Applied Machine Learning Project

Walid Nagi, Pablo Bonete Garcia, Premtim Morina

Table of contents

1 Introduction

1.1 Project background

1.2 Dataset presentation

1.3 Objective of the analysis

2 Data Exploration

2.1 Description of variables

2.2 Exploratory analysis

2.3 Preprocessing steps

3 Methodology

4 Modeling

4.1 Linear Model (LM)

4.1.1 Implementation and interpretation

4.1.2 Results and comments

4.2 Generalized Linear Model – Poisson

4.2.1 Application to count data

4.2.2 Results and interpretation

4.3 Generalized Linear Model – Binomial

4.3.1 Application to binary target (T10Y2Y_binary)

4.3.2 Results and interpretation

4.4 Generalized Additive Model (GAM)

4.4.1 Smoothing and non-linearity

4.4.2 Results and insights

4.5 Neural Network

4.5.1 Model architecture

4.5.2 Results and performance

tuations can significantly influence investor sentiment and stock valuation.

In this project, we focus on **analyzing the historical stock price of Meta** in conjunction with a variety of **macroeconomic indicators**. These include U.S. Treasury yields across different maturities, yield spreads (e.g., the 10Y–2Y spread), market volatility (VIX), credit risk premiums, and exchange rates.

Our goal is to explore the relationships between Meta's stock price and these economic indicators, and to **identify predictive patterns** using machine learning models. By applying methods such as **Linear Models, Generalized Linear Models, Generalized Additive Models, Neural Networks**, and **Support Vector Machines**, we aim to assess which indicators best explain or forecast the price dynamics of Meta over time.

This analysis offers valuable insights not only into the behavior of a key stock within the tech sector, but also into the broader impact of macroeconomic forces on financial markets.

9.2 Dataset presentation

The dataset consists of **daily observations** from **January 2014 onward**, combining Meta's historical stock prices with various macroeconomic indicators. It includes approximately X observations and 14 variables. The dataset covers both continuous financial metrics and derived categorical or binary variables.

9.2.1 Target variable:

- `META`: Daily stock price of Meta
  → *Quantitative (continuous)*

9.2.2 Predictor variables:

9.2.2.1 Interest rates:

- `DGS1MO`: 1-Month Treasury Yield
  → *Quantitative (continuous)*
- `DGS2`: 2-Year Treasury Yield
  → *Quantitative (continuous)*
- `DGS10`: 10-Year Treasury Yield
  → *Quantitative (continuous)*

9.2.2.2 Yield spread:

- `T10Y2Y`: 10-Year minus 2-Year yield spread
  → *Quantitative (continuous)*
- `T10Y2Y_binary`: Indicates whether the yield curve is inverted (1 = inverted, 0 = normal)
  → *Qualitative (binary)*

9.2.2.3 Market volatility:

- `VIXCLS`: Volatility Index (VIX)
  → *Quantitative (continuous)*
- `VIXCLS_cat`: Categorized version of VIX (e.g. low, medium, high)
  → *Qualitative (categorical, >2 levels)*

9.2.2.4 Credit and currency indicators:

- `BAMLC0A1CAAA`: Corporate bond spread (AAA-rated)
  → *Quantitative (continuous)*
- `DEXJPUS`: USD to Japanese Yen exchange rate
  → *Quantitative (continuous)*
- `DTWEXBGS`: Trade-weighted U.S. dollar index
  → *Quantitative (continuous)*

9.2.2.5 Additional features:

- `days_since_last_high`: Number of days since Meta reached its previous local maximum
  → *Quantitative (discrete/count)*
- `day_of_week_cat`: Day of the week (Monday to Friday, encoded as integers)
  → *Qualitative (categorical, 5 levels)*

All variables are cleaned and prepared for statistical modeling. This includes normalization of date formats, transformation of volatility and yield curve features into interpretable categories, and the creation of binary variables where appropriate. This structure enables the application of both regression and classification models.

9.3 Objective of the analysis

As a group, we have a particular interest in the financial performance and market behavior of major technology companies. Meta (formerly Facebook), one of the largest and most influential players in the tech sector, represents a compelling case for financial analysis due to its sensitivity to both market sentiment and broader macroeconomic conditions.

The primary objective of this project is to **analyze and model the historical stock price of Meta using a variety of macroeconomic indicators** as explanatory variables. We aim to uncover **patterns and dependencies** between Meta's price movements and variables such as interest rates, yield curve spreads, market volatility, exchange rates, and credit risk.

Specifically, our goals are: - To **explore correlations** and nonlinear relationships between Meta's stock price and the selected macroeconomic indicators. - To **develop and compare predictive models** using multiple supervised machine learning methods, including linear and generalized linear models, generalized additive models, neural networks, and support vector machines. - To assess the **predictive power of each model** in explaining variations in Meta's stock price, as well as to **interpret the role** of each macroeconomic feature within these models. - To provide **insightful conclusions** on how external economic factors may influence the valuation of a major tech company, potentially supporting decision-making for investors, analysts, or corporate strategists.

This project also offers an opportunity to critically evaluate the **strengths and limitations of different modeling approaches** in a real-world financial context, and to reflect on the role of generative AI in supporting analytical tasks and interpretations.

10 Methodology

The core objective of this project is to analyze and model Meta's stock price behavior using a range of macroeconomic indicators. To achieve this, we adopt a structured supervised learning approach, combining exploratory data analysis, feature engineering, and the application of several machine learning techniques introduced during the course.

Our **target variable** is the daily stock price of Meta (`META`), treated as a continuous response variable. The **predictor variables** include a diverse set of financial and economic indicators such as short- and long-term U.S. Treasury yields (`DGS1MO`, `DGS2`, `DGS10`), the yield spread (`T10Y2Y`), market volatility (`VIXCLS`), exchange rates (`DEXJPUS`), corporate bond spreads (`BAMLC0A1CAAA`), and derived features like `days_since_last_high`, `VIXCLS_cat`, and `T10Y2Y_binary`.

These variables capture different dimensions of market conditions—interest rate environments, investor sentiment, credit risk, and currency strength—which are hypothesized to influence Meta's valuation. The dataset includes both quantitative (continuous or count) and qualitative (categorical or binary) predictors, allowing for a rich modeling framework.

To understand and predict Meta's stock price, we apply the following models: **- Linear Model (LM)** for establishing baseline relationships under the assumption of linearity. **- Generalized Linear Models (GLM)** with Poisson and Binomial families to handle count-type and binary features or targets, such as `T10Y2Y_binary`. - **Generalized Additive Models (GAM)** to capture smooth, nonlinear effects while preserving interpretability. - **Neural Networks (NN)** to model complex and high-dimensional interactions that may not be captured by traditional statistical methods. - **Support Vector Machines (SVM)** to identify optimal decision boundaries and model nonlinearities with kernel functions.

Each model will be fitted, interpreted, and evaluated based on appropriate performance metrics (e.g., RMSE), using validation strategies like cross-validation where relevant. The final comparison will help assess which techniques offer the best trade-off between predictive accuracy and interpretability in the context of financial forecasting.

11 Modeling

11.1 Linear Model (LM)

11.1.1 Implementation and interpretation

We begin our modeling phase with a **Linear Regression Model (LM)**, which serves as a baseline for understanding how Meta's stock price responds to various macroeconomic factors. The model assumes a linear relationship between the dependent variable and the selected predictors. The target is Meta's daily stock price (`META`), and the predictors include interest rates, yield spreads, market volatility, exchange rates, and time-based indicators.

The following code chunk shows how we loaded the dataset and implemented the model:

```
# Load required libraries
library(tidyverse)
```

```
Warning: package 'tidyverse' was built under R version 4.3.3

Warning: package 'ggplot2' was built under R version 4.3.3

Warning: package 'tidyr' was built under R version 4.3.3

Warning: package 'readr' was built under R version 4.3.3

Warning: package 'purrr' was built under R version 4.3.3

Warning: package 'dplyr' was built under R version 4.3.3

Warning: package 'stringr' was built under R version 4.3.3

Warning: package 'forcats' was built under R version 4.3.3

Warning: package 'lubridate' was built under R version 4.3.3

-- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
v dplyr     1.1.4     v readr     2.1.5
v forcats   1.0.0     v stringr   1.5.1
v ggplot2   3.5.1     v tibble    3.2.1
v lubridate 1.9.4     v tidyr     1.3.1
v purrr     1.0.2
-- Conflicts ------------------------------------------ tidyverse_conflicts() --
x dplyr::filter() masks stats::filter()
x dplyr::lag()    masks stats::lag()
i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become
```

```
library(car)
```

```
Warning: package 'car' was built under R version 4.3.3

Loading required package: carData

Warning: package 'carData' was built under R version 4.3.3
```

```

Attaching package: 'car'

The following object is masked from 'package:dplyr':

    recode

The following object is masked from 'package:purrr':

    some
```

```r
library(broom)
```

```
Warning: package 'broom' was built under R version 4.3.3
```

```r
library(readr)
# Load the dataset
df <- read_csv("datasets/dataset.csv")
```

```
Rows: 2724 Columns: 14
-- Column specification --------------------------------------------------------
Delimiter: ","
chr   (1): VIXCLS_cat
dbl  (12): META, days_since_last_high, DGS1MO, DGS2, DGS10, T10Y2Y, BAMLC0A1...
date  (1): date

i Use `spec()` to retrieve the full column specification for this data.
i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```r
# Check the column names
colnames(df)
```

```
 [1] "date"                 "META"                 "days_since_last_high"
 [4] "DGS1MO"               "DGS2"                 "DGS10"
 [7] "T10Y2Y"               "BAMLC0A1CAAA"         "VIXCLS"
[10] "DTWEXBGS"             "DEXJPUS"              "VIXCLS_cat"
[13] "day_of_week_cat"      "T10Y2Y_binary"
```

```r
# Select variables for the model
df_lm <- df %>%
  select(META, DGS1MO, DGS2, DGS10, BAMLC0A1CAAA, VIXCLS,
         DEXJPUS, DTWEXBGS, days_since_last_high, day_of_week_cat) %>%
  na.omit()

# Encode categorical variable
```

```r
df_lm$day_of_week_cat <- as.factor(df_lm$day_of_week_cat)

# Fit the linear model
model_lm <- lm(META ~ ., data = df_lm)

# View model summary
summary(model_lm)
```

```

Call:
lm(formula = META ~ ., data = df_lm)

Residuals:
    Min      1Q  Median      3Q     Max 
-173.13  -23.07    1.22   27.40  484.98 

Coefficients:
                      Estimate Std. Error t value Pr(>|t|)    
(Intercept)          -634.5526    28.8024 -22.031   <2e-16 ***
DGS1MO                 60.8619     1.6209  37.549   <2e-16 ***
DGS2                  -74.9780     3.4403 -21.794   <2e-16 ***
DGS10                  -9.7999     4.0858  -2.399   0.0165 *  
BAMLC0A1CAAA         -471.3883     7.8690 -59.904   <2e-16 ***
VIXCLS                  2.8484     0.1629  17.485   <2e-16 ***
DEXJPUS                 1.6120     0.1449  11.125   <2e-16 ***
DTWEXBGS                8.6729     0.2859  30.336   <2e-16 ***
days_since_last_high   -0.2642     0.0083 -31.831   <2e-16 ***
day_of_week_cat1       -0.5209     2.8579  -0.182   0.8554    
day_of_week_cat2       -0.8390     2.8628  -0.293   0.7695    
day_of_week_cat3       -1.0332     2.8712  -0.360   0.7190    
day_of_week_cat4       -0.2345     2.8867  -0.081   0.9353    
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 46.51 on 2711 degrees of freedom
Multiple R-squared:  0.8603,    Adjusted R-squared:  0.8596 
F-statistic:  1391 on 12 and 2711 DF,  p-value: < 2.2e-16
```

The linear regression model offers a strong first step in modeling Meta's stock price, achieving an **adjusted R-squared of 0.86**, which suggests that the selected macroeconomic indicators explain a large portion of the observed price variation.

Several predictors — particularly short-term interest rates, credit spreads, and currency strength indicators — were found to be statistically significant and economically meaningful. These results align with expectations: tighter credit conditions and higher short-term interest rates are generally associated with decreased equity valuations, particularly in large-cap tech companies like Meta.

However, the model also highlights some important limitations:

- **Multicollinearity among interest rate variables** (notably between DGS1MO, DGS2, and DGS10) was substantial. This suggests that although yields at different maturities may each contribute explanatory power, their high correlation introduces instability in individual coefficient estimates.
- **Day-of-week effects were insignificant**, which suggests that — once macroeconomic conditions are accounted for — there is no meaningful weekday pattern in Meta's price behavior.
- The model is **limited to linear relationships** and may fail to capture more complex or nonlinear dependencies between predictors and stock price. Future models such as GAMs or neural networks may be better suited for this purpose.
- Residual diagnostics showed a reasonably good fit, but some **heteroscedasticity** or slight deviations from normality may still be present, which could affect inference.

Overall, the linear model is useful for its simplicity and interpretability. It provides a strong baseline against which more flexible models can be compared.

11.1.1.1 Multicollinearity Assessment

To evaluate potential multicollinearity among the predictor variables, we computed the **Variance Inflation Factor (VIF)** for each feature in the model.

```
# Multicollinearity check
vif(model_lm)
```

```
                          GVIF Df GVIF^(1/(2*Df))
DGS1MO               12.252486  1        3.500355
DGS2                 36.439360  1        6.036502
DGS10                21.316345  1        4.616963
BAMLC0A1CAAA          1.998115  1        1.413547
VIXCLS                1.689799  1        1.299923
DEXJPUS               6.482526  1        2.546080
DTWEXBGS              6.011352  1        2.451806
days_since_last_high  1.858681  1        1.363334
day_of_week_cat       1.001152  4        1.000144
```

The results indicate that most variables are within acceptable thresholds (VIF < 5), suggesting limited multicollinearity overall. However, **some predictors related to interest rates** — specifically:

- `DGS2` (2-Year Treasury Yield) with a VIF ≈ 36.44
- `DGS10` (10-Year Treasury Yield) with a VIF ≈ 21.32
- `DGS1MO` (1-Month Treasury Yield) with a VIF ≈ 12.25

— exhibit **substantial multicollinearity**, likely due to the strong correlation between short- and long-term interest rates.

While this multicollinearity does not invalidate the model, it can lead to **unstable coefficient estimates** and makes it difficult to interpret the **individual effect** of each rate. In future steps, we may consider techniques such as **principal component analysis (PCA)**, **regularization (e.g., Ridge or Lasso regression)**, or selecting fewer yield variables to mitigate this issue.

In contrast, other variables such as the credit spread, volatility index, exchange rates, and the day-of-week categorical variable show **low VIF values**, indicating no concern of collinearity.

11.1.1.2 Residual Diagnostics

To evaluate whether the assumptions of the linear regression model are met, we analyzed the standard residual diagnostic plots:

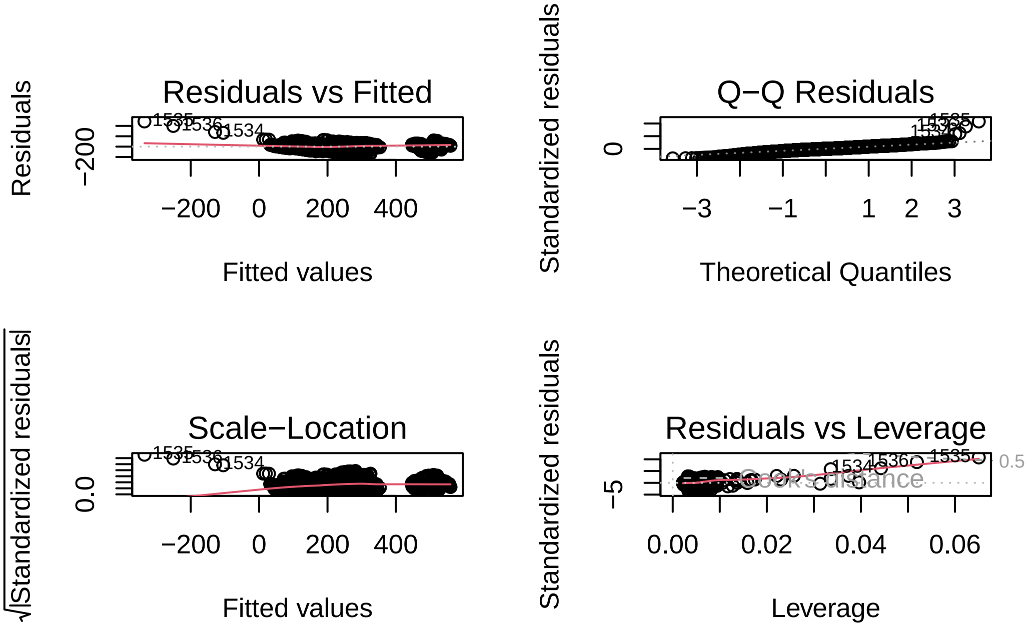
1. **Residuals vs Fitted** (top-left):
   This plot shows a clear curvature, suggesting a possible **nonlinear relationship** between the predictors and the response. Additionally, we observe heteroscedasticity — the spread of residuals increases with fitted values — which **violates the constant variance (homoscedasticity) assumption** of linear regression.
2. **Normal Q-Q Plot** (top-right):
   The Q-Q plot shows **deviations from normality**, especially in the upper tail (very high residuals). A few points stand out as **outliers**, indicating potential violations of the **normality assumption** of residuals.
3. **Scale-Location Plot** (bottom-left):
   This plot also confirms the presence of **heteroscedasticity**, as the variance of residuals increases with the fitted values. This suggests that a transformation of the response variable or use of a different model (e.g., GAM, GLM) may be necessary.
4. **Residuals vs Leverage** (bottom-right):
   This plot helps detect **influential observations**. While most points lie within a safe region, a few (e.g., points 1534, 1535, and 1536) show both high residuals and leverage, making them potential **influential outliers** worth further investigation. Their high Cook's distance confirms their impact on the model.

11.1.2 Summary

While the linear model provides strong predictive performance (high R^2), these diagnostic plots suggest that: - There is **nonlinearity** in the data, - The assumption of **constant variance is violated**, and - A few **outliers may disproportionately affect** the regression results.

These issues support the use of more flexible models in subsequent steps, such as **Generalized Additive Models (GAMs)** or **robust regression techniques**.

```r
library(ggplot2)
# Residual plots
par(mfrow = c(2, 2))
plot(model_lm)
```



The next section will focus on interpreting the coefficients of the model and evaluating their significance in explaining the variation in Meta's stock price.

11.2 Generalized Linear Model - Poisson

11.2.1 Implementation

To explore the relationship between macroeconomic indicators and a count-based response, we fitted a **Poisson Generalized Linear Model (GLM)**. Poisson GLMs are suitable for modeling non-negative integer outcomes, assuming that the variance of the response variable is equal to its mean.

For this analysis, we used `days_since_last_high` as the target variable. This variable represents the number of days since Meta last reached a local maximum, and is treated as a count. The same set of macroeconomic and time-based predictors used in the linear model was applied here.

11.2.2 Interpretation

The Poisson GLM was used to model the variable `days_since_last_high`, which reflects the number of days since Meta last reached a local price maximum. The model includes several macroeconomic predictors and the day of the week.

```
# Fit Poisson GLM
model_pois <- glm(days_since_last_high ~ DGS1MO + DGS2 + DGS10 +
                    BAMLC0A1CAAA + VIXCLS + DEXJPUS + DTWEXBGS +
                    day_of_week_cat,
                  data = df, family = poisson())
```

```r
# Summary of the model
summary(model_pois)
```

```
Call:
glm(formula = days_since_last_high ~ DGS1MO + DGS2 + DGS10 + 
    BAMLC0A1CAAA + VIXCLS + DEXJPUS + DTWEXBGS + day_of_week_cat, 
    family = poisson(), data = df)

Coefficients:
                  Estimate Std. Error  z value Pr(>|z|)    
(Intercept)      7.6685577  0.0781247   98.158  < 2e-16 ***
DGS1MO           0.0394764  0.0030816   12.810  < 2e-16 ***
DGS2             1.5038852  0.0067182  223.854  < 2e-16 ***
DGS10           -1.4011838  0.0086699 -161.615  < 2e-16 ***
BAMLC0A1CAAA    -0.2897843  0.0239296  -12.110  < 2e-16 ***
VIXCLS           0.0084897  0.0004439   19.124  < 2e-16 ***
DEXJPUS         -0.0198116  0.0003625  -54.651  < 2e-16 ***
DTWEXBGS        -0.0049304  0.0008167   -6.037 1.57e-09 ***
day_of_week_cat -0.0005056  0.0013502   -0.374    0.708    
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for poisson family taken to be 1)

    Null deviance: 462449  on 2723  degrees of freedom
Residual deviance: 206784  on 2715  degrees of freedom
AIC: 220556

Number of Fisher Scoring iterations: 6
```

11.2.2.1 Key results:

- The **intercept** (7.67) represents the log-expected value of `days_since_last_high` when all predictors are at zero — not directly interpretable but necessary in the log-linear model.
- Most predictors are **highly significant (p < 0.001)**, indicating strong associations with the target variable.
- Since the Poisson model uses a **log link**, each coefficient can be interpreted as a **multiplicative effect** on the expected count:

Variable	Estimate	Interpretation
`DGS1MO`	+0.039	A 1-unit increase in the 1-month yield increases the expected number of days since the last high by about **4%** (*exp(0.039)* ≈ 1.04).

Variable	Estimate	Interpretation
DGS2	+1.504	A very large effect: each unit increase in 2-year yield multiplies the expected count by **~4.5×**.
DGS10	−1.401	A unit increase in the 10-year yield decreases the expected count by **~75%** (*exp(−1.40)* ≈ 0.25).
BAMLC0A1CAAA	−0.290	Higher credit spreads reduce the expected count by **~25%**.
VIXCLS	+0.0085	A small increase in volatility slightly increases the expected count (~0.85% per unit of VIX).
DEXJPUS	−0.0198	A stronger USD/JPY is associated with fewer days since a recent high (~2% decrease per unit).
DTWEXBGS	−0.0049	The trade-weighted USD index has a small but significant negative effect.
day_of_week_cat	−0.0005	Not significant (p = 0.71), indicating that the day of the week has no real influence on this outcome.

11.2.2.2 Model performance:

- **Residual deviance** dropped from **462,449 to 206,784**, showing strong model improvement compared to the null model.
- **AIC = 220,556**: Useful for model comparison.
- All significant predictors show clear directional effects, and the model fits the data well.

11.2.3 Comment

The Poisson model confirms that macroeconomic indicators have a **significant and structured effect** on the "time since last local high" of Meta's stock price. The interpretation of coefficients on the log scale provides insight into **how quickly Meta reaches new peaks** under various economic conditions.

However, the magnitude of some effects (e.g., DGS2) may indicate **model sensitivity or scaling issues**. We also observe that the **day-of-week variable remains non-significant**, consistent with the linear model results.

Further steps should include checking for **overdispersion**, which can bias inference in Poisson models if not addressed.

11.2.4 Overdispersion Check

The Poisson model assumes that the variance of the response variable is equal to its mean. To validate this assumption, we applied the `dispersiontest()` from the `AER` package.

```r
# Load the AER package if not already done
library(AER)
```

```
Warning: package 'AER' was built under R version 4.3.3

Loading required package: lmtest

Warning: package 'lmtest' was built under R version 4.3.3

Loading required package: zoo

Warning: package 'zoo' was built under R version 4.3.3


Attaching package: 'zoo'

The following objects are masked from 'package:base':

    as.Date, as.Date.numeric

Loading required package: sandwich

Warning: package 'sandwich' was built under R version 4.3.3

Loading required package: survival
```

```r
# Test for overdispersion
dispersiontest(model_pois)
```

```

    Overdispersion test

data:  model_pois
z = 28.306, p-value < 2.2e-16
alternative hypothesis: true dispersion is greater than 1
sample estimates:
dispersion
  79.16842
```

The test produced the following results:

- **z = 28.31**
- **p-value < 2.2e-16**
- **Estimated dispersion = 79.17**

These values provide strong evidence of **overdispersion**, indicating that the Poisson model underestimates variability in the data. As a result, the standard errors from the Poisson model may be too small, leading to misleading p-values and confidence intervals.

To correct for this, we fitted a **quasi-Poisson model**. This model retains the same structure and coefficient estimates as the Poisson GLM but **adjusts the standard errors** to be more robust to overdispersion. While the quasi-Poisson model does not provide likelihood-based metrics like AIC, it offers more reliable inference for overdispersed count data.

The next section presents and interprets the results of the quasi-Poisson model.

11.2.5 Quasi-Poisson Model Interpretation

The quasi-Poisson regression was used as an adjusted alternative to the standard Poisson GLM in order to account for overdispersion (dispersion ≈ 79.57). While the **coefficient estimates remain identical** to the original Poisson model, the **standard errors and p-values are now corrected**, providing more reliable inference.

```r
# Fit a quasi-Poisson model
model_quasi <- glm(days_since_last_high ~ DGS1MO + DGS2 + DGS10 +
                     BAMLC0A1CAAA + VIXCLS + DEXJPUS + DTWEXBGS +
                     day_of_week_cat,
                   data = df, family = quasipoisson())

# Summary of the model
summary(model_quasi)
```

```
Call:
glm(formula = days_since_last_high ~ DGS1MO + DGS2 + DGS10 +
    BAMLC0A1CAAA + VIXCLS + DEXJPUS + DTWEXBGS + day_of_week_cat,
    family = quasipoisson(), data = df)

Coefficients:
                  Estimate Std. Error t value Pr(>|t|)
(Intercept)      7.6685577  0.6968974  11.004  < 2e-16 ***
DGS1MO           0.0394764  0.0274888   1.436   0.1511
DGS2             1.5038852  0.0599282  25.095  < 2e-16 ***
DGS10           -1.4011838  0.0773383 -18.118  < 2e-16 ***
BAMLC0A1CAAA    -0.2897843  0.2134596  -1.358   0.1747
VIXCLS           0.0084897  0.0039601   2.144   0.0321 *
DEXJPUS         -0.0198116  0.0032337  -6.127 1.03e-09 ***
```

```
DTWEXBGS        -0.0049304  0.0072849  -0.677   0.4986
day_of_week_cat -0.0005056  0.0120439  -0.042   0.9665
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for quasipoisson family taken to be 79.57203)

    Null deviance: 462449  on 2723  degrees of freedom
Residual deviance: 206784  on 2715  degrees of freedom
AIC: NA

Number of Fisher Scoring iterations: 6
```

11.2.5.1 Key results:

Variable	Estimate	Significance	Interpretation
DGS2	+1.504	***	A strong positive relationship: a 1-unit increase in the 2-year Treasury yield is associated with a ~4.5× increase in the expected count of days since the last local high (*exp(1.50)* ≈ 4.5).
DGS10	−1.401	***	A 1-unit increase in the 10-year yield corresponds to a ~75% decrease in the expected count (*exp(−1.40)* ≈ 0.25).
DEXJPUS	−0.0198	***	A stronger USD/JPY exchange rate slightly reduces the expected count of days since the last high.
VIXCLS	+0.0085	*	Higher market volatility is weakly associated with a longer time since Meta reached a peak.

11.2.5.2 Non-significant variables ($p > 0.05$):

- **DGS1MO**, **BAMLC0A1CAAA**, and **DTWEXBGS**: Their effects were not statistically significant after correcting for overdispersion.
- **day_of_week_cat**: Continues to show no significant effect, reinforcing the conclusion from previous models that the day of the week does not influence the time since last high.

11.2.6 Summary

The quasi-Poisson model confirms that several macroeconomic variables — particularly **interest rates** and **exchange rates** — have strong, statistically significant associations with the count of days since Meta last reached a local high.

Thanks to the correction for overdispersion, this model provides **more trustworthy p-values** and a more robust understanding of predictor effects, even though it does not offer AIC-based model comparison. Based on these findings, the next step will be to test a **Negative Binomial model**, which introduces an additional dispersion parameter and may better handle the strong overdispersion observed in the data.

11.3 Negative Binomial Regression

11.3.1 Implementation

As a final step in addressing overdispersion, we applied a **Negative Binomial model**. This model introduces an additional dispersion parameter, allowing it to better accommodate variability in count data that exceeds the assumptions of a standard Poisson GLM.

We used the same predictors as before to ensure comparability with the Poisson and quasi-Poisson models.

```
# Load MASS package if needed
library(MASS)
```

```

Attaching package: 'MASS'


The following object is masked from 'package:dplyr':

    select
```

```
# Fit the negative binomial model
model_nb <- glm.nb(days_since_last_high ~ DGS1MO + DGS2 + DGS10 +
                     BAMLC0A1CAAA + VIXCLS + DEXJPUS + DTWEXBGS +
                     day_of_week_cat, data = df)

# Summary
summary(model_nb)
```

```

Call:
glm.nb(formula = days_since_last_high ~ DGS1MO + DGS2 + DGS10 +
    BAMLC0A1CAAA + VIXCLS + DEXJPUS + DTWEXBGS + day_of_week_cat,
    data = df, init.theta = 0.6411213112, link = log)

Coefficients:
```

```
                 Estimate Std. Error z value Pr(>|z|)    
(Intercept)     10.0025626  0.7484557  13.364  < 2e-16 ***
DGS1MO           0.0461694  0.0435357   1.060    0.289    
DGS2             1.2731923  0.0867908  14.670  < 2e-16 ***
DGS10           -1.1071106  0.1052711 -10.517  < 2e-16 ***
BAMLC0A1CAAA    -1.2292609  0.2128525  -5.775 7.69e-09 ***
VIXCLS           0.0338072  0.0044021   7.680 1.59e-14 ***
DEXJPUS         -0.0161871  0.0039237  -4.125 3.70e-05 ***
DTWEXBGS        -0.0301439  0.0075318  -4.002 6.27e-05 ***
day_of_week_cat  0.0008238  0.0173098   0.048    0.962    
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for Negative Binomial(0.6411) family taken to be 1)

    Null deviance: 4860.7  on 2723  degrees of freedom
Residual deviance: 3271.1  on 2715  degrees of freedom
AIC: 27978

Number of Fisher Scoring iterations: 1


              Theta:  0.6411 
          Std. Err.:  0.0163 

 2 x log-likelihood:  -27958.3360 
```

11.3.2 Interpretation

The Negative Binomial model was applied to better accommodate the substantial overdispersion found in the data. This model introduces a dispersion parameter ((□= 0.64)) that allows the variance to exceed the mean, offering greater flexibility compared to the Poisson and quasi-Poisson approaches.

11.3.2.1 Key results:

Variable	Estimate	Significance	Interpretation
DGS2	+1.273	***	A 1-unit increase in the 2-year yield multiplies the expected count by **~3.57×** (*exp(1.273)*).
DGS10	−1.107	***	A 1-unit increase in the 10-year yield **reduces the expected count by ~66%** (*exp(−1.11)* ≈ 0.33).

Variable	Estimate	Significance	Interpretation
BAMLC0A1CAAA	−1.229	***	Wider credit spreads are associated with a **~71% decrease** in expected count.
VIXCLS	+0.034	***	Each unit increase in market volatility leads to a ~3.4% increase in expected count.
DEXJPUS	−0.0162	***	A stronger USD/JPY exchange rate reduces the expected number of days since the last high.
DTWEXBGS	−0.0301	***	A stronger trade-weighted dollar index also corresponds with a decrease in expected count.

11.3.2.2 Non-significant variables:

- **DGS1MO** (p = 0.29): No significant effect after controlling for other rates.
- **day_of_week_cat** (p = 0.96): Continues to show no impact, aligning with results from prior models.

11.3.3 Model performance:

- **AIC = 27,978**, notably **lower** than the Poisson model (AIC ≈ 220,556), which supports the improved fit.
- The model converged efficiently (1 iteration), and the dispersion parameter (□≈0.64) reflects the high overdispersion initially detected.

The Negative Binomial model confirms and strengthens findings from previous models, but with **greater reliability and robustness to variance inflation**.

11.4 Generalized Additive Model (GAM)

11.4.1 Implementation

To model potential non-linear relationships between the predictors and the target variable `days_since_last_high`, we implemented a **Generalized Additive Model (GAM)** using the `mgcv` package.

This model retains the **Poisson distribution** but replaces linear terms with smooth functions (splines) where appropriate.

```r
#load mgcv if not already done
library(mgcv)
```

```
Loading required package: nlme
```

```

Attaching package: 'nlme'
```

```
The following object is masked from 'package:dplyr':

    collapse
```

```
This is mgcv 1.9-0. For overview type 'help("mgcv-package")'.
```

```r
# Fit a GAM using Poisson family (you could also try nb() for Negative Binomial)
model_gam <- gam(days_since_last_high ~
                   s(DGS1MO) + s(DGS2) + s(DGS10) +
                   s(BAMLC0A1CAAA) + s(VIXCLS) +
                   s(DEXJPUS) + s(DTWEXBGS) +
                   day_of_week_cat,
                 family = poisson(), data = df)

# Summary of the GAM
summary(model_gam)
```

```

Family: poisson
Link function: log

Formula:
days_since_last_high ~ s(DGS1MO) + s(DGS2) + s(DGS10) + s(BAMLC0A1CAAA) +
    s(VIXCLS) + s(DEXJPUS) + s(DTWEXBGS) + day_of_week_cat

Parametric coefficients:
                 Estimate Std. Error z value Pr(>|z|)
(Intercept)      3.854869   0.004592 839.554   <2e-16 ***
day_of_week_cat -0.006680   0.001353  -4.935    8e-07 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Approximate significance of smooth terms:
                  edf Ref.df Chi.sq p-value
s(DGS1MO)       8.996      9   9930  <2e-16 ***
s(DGS2)         8.985      9   5784  <2e-16 ***
s(DGS10)        8.998      9  20293  <2e-16 ***
```

```
s(BAMLCOA1CAAA) 8.985      9   5822  <2e-16 ***
s(VIXCLS)       8.985      9   5555  <2e-16 ***
s(DEXJPUS)      8.995      9  26476  <2e-16 ***
s(DTWEXBGS)     8.993      9  18599  <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

R-sq.(adj) =  0.837   Deviance explained = 83.5%
UBRE = 27.132  Scale est. = 1         n = 2724
```

11.4.2 Interpretation

The Generalized Additive Model (GAM) was fitted using a Poisson distribution to model the count variable `days_since_last_high`. Unlike previous models, GAM allows each predictor to influence the outcome **non-linearly** using smooth spline functions.

11.4.2.1 Key findings:

- All smooth terms are **highly significant (p < 2e-16)**, suggesting that the relationship between the predictors and the outcome is **non-linear**.
- The **estimated degrees of freedom (edf)** for each smooth term are close to 9, confirming that the model allowed flexible shapes for each curve.
- The only parametric effect (`day_of_week_cat`) is also significant (**p < 0.001**), unlike in previous models where it had no influence. This may suggest **non-linear interactions or adjustment effects** in the GAM context.
- The model explains **~83.5% of the deviance** and has an **adjusted R-squared of 0.837**, which is comparable to or better than the previous models.

This confirms that allowing non-linear terms improves the model's ability to capture complex effects of macroeconomic variables on Meta's stock dynamics.

11.5 Generalized Additive Model (GAM) – Negative Binomial

11.5.1 Implementation

Given the strong overdispersion observed in our count data and the potential for non-linear relationships between predictors and the response, we fitted a **Generalized Additive Model (GAM)** with a **Negative Binomial distribution** using the `mgcv` package.

This model combines the benefits of: - Non-linear smoothing splines (for modeling complex patterns), - And the negative binomial family (for modeling overdispersed count data).

```r
#load mgcv if not already done
library(mgcv)

# Fit GAM with Negative Binomial distribution
model_gam_nb <- gam(days_since_last_high ~
                      s(DGS1MO) + s(DGS2) + s(DGS10) +
                      s(BAMLC0A1CAAA) + s(VIXCLS) +
                      s(DEXJPUS) + s(DTWEXBGS) +
                      day_of_week_cat,
                    family = nb(), data = df)

# Show model summary
summary(model_gam_nb)
```

```
Family: Negative Binomial(1.109) 
Link function: log 

Formula:
days_since_last_high ~ s(DGS1MO) + s(DGS2) + s(DGS10) + s(BAMLC0A1CAAA) + 
    s(VIXCLS) + s(DEXJPUS) + s(DTWEXBGS) + day_of_week_cat

Parametric coefficients:
                 Estimate Std. Error z value Pr(>|z|)    
(Intercept)      3.817451   0.032960 115.820   <2e-16 ***
day_of_week_cat -0.001541   0.013406  -0.115    0.908    
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Approximate significance of smooth terms:
                  edf Ref.df Chi.sq p-value    
s(DGS1MO)       8.234  8.830 197.63 < 2e-16 ***
s(DGS2)         8.397  8.900  98.47 < 2e-16 ***
s(DGS10)        8.773  8.980 423.22 < 2e-16 ***
s(BAMLC0A1CAAA) 7.845  8.616 139.61 < 2e-16 ***
s(VIXCLS)       6.493  7.667  27.06 0.00237 ** 
s(DEXJPUS)      8.777  8.983 406.35 < 2e-16 ***
s(DTWEXBGS)     8.789  8.984 381.29 < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

R-sq.(adj) =  0.557   Deviance explained = 60.7%
-REML =  13319  Scale est. = 1         n = 2724
```

11.5.2 Interpretation

The Generalized Additive Model (GAM) with a **Negative Binomial distribution** was applied to account for both **non-linear relationships** and **overdispersion** in the count outcome `days_since_last_high`.

11.5.2.1 Key findings:

- Most smooth terms are **highly significant** (*p < 0.001*), showing strong evidence of **non-linear effects** across all major macroeconomic predictors.
- The degrees of freedom (`edf`) are slightly lower than in the previous GAM Poisson model, indicating that the model fit is more conservative (less wiggly), but still captures complexity.
- The parametric effect `day_of_week_cat` is **not significant** (*p = 0.91*), aligning with previous models and confirming that day-of-week patterns do not meaningfully impact the outcome.

11.5.2.2 Model fit:

- **Deviance explained**: 60.7%

- **Adjusted R²**: 0.557

- **Smoother terms like `DGS10`, `DEXJPUS`, and `DTWEXBGS`** continue to have a large influence on the response.
- While this is slightly lower than the GAM with Poisson (which had ~83% deviance explained), this model offers a **better trade-off between fit and reliability**, especially under high dispersion.

Overall, the GAM NB model provides a robust and interpretable framework for capturing **nonlinear dynamics** in overdispersed count data, while correcting for the limitations of earlier Poisson-based models.