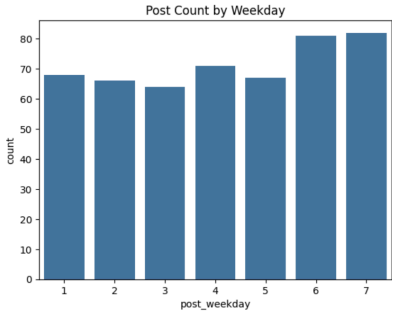
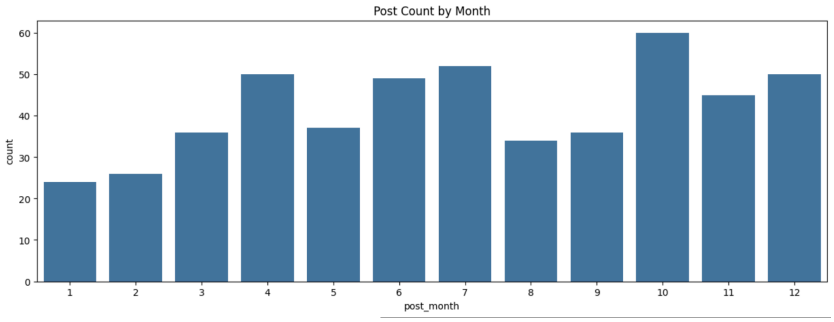
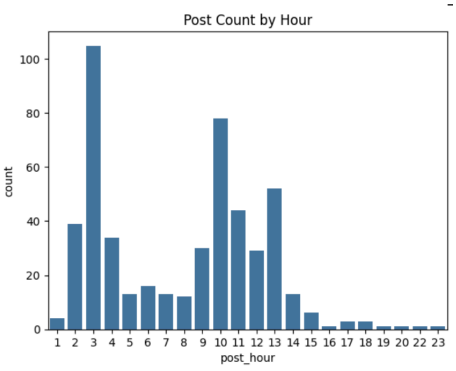
CSci 4521 Homework 3 Writeup
Data Visualization and Analysis



The company's Facebook posting strategy reveals patterns across weekdays, months, and hours. The data for weekdays, months, and hours is not approximately normally distributed. A normal distribution would be a bell-shaped curve with a single peak in the center which is not the case for any given data. The distribution for all the histograms can be described as multimodal with multiple peaks. For the weekdays, there's a noticeable upward trend on the weekends (6 and 7). This means that the most common days are Saturday and Sunday. This distribution might be due to audience availability, content type, strategy, and trends. People are usually more active on social media during the weekends when they have more free time, and content posted on the weekends might be more engaging and relevant to the audience, leading to higher posting frequency. Companies might have a deliberate strategy to focus on weekend posting to maximize reach and engagement. The company's current strategy appears to be heavily favor weekend posting. This suggests a focus on reaching users during their leisure time.

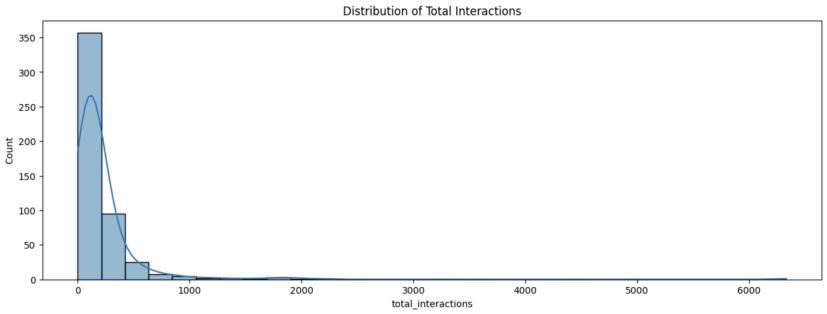


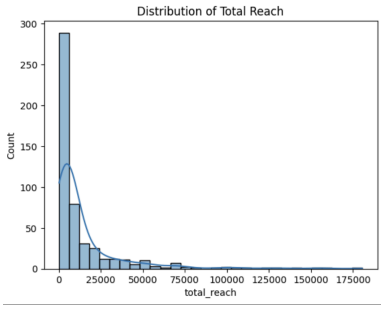
For the month, there are peaks and dips, indicating fluctuation in posting activity throughout the year. The months with the highest (most common) post counts are October (10) and July (7). Some factors that could contribute to this distribution include seasons, marketing campaigns, content planning, and resources. Some companies experience seasonal fluctuations, for example, there may be higher posting activity during the holiday seasons (like October). Specific marketing campaigns might be launched during certain months, leading to increased posting. The company might also have a content calendar that focuses on specific themes or events during certain months. Resource availability could fluctuate throughout the year, impacting posting frequency. The company's current strategy appears to have many fluctuations in posting activity throughout the year. The significant peak in October suggests a focus on content during that month.



For the hours, there are distinct peaks at specific hours, indicating a non-uniform pattern. The common posting times are 3 AM and 10 AM. This distribution might be due to audience time zones, scheduling tools, content types, and trends. If the audience is spread across multiple time zones, the peaks might reflect optimal posting times for different regions. The company might be using scheduling tools that automatically publish posts

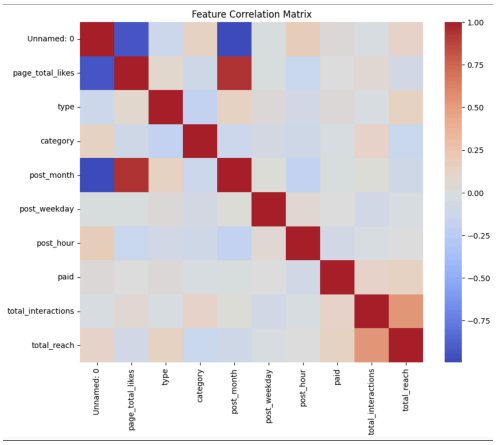
at specific times. Certain types of content might be more effective at specific times. Specific companies might have established best practices for posting times. The company's current strategy appears to have two distinct peak posting times which are 3 AM and 10 AM. This is a very early morning peak which likely means companies are targeting audiences in different time zones. This is a more typical morning peak, likely aligning with the start of the workday for many people.





The distribution of total interaction in the histogram is right-skewed. The majority of the data is concentrated on the left side of the distribution, with the tail extending to the right. Most posts have relatively few interactions, while a smaller number of posts have a high number of interactions. The distribution of total reach in the histogram is strongly right-skewed. This indicates that most posts have a low reach, while only a few have a high reach. Some posts are significantly more successful in reaching a wider audience than the majority of posts.

The correlation matrix reveals several key relationships between Facebook post features. Notably, page total likes are strongly positively correlated with both total interactions and total reach, suggesting that larger pages generally achieve higher engagement. Conversely, the post type, category, and month show negative correlations with interactions and reach, implying certain content types or periods may underperform. As expected, total interactions and total reach are highly positively correlated, indicating that posts with more engagement tend to reach a wider audience. Paid promotion also positively correlates with interactions and reach. Interestingly, the unnamed index shows a negative correlation with page total likes, potentially reflecting a trend of decreasing page growth over time. These correlations offer insights into factors driving Facebook post performance and can inform strategies for optimizing content and engagement.



Parametric Modeling

To analyze the performance of the 3 predictive models for post-interactions, we will discuss the inputs that were selected for each model, describe the normalization techniques used, present the necessary equations, and analyze the results of each of the model's performance.

Choosing the inputs for each model depends on the heatmap we discussed in Data Visualization and Analysis. Based on the heatmap, the features that showed the most positive correlation (this also includes negative features that are lower in magnitude in comparison to other correlations) with total_interactions were chosen. Those features are page_total_likes, category, post_month, and paid. The single feature model uses page total likes as the input feature, normalized for comparison (page_total_likes_normalized). The multiple features linear model incorporates multiple normalized input features to capture more complex relationships between features and post-interactions. The features used are category, post_month, and paid which then are normalized to category_normalized, post_month_normalized, and paid_normalized. The nonlinear features model captures nonlinear relationships by incorporating both polynomial and interaction terms in the input features. It uses the following nonlinear features: category_sq_normalized, post_hour_cu_normalized, and category_paid_normalized (interaction term between category and paid). The function gradDec uses a learning rate (lr) of 0.001, which was chosen by testing different values and seeing which one gave the best results in terms of loss.

All input features were normalized using the formula: $x' = \frac{x - x.mean}{x.std}$. This ensures that all input features are on a comparable scale preventing any feature from disproportionately influencing the model due to scale. The target variable (output feature), total_interactions, was also normalized to ensure comparability across different models. This way we can ensure that both the feature and target variables are on the same scale which can improve the training of the models. The formula used for y (target variable) is: $y' = \frac{y - y.mean}{y.std}$.

To predict the unnormalized post-interactions (y_{raw}) from the unnormalized feature inputs (x_{raw}), we need to undo the normalization transformation applied (see paragraph above). The model is trained on the normalized variables: $y' = p_0 x_0' + p_1 x_1' + p_2 x_2' + p_3$. To get to the unnormalized prediction, first, we need to express y_{raw} in terms of y'. This will be given by: $y_{raw} = y' * y.std + y.mean$. Now, we can substitute y' in the equation: $y_{raw} = (p_0 x_0' + p_1 x_1' + p_2 x_2' + p_3) * y.std + y.mean$. Second, we need to express x' in terms of x_{raw} which is given by $x' = \frac{x_{raw} - x.mean}{x.std}$. Substituting this into the equation, we get:

$$y_{raw} = \left(p_0 \frac{x_{0,raw} - x_0.mean}{x_0.std} + p_1 \frac{x_{1,raw} - x_1.mean}{x_1.std} + p_2 \frac{x_{2,raw} - x_2.mean}{x_2.std} + p_3\right) * y.std + y.mean.$$

This equation allows us to predict unnormalized post interactions from unnormalized inputs by reversing the normalization process. To present a numerical analysis, we will use the features (x) of the multiple features linear model and use total_interactions as our y. After printing out the means and standard deviations of the x's (features), we get: category ($x_0.mean$ = 1.88 and $x_0.std$=0.85), post_month ($x_1.mean$ = 7.05 and $x_1.std$=3.30), and paid ($x_2.mean$ = 0.28 and

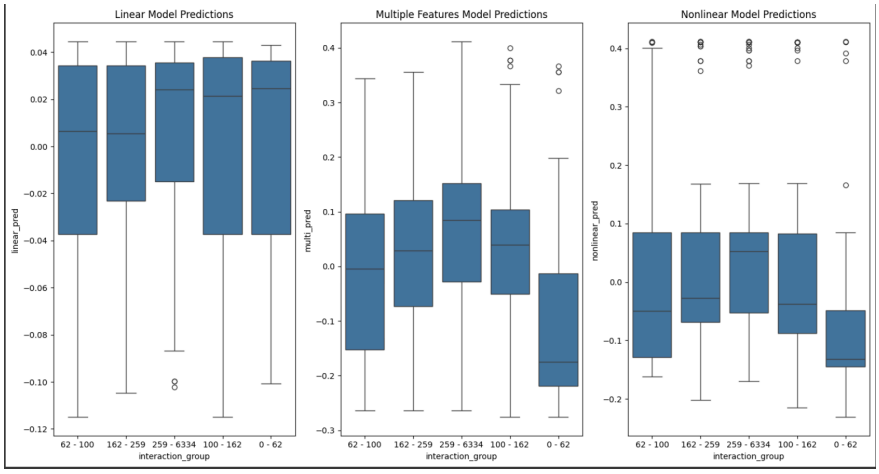
$x_2.std$=0.45). For y, we get: total_interactions (y.mean = 212.31 and y.std = 380.59). We can also get the computed parameters: $p_0\,(0.1344),\ p_1(0.0366),\ p_2(0.1113),\ and\ p_3(0.0002)$ of the multiple-feature linear model. Therefore, the prediction equation for unnormalized post-interactions looks like the following:

$$y_{raw} = (0.1344\ \frac{x_{0,raw} - 1.88}{0.85d} + 0.0366\ \frac{x_{1,raw} - 7.05}{3.30} + 0.1113\ \frac{x_{2,raw} - 0.28}{0.45} + 0.0002)\ *\ 380.59 + 212.31.$$

We evaluate the performance of each model based on two metrics, mean squared error (MSE) and mean absolute error (MAE). For the single-feature linear model, the MSE is 0.996 and the MAE is 0.453. The multiple features model has an MSE of 0.968 and an MAE of 0.438. And finally, the nonlinear model has an MSE of 0.974 and an MAE of 0.430. These results show that the multiple features model performs the best in terms of MSE and MAE. While the nonlinear model has a slightly higher MSE, it still offers competitive performance, and the single feature model performs the "worst" overall.

```
Single Feature Model - MSE: 0.9960406422615051, MAE: 0.4350050985813141
Multiple Features Model - MSE: 0.9682572484016418, MAE: 0.4279155731201172
Nonlinear Features Model - MSE: 0.9737780690193176, MAE: 0.430330753326416
```

To assess how well each model predicts the post metric across different interaction groups, we segment the posts into five groups based on the post metric (divisions of total interactions into five). We then use boxplots to visualize the performance of the models. The linear model shows consistent predictions across the interaction groups, but with a narrower range of predictions compared to the other models. The multiple features model exhibits a wider range of predictions between interaction groups. It shows better flexibility in handling diverse post metrics. The nonlinear model shows the broadest range of predictions, indicating its sensitivity to extreme post metrics. It also shows a similar range of performance across interaction groups, with some outliers.



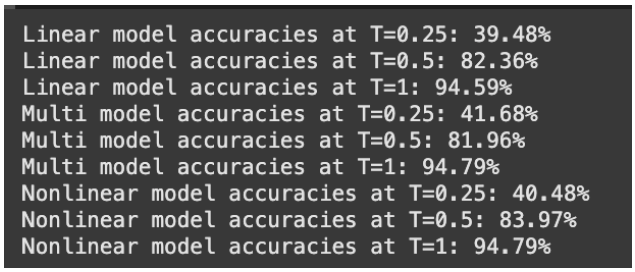
Based on the MSE and MAE results, the multiple features model outperforms both the single-feature linear model and the nonlinear features model. The nonlinear model, while more complex, does not offer substantial improvements over the multiple features model in terms of error metrics. The visualizations further support these findings, showing that the multiple

features model provides the most consistent and reliable predictions across different post-interaction groups.

Testing Accuracy

To evaluate the accuracy of the models, we computed a threshold accuracy metric where a prediction is considered correct if it is within T standard deviations of the actual post-success metric. The evaluation was performed at three different threshold levels: *T=0.25*, *T=0.5*, and *T=1.0*. The results show that at the highest threshold (T=1.0), all three models performed similarly, correctly predicting over 94% of cases within one standard deviation of the actual metric. At T=0.25, the multiple features model achieved the highest accuracy (41.68%), followed by the nonlinear model (40.48%) and the linear model (39.48%). At T=0.5, the nonlinear model outperformed the others with an accuracy of 83.97%, slightly higher than the linear model (82.36%) and the multiple features model (81.96%).

While the differences suggest that adding multiple features or incorporating nonlinear relationships improves predictive performance, the overall performance differences are relatively small. The nonlinear model consistently performs slightly better at the middle threshold (T=0.5), while the multiple features model excels in the smaller threshold (T=0.25). However, given that all models achieved similar performance at T=1.0 and relatively small differences at smaller thresholds, we conclude that no single model dramatically outperforms the others. Nonetheless, the multiple features and nonlinear models appear to offer slight improvements over the linear model, suggesting that incorporating additional features and nonlinear relationships may enhance the model's ability to capture variations in post-success.



Cross Validation

To assess the robustness of our models, we recomputed the thresholded accuracy using cross-validation, reporting the mean accuracy and standard deviation across multiple training/testing sets. At the smallest threshold (T=0.25), the linear and nonlinear models performed equally well (34.95%), while the multiple features model slightly underperformed (34.55%). However, the standard deviations were quite large (~ 20%), indicating significant variability in model performance across different training splits. At T=0.5, the nonlinear model achieved the highest accuracy (67.68%), outperforming the multiple features model (63.23%) and the linear model (59.60%). The differences at this threshold were more noticeable, with the nonlinear model leading by over 4 percentage points compared to the multiple features model and by more than 8 points compared to the linear model. Finally, at T=1.0, all models performed similarly, with the nonlinear model achieving the highest mean accuracy (91.31%), slightly

above the linear model (90.71%), and the multiple features model (90.30%). However, the standard deviations at this level were significantly lower (~ 4%), indicating more stable performance.

The nonlinear model consistently outperforms the other two, particularly at the middle threshold (T=0.5), the standard deviations suggest considerable overlap between the models' performance. The differences between models are comparable to or smaller than the within-model variability (standard deviations), meaning that while the nonlinear model appears slightly superior, the improvements are not strongly significant given the variability in accuracy across different data splits. This suggests that while incorporating multiple features and nonlinear relationships can enhance predictive performance, the gains are modest and sensitive to the specific training set used.

```
Linear model at T=0.25: Mean Accuracy = 34.95%, Std Dev = 21.99%
Linear model at T=0.5: Mean Accuracy = 59.60%, Std Dev = 24.33%
Linear model at T=1: Mean Accuracy = 90.71%, Std Dev = 4.49%
Multi model at T=0.25: Mean Accuracy = 34.55%, Std Dev = 20.14%
Multi model at T=0.5: Mean Accuracy = 63.23%, Std Dev = 18.39%
Multi model at T=1: Mean Accuracy = 90.30%, Std Dev = 4.50%
Nonlinear model at T=0.25: Mean Accuracy = 34.95%, Std Dev = 20.43%
Nonlinear model at T=0.5: Mean Accuracy = 67.68%, Std Dev = 16.16%
Nonlinear model at T=1: Mean Accuracy = 91.31%, Std Dev = 3.92%
```

**Note: All metrics and numbers are based on one run of the program and may change when the program is run again.*