This paper discusses the models used to predict tips. It will compare different models and explain the performance of the models. The goal of the project is to predict the tip given the features: total bill, party size, time of day, and day. Two modeling experiments were conducted: one using the original model and the other using Principal Component Analysis (PCA). The models built were Linear Regression and Neural Network models; their performance was analyzed and compared to see which model was the most accurate and why.

Data Preparation and Exploratory Data Analysis

The project uses the Seaborn Tips dataset, which contains data about a restaurant with the features: total_bill, smoker, sex, tip, day, time, and size. But before I could start the modeling experiment or perform my analysis, I had to clean and preprocess the data.

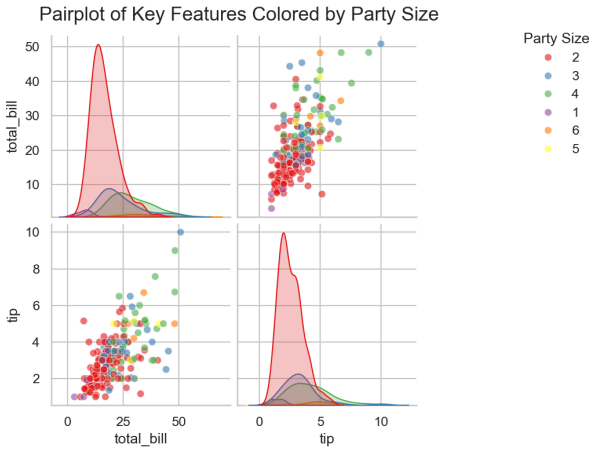
I had to remove of any NaNs or zero total bills, along with the removal of outliers to prevent bias and incorrect predictions.

- Encoding the categorical data:
    - Day: Thur=0, Fri=1, Sat=2, and Sun=3
    - Time: Lunch=0, Dinner=1

Then, feature selection based on our knowledge of the dataset and correlation analysis, which was: time, day, party size, and total bill

The correlation matrix shown reveals important insights between the different features. First, there is a strong correlation between total bill and tip amount, meaning that there is a strong linear relationship

between them; they had a score of 0.68. There is a moderate correlation between party size and tip amount, generating a score of 0.49. There are weak correlations between day/time codes and tip amount. But there is no significant multicollinearity between the features.



The pairplot visualized the relationships in the correlation matrix and highlighted the nonlinear patterns, suggesting that Neural networks would capture the complex interactions better compared to Linear Regression, which only models straight line relationships.

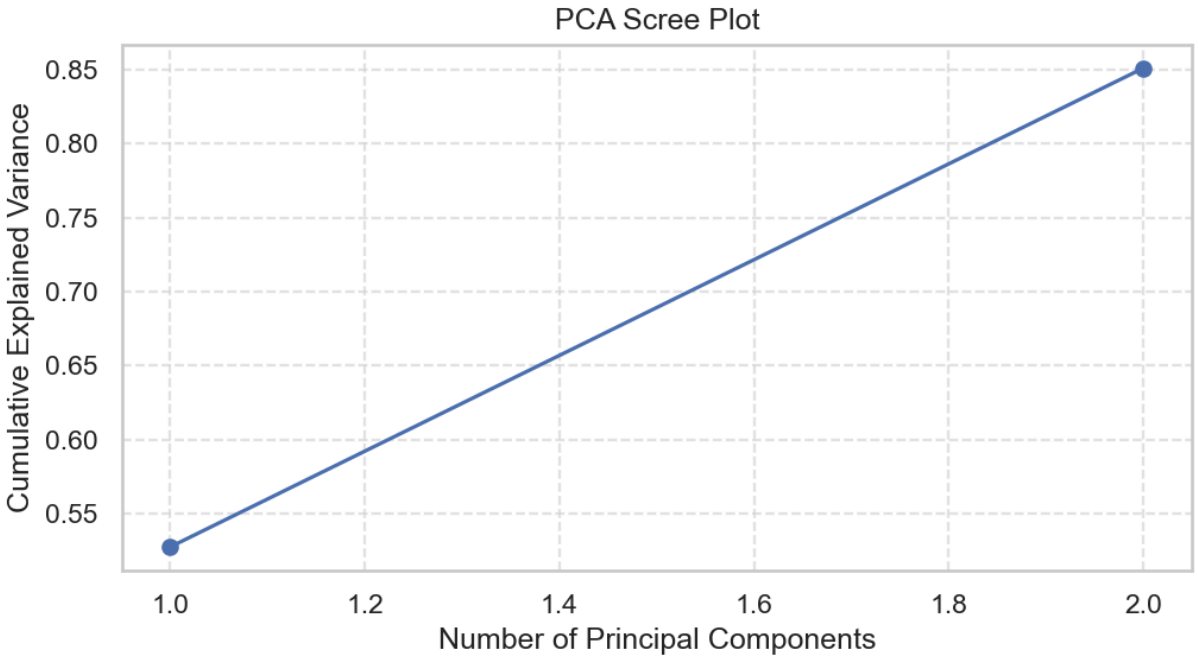
Modeling Experiments

The two experiments were designed to compare the model's performance.

Experiment A used the original features:

- Total bill
- Party size
- Day code
- Time code

Preservation of the original features allowed for more interpretability, and with standardization using StandardScaler, it ensured that no feature dominated the model due to the value ranges.



Experiment B used PCA to reduce the number of features while also keeping 85% of the original data's variance. This removed redundant information, which could help boost the model's performance. The scree plot shows that you can summarize the key data using only two features.

Model Development

Both models were trained with a 60/20/20 split; 60 for training, 20 for validation, and 20 for testing.

Linear Regression assumes that there is a linear relationship between the features and the target. It is easier to interpret while training and predicting quickly using fewer resources.

Neural Network Regression had a feed-forward model where data flows from input to output through two hidden layers. This model helps learn complex nonlinear patterns.

When evaluating the metrics, I used the following tools to help: MSE, which weights errors more heavily, MAE to interpret the results in dollars, MAPE to measure the error, and R^2 to measure explained variance.

Detailed Analysis and Recommendations

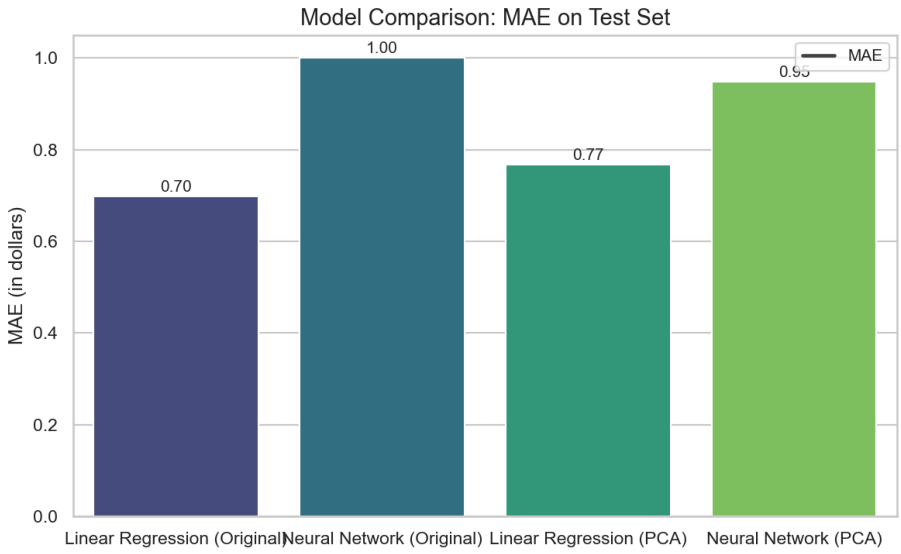
Model	MAE (in $)	MAPE	R^2
L.R. (original)	0.698 (0.7)	26.60	0.285
N.N. (original)	1.000	42.20	-0.384
L.R. (PCA)	0.767 (0.77)	29.58	0.111
N.N. (PCA)	0.947 (0.95)	39.68	-0.178

The Linear Regression model with the original features demonstrated an extremely strong performance:

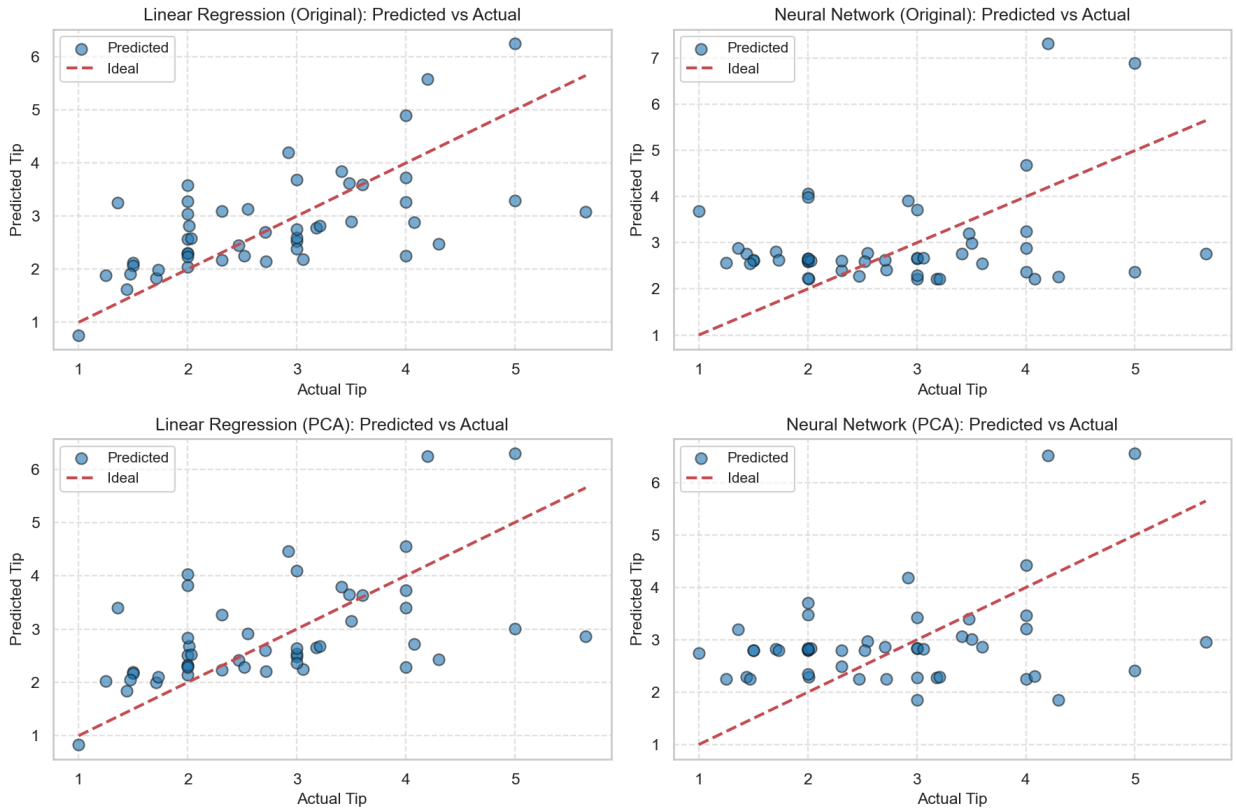
- MAE = $0.698
- R^2 = 0.285
- MAPE: 26.60%

The Linear Regression model performed the best due to the relationships in the original features being linear and the dataset having low noise. PCA hurt interpretability by removing important

context since the original features had low redundancy. The R^2 in both Neural Network models is negative due to overfitting in the training data, and the PCA makes the data more abstract.



The model comparison chart illustrates that the Linear Regression model with the original features greatly outperformed the other models. The PCA transformation was not helpful and degraded the performance of the models. The Neural Network underperformed compared to Linear Regression.



The predicted vs. actual plots revealed that Linear Regression was able to fit the data better, with all of the points being close to the ideal line. Neural Networks and PCA-transformed models were more scattered and less accurate.

Summary

Feature engineering would improve results, adding more time-related variables (peak hours, holidays) to better capture how time affects tips. Using a Random Forest model for neural networking would capture the nonlinearities better while avoiding instability. More data always helps; it allows complex models to learn new data faster. Additional features like server experience, location, and average server rating would improve predictions because those directly affect tipping behavior.

The project highlights when to use simple and complex models. Simple models perform better with limited data and features, but adding more data and richer features would improve the performance of complex models. If you incorporate the improvements, future models would have higher accuracy and become more valuable in the real world.