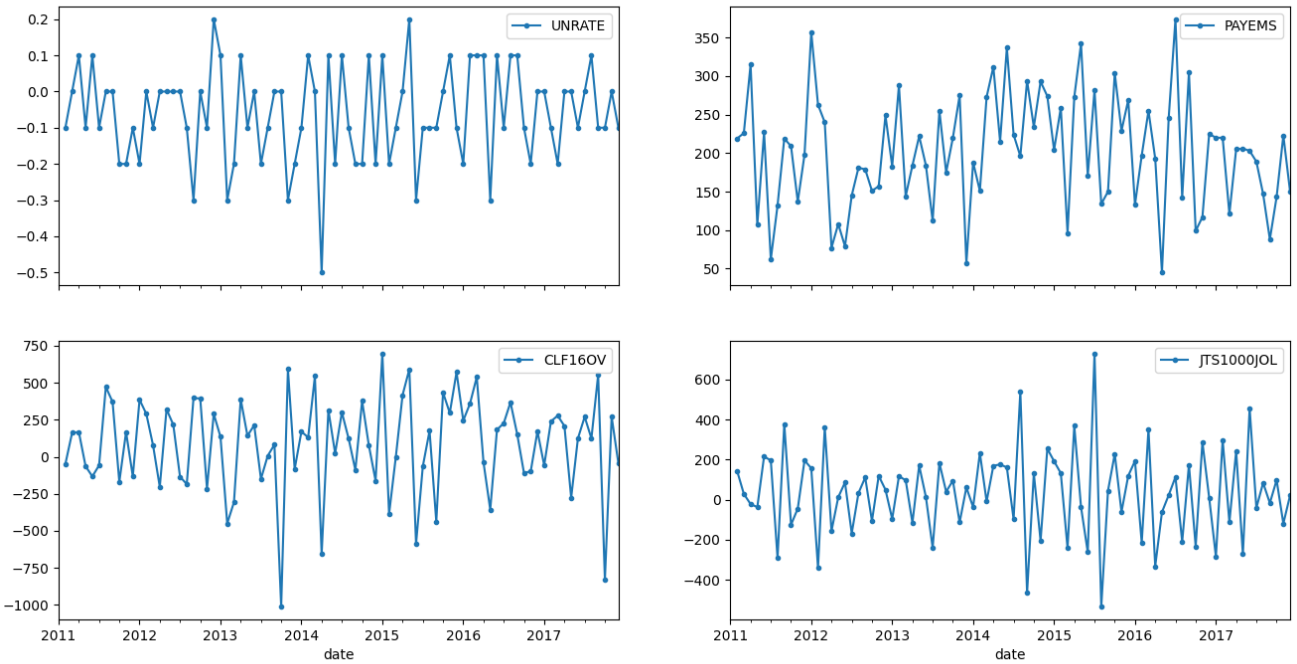


Figure 2.4: predictor variables, changes

I trained a few models on the series from 2011 to 2016 and then *predicted* the values for 2017. See the true vs. predicted values below.

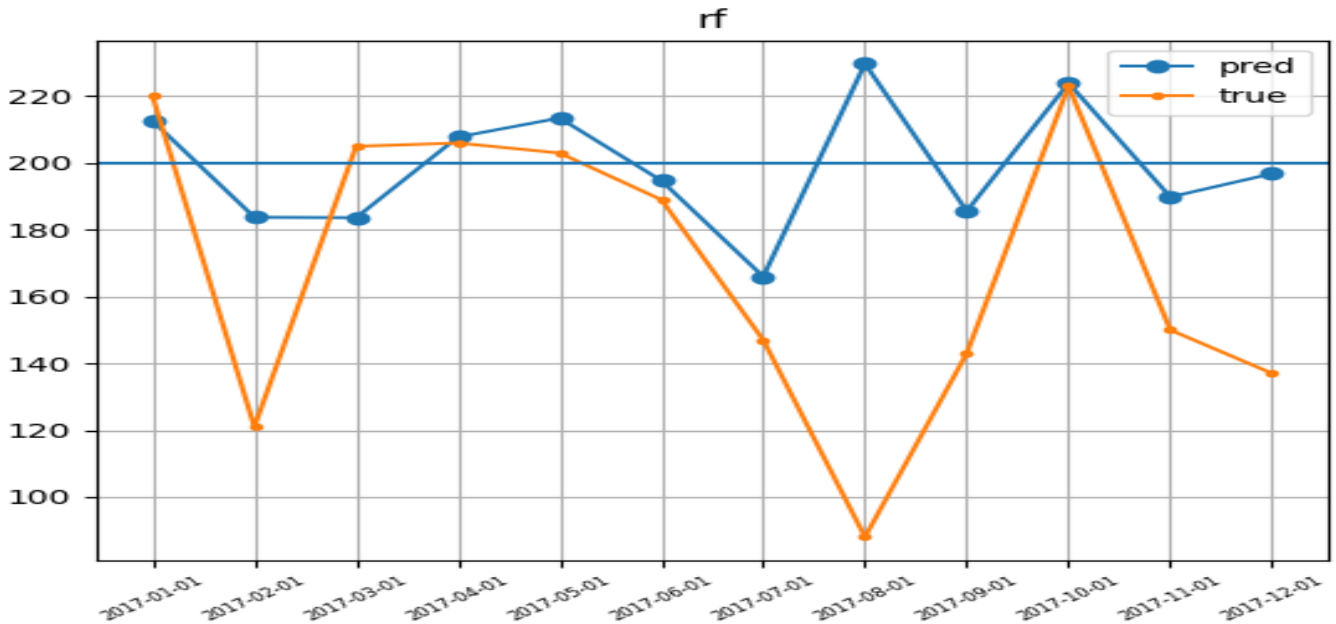


Figure 2.5: results using a random forest model

We can use the mean of the training data, 200, as a proxy for expectation. Are we accurately predicting above or below expectations?

To wrap up this section, I wanted to show that the data series is not always so regular. Below is the same series, but more historical data

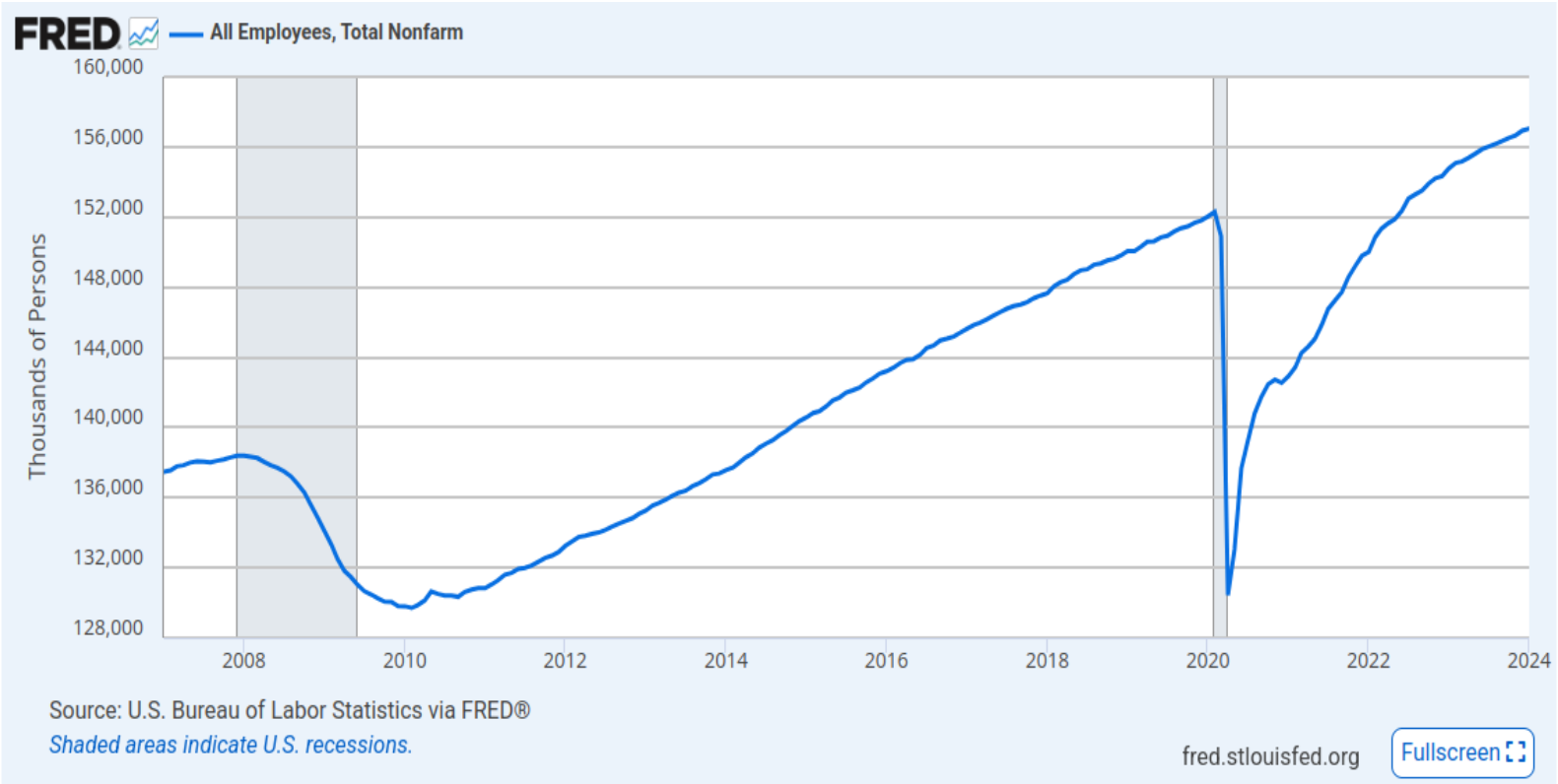


Figure 2.6: NonFarm Payroll level, 2008 - 2023

Embeddings and Recommendations

Embeddings: A mathematical/quantitative way of representing affinity of customers to products. Customers and products are assigned a list, typically a long list, of numbers. The embeddings are designed in such a way that things that are *similar* have lists that are nearby.

Given such an assignment, we can do magic like:

- find all customers similar to a given product
- find all products similar to a given product
- find all customers similar to a given customer

A Pytorch model to do embeddings is fairly simple since *embeddings* is a primitive in the framework/library.
See the code for the model below for 100 dimensional embeddings.

```
class MF(nn.Module):
    def __init__(self, n_users, n_movies, emb_size=100):
        super(MF, self).__init__()
        self.n_users = n_users
        self.n_movies = n_movies
        self.user_emb = nn.Embedding(n_users, emb_size)
        self.movie_emb = nn.Embedding(n_movies, emb_size)

        # initializing the matrices with a positive number
        # supposed to help generally will yield better results
        self.user_emb.weight.data.uniform_(0, 0.5)
        self.movie_emb.weight.data.uniform_(0, 0.5)

    def forward(self, users, movies):
        m = self.movie_emb(movies)
        u = self.user_emb(users)
        return (u * m).sum(1)  # taking the dot product
```

Below are the results of training the Pytorch model on Netflix movie review data. The plots show how loss changes given changes in the *hyper-parameters* embedding size, step size and training epochs.

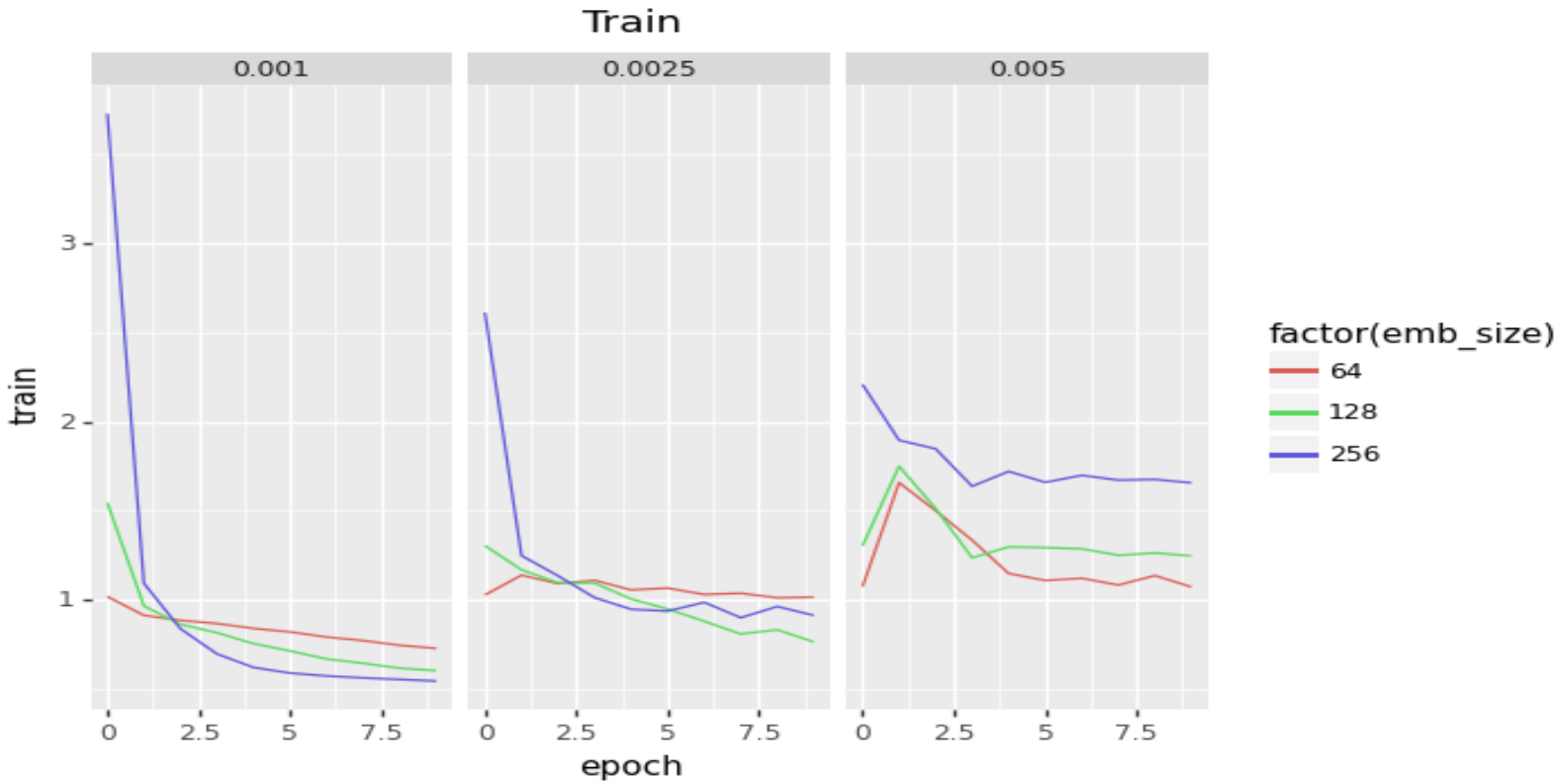


Figure 2.7: Loss as a function of epochs, step size and embedding size

and the validation set loss.

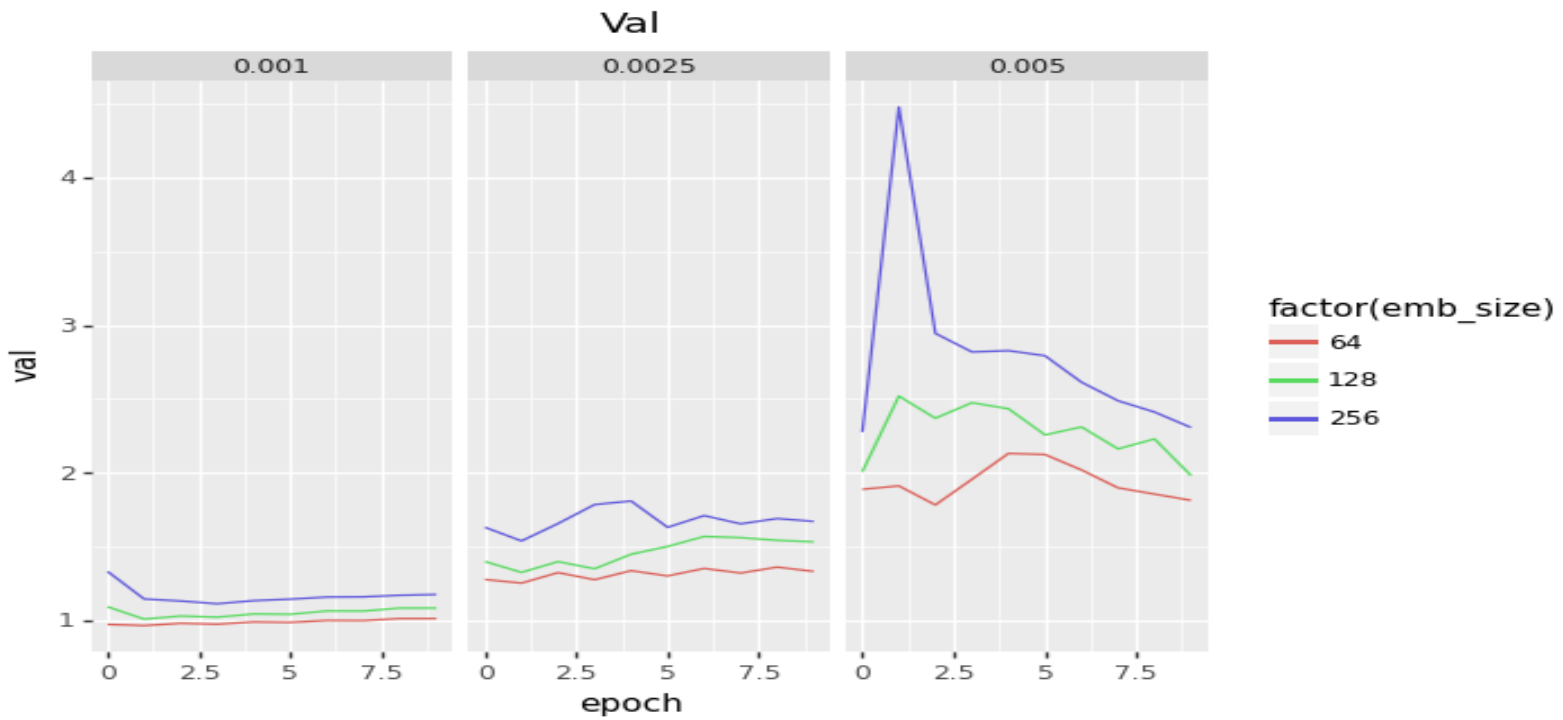


Figure 2.8: Validation loss as a function of hyper-parameters

Note that 64 dimensional embeddings are the best on the validation set.

Generative Artificial Intelligence

We cannot conclude without at least a brief mention of Generative AI.

Below are just some of the many, very practical applications of GenAI:

- Given a minimal set of *facts* related to interaction with a customer, generate a letter or an email to the customer,
- Given a set of product attributes, generate the product description for the website,
- Given a lengthy text document, generate a summary of the major points,
- Given the description of a process, find potential weaknesses.

These are situations regularly faced by knowledge workers. Having a tool to help with these means saving hours of workers time.

Example: given basic facts about problems on a trip, generate a complaint letter to the airline. The simple interface below collects the facts and also sets the *tone* for the letter.

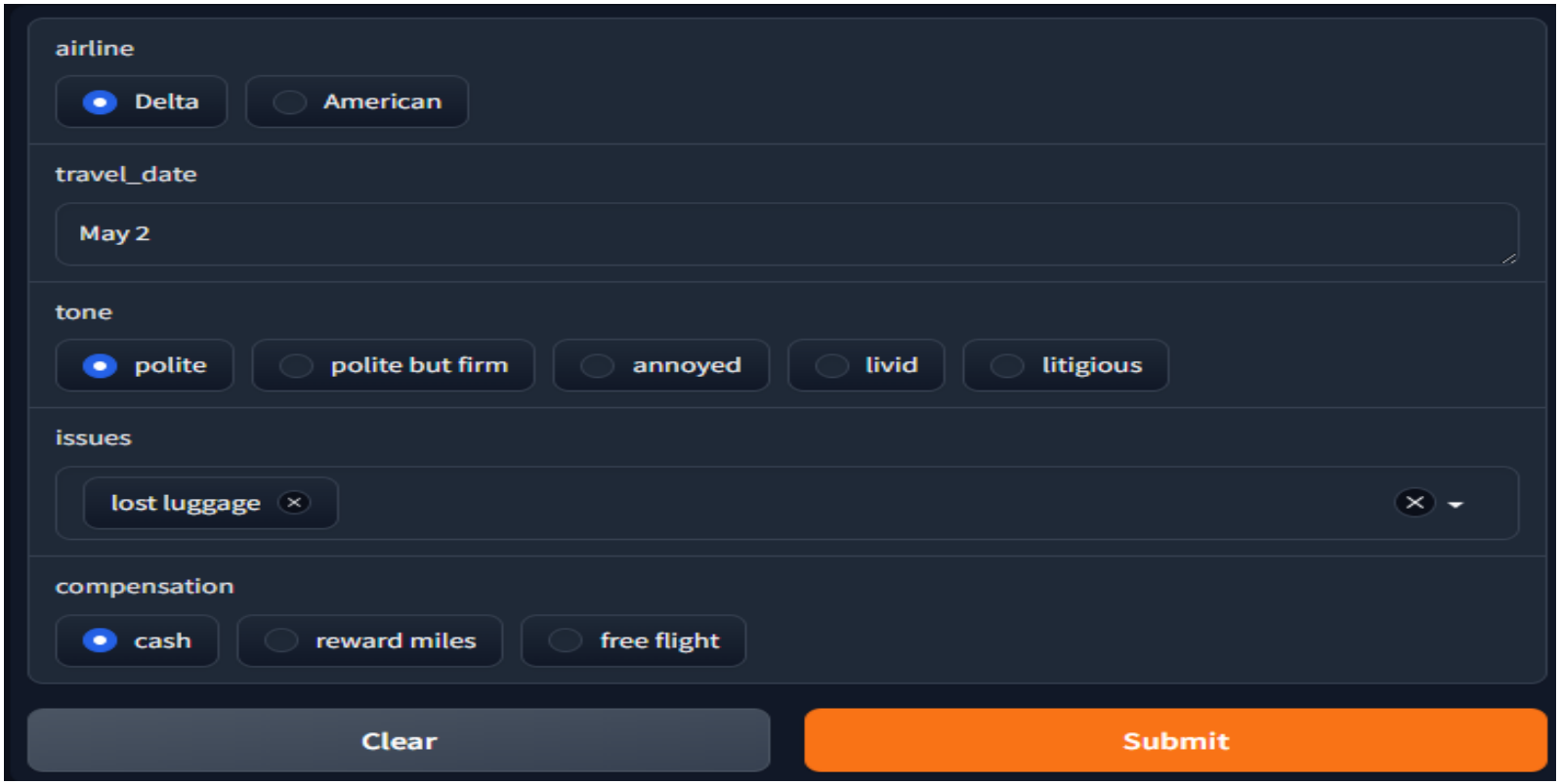
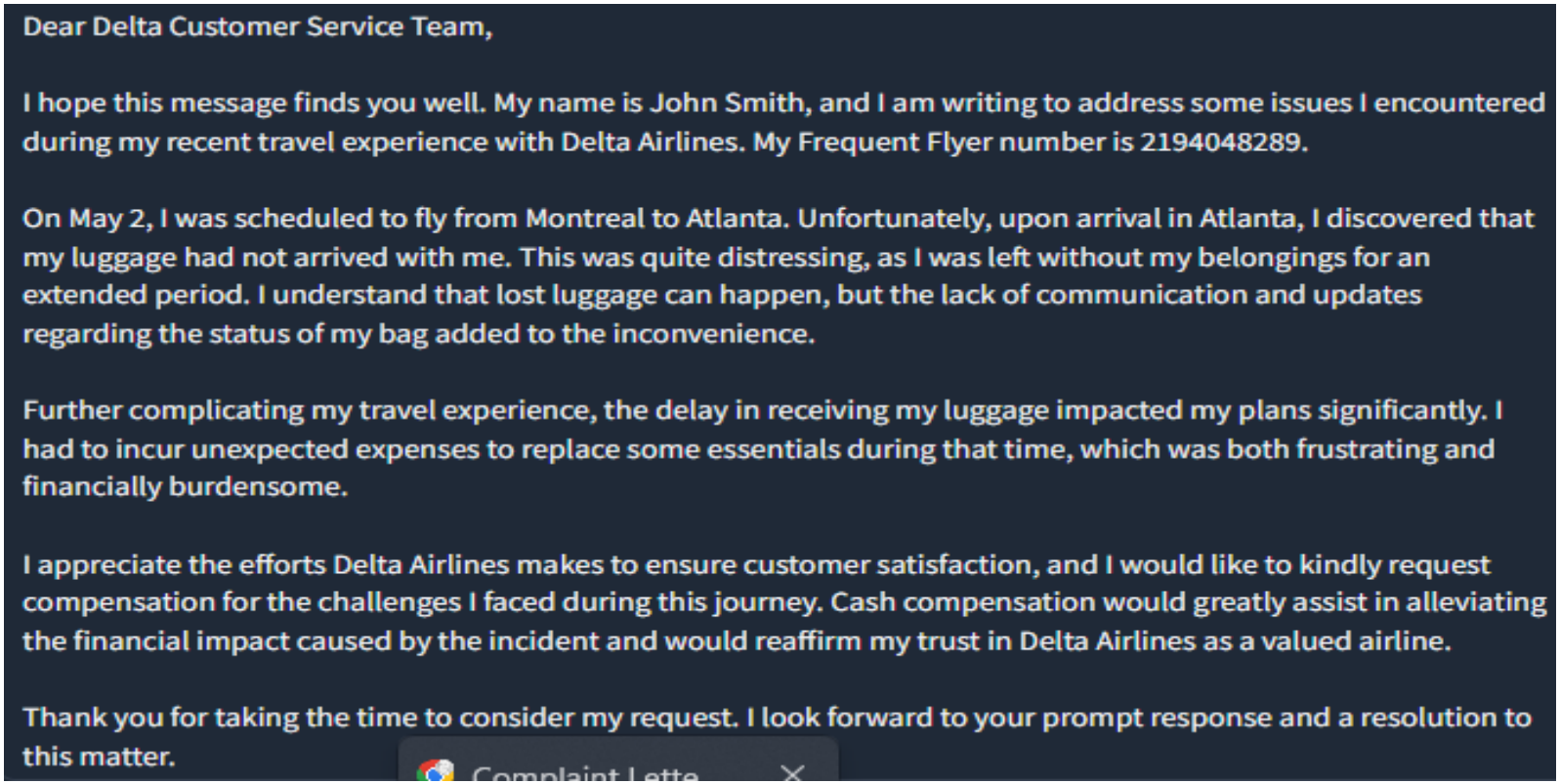


Figure 2.9: Interface to gather basic facts

The resulting letter with tone set to "polite".


Dear Delta Customer Service Team,

I hope this message finds you well. My name is John Smith, and I am writing to address some issues I encountered during my recent travel experience with Delta Airlines. My Frequent Flyer number is 2194048289.

On May 2, I was scheduled to fly from Montreal to Atlanta. Unfortunately, upon arrival in Atlanta, I discovered that my luggage had not arrived with me. This was quite distressing, as I was left without my belongings for an extended period. I understand that lost luggage can happen, but the lack of communication and updates regarding the status of my bag added to the inconvenience.

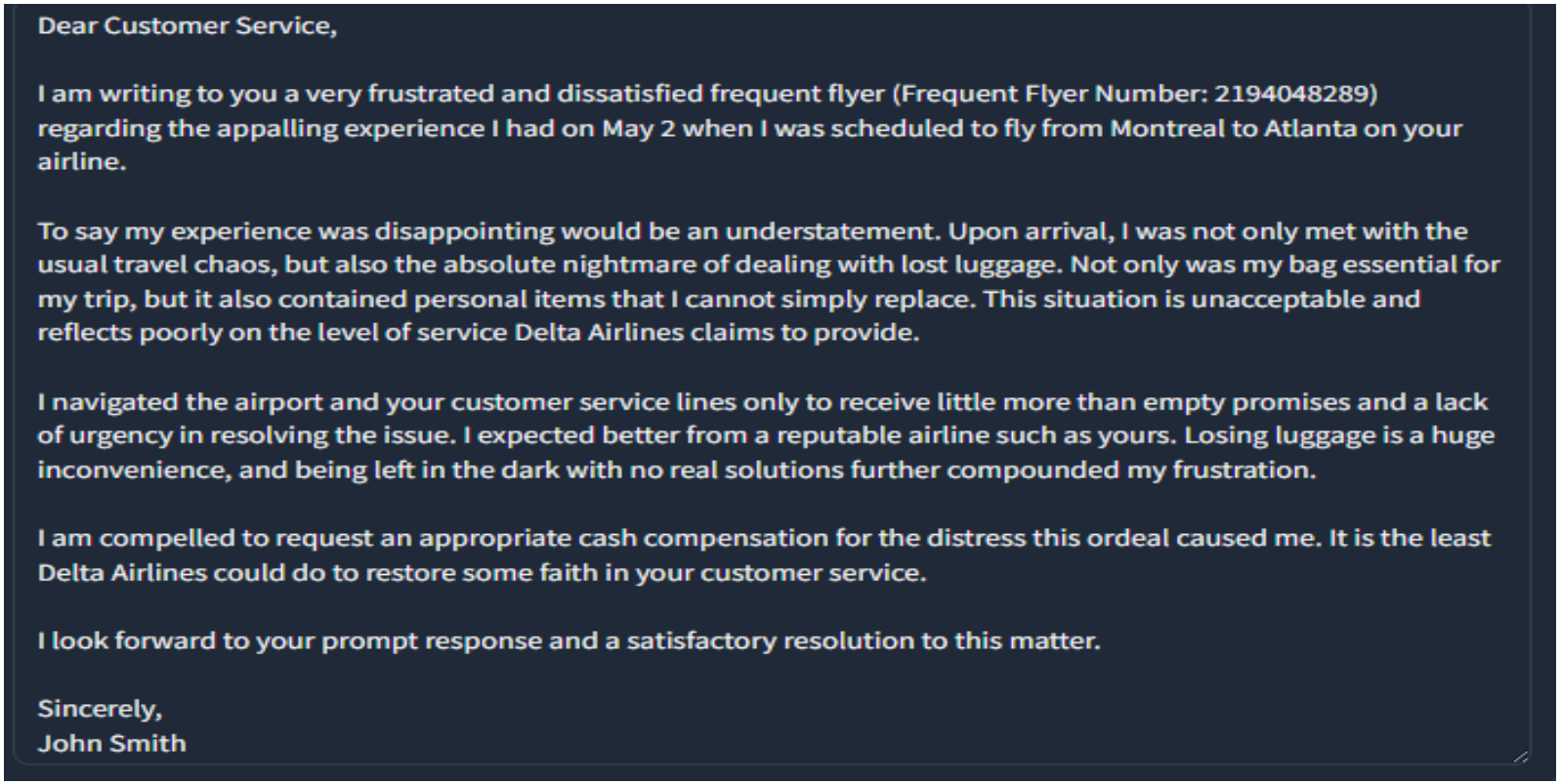
Further complicating my travel experience, the delay in receiving my luggage impacted my plans significantly. I had to incur unexpected expenses to replace some essentials during that time, which was both frustrating and financially burdensome.

I appreciate the efforts Delta Airlines makes to ensure customer satisfaction, and I would like to kindly request compensation for the challenges I faced during this journey. Cash compensation would greatly assist in alleviating the financial impact caused by the incident and would reaffirm my trust in Delta Airlines as a valued airline.

Thank you for taking the time to consider my request. I look forward to your prompt response and a resolution to this matter.

Figure 2.10: resulting "polite" letter

The resulting letter with tone set to "livid".


Dear Customer Service,

I am writing to you a very frustrated and dissatisfied frequent flyer (Frequent Flyer Number: 2194048289) regarding the appalling experience I had on May 2 when I was scheduled to fly from Montreal to Atlanta on your airline.

To say my experience was disappointing would be an understatement. Upon arrival, I was not only met with the usual travel chaos, but also the absolute nightmare of dealing with lost luggage. Not only was my bag essential for my trip, but it also contained personal items that I cannot simply replace. This situation is unacceptable and reflects poorly on the level of service Delta Airlines claims to provide.

I navigated the airport and your customer service lines only to receive little more than empty promises and a lack of urgency in resolving the issue. I expected better from a reputable airline such as yours. Losing luggage is a huge inconvenience, and being left in the dark with no real solutions further compounded my frustration.

I am compelled to request an appropriate cash compensation for the distress this ordeal caused me. It is the least Delta Airlines could do to restore some faith in your customer service.

I look forward to your prompt response and a satisfactory resolution to this matter.

Sincerely,
John Smith

Figure 2.11: resulting "livid" letter

For an example of summarization, click on the link below to see my an app that summarizes NPR news stories. The original story is typically 3 or more pages and the summary is about 2-3 paragraphs.

Link to summarization app

Please note a few things:

- You will almost certainly have to click on "Restart the space" as a first step.
- Please be patient, the AI only uses CPU processing, no GPU, so it is a bit slow. Note the "running" animation icon in the top right to see if it is running.

Conclusion and next steps

Let me help you solve your problems.

Next steps?

Conclusion and next steps

Thank you!

John H. Muller

jmuller.ics88@gtalumni.org

+1 (617) 669-2204