AMERICAN UNIVERSITY OF BEIRUT - Mediterraneo

Optimizing Lead Acquisition Costs for Nuun Digital: A Predictive Analysis Using Machine Learning

by
Aya El Saoudi

Advisor(s)
Dr. Christos Nicolaides

A Capstone Project
submitted in partial fulfillment of the requirements
for the degree of Master's in Business Analytics
to the Faculty of Business
at the American University of Beirut - Mediterraneo

Paphos, Cyprus
August 2024

AMERICAN UNIVERSITY OF BEIRUT
Mediterraneo

Abstract

Aya El Saoudi for Master's in Business Analytics (MSBA)

Title: Optimizing Lead Acquisition Costs for Nuun Digital: A Predictive Analysis Using Machine Learning

This This project explores the application of machine learning techniques to optimize Cost Per Lead (CPL) for Nuun Digital, focusing on predictive modeling and feature impact analysis to enhance marketing efficiency. The project involved evaluating various models, including linear regression, decision trees, and neural networks, to identify the most effective approach for predicting CPL. Among these, the neural network emerged as the most accurate, demonstrating a high R-squared value and low error metrics, reflecting its ability to capture complex, non-linear relationships within the data.

Also, feature impact analysis using SHAP values revealed that key features, such as total number of leads and total marketing spend, have a substantial effect on CPL. These insights emphasize the importance of focusing on these variables to achieve effective CPL optimization. The project also highlighted the need for ongoing data enrichment to improve model accuracy and relevance.

On the other hand, optimization tasks involved analyzing the impact of changes in features on CPL and determining the adjustments needed to achieve target reductions. This approach allowed for actionable recommendations to refine marketing strategies and optimize lead acquisition processes effectively. By identifying the most impactful features and understanding their sensitivity, the project provides strategic guidance for resource allocation and decision-making.

Table of Contents

Table of Figures

1. Introduction

In today's competitive marketing landscape, understanding the customer journey is paramount for businesses aiming to optimize their acquisition strategies. The marketing funnel, or acquisition funnel, serves as a critical framework that delineates the stages a potential customer traverses, from the moment they first encounter an advertisement to their eventual conversion into a loyal customer. This process can be broken down into four key stages: Impression, where the user sees the ad; Click, where interest is demonstrated by clicking the ad; Lead, where deeper engagement occurs through actions like form submissions; and finally, Acquisition, where the user makes a purchase or subscription [1]. Each transition between these stages represents a conversion, and tracking conversion rates and associated costs is essential for refining marketing efforts.

To visually represent this process, Figure 1 illustrates the marketing funnel and its stages. This image highlights the flow from initial exposure to conversion, providing a clear depiction of how each stage builds upon the previous one.

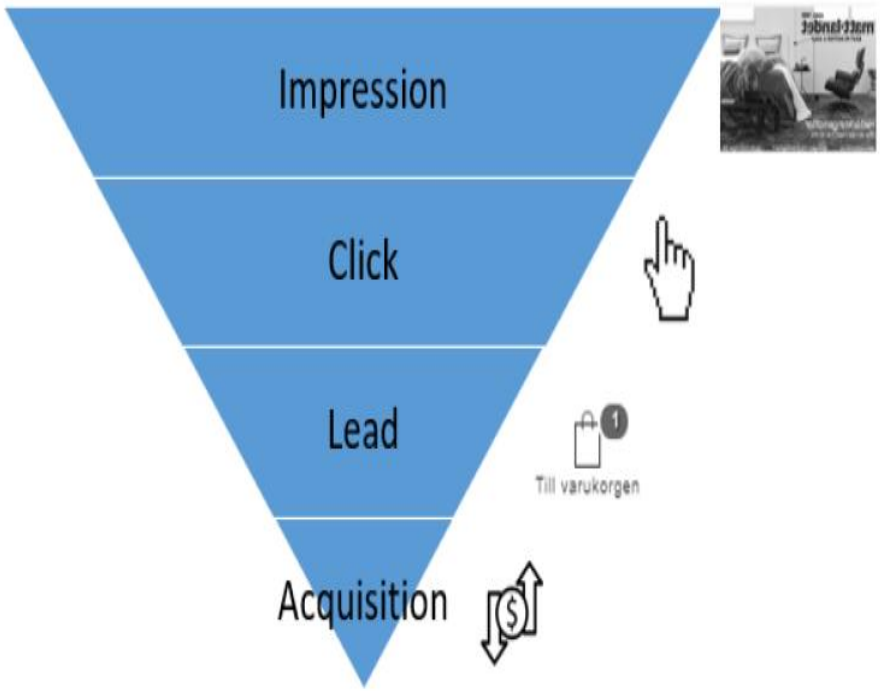

Figure 1- The Marketing Funnel - Stages from Initial Exposure to Conversion

In the marketing world, the significance of the acquisition funnel lies in its ability to provide actionable insights into consumer behavior, allowing businesses to identify bottlenecks, allocate

resources effectively, and enhance overall campaign performance. By analyzing each stage of the funnel, marketers can develop targeted strategies that not only increase conversion rates but also optimize customer acquisition costs, ultimately driving sustainable growth and improving return on investment. As companies increasingly rely on data-driven decision-making, mastering the marketing funnel has never been more critical for achieving competitive advantage and long-term success.

In this capstone project, we will focus on the Lead stage and aim to predict its acquisition cost (Cost per lead) and optimize it accordingly for Nuun Digital. Nuun Digital is a powerhouse that seamlessly integrates strategy consultancy, digital marketing, digital agency services, and software development and design into one cohesive entity. The Lead stage is particularly crucial as it represents a deeper level of engagement where potential customers have shown genuine interest by taking specific actions, such as filling out a form or signing up for more information. This stage is a pivotal point where businesses have the opportunity to convert interest into tangible results. By accurately predicting and optimizing acquisition costs at this stage, we can enhance the efficiency of marketing efforts and maximize the return on investment, ultimately leading to better resource allocation and increased customer acquisition success.

2. Background and Related Work

Accurate prediction and optimization of Lead Acquisition Cost (cost per lead) are crucial for refining marketing strategies and maximizing return on investment. Advanced machine learning and data science techniques play a pivotal role in achieving these objectives. Several key studies provide valuable methodologies and insights that are instrumental in predicting lead acquisition costs for Nuun Digital.

The paper "Prediction of CPC Using Neural Networks for Minimization of Cost" by Chaudhari, Dingankar, and Chaudhari (2013) demonstrates the efficacy of neural networks in optimizing the prediction of Cost Per Click (CPC) in online advertising [2]. The study highlights how neural networks, through their advanced pattern recognition capabilities, can evaluate keyword efficiency by analyzing stored numeric values. This approach effectively reveals the intricate relationships within the data, leading to more accurate CPC predictions. The findings from this paper underscore the strength of neural networks in enhancing cost predictions and provide valuable techniques that can be applied to improve the accuracy of lead acquisition cost estimations.

In the study "Cost Prediction in Acquiring Customers Using Machine Learning" by Darapaneni et al. (2023), the focus is on predicting Customers Acquiring Costs (CAC) through various machine learning algorithms, including Decision Tree (DT), Random Forest (RF), and Bagging Forest (BF) [3]. This research addresses the complexity of predicting CAC due to numerous independent variables and high cardinality categorical data. The study highlights that Decision Tree demonstrated the highest accuracy among the algorithms tested. This research is highly relevant to the capstone project as it provides practical insights into the effectiveness of different machine learning models for cost prediction. Implementing the Decision Tree algorithm or other high-performing models identified in this study can significantly enhance the precision of lead acquisition cost predictions. The ability to compare and select the most accurate algorithms will guide the development of robust predictive models and optimize marketing strategies for Nuun Digital.

The paper "Machine Learning Based Cost Prediction for Acquiring New Customers" by Prasad et al. (2024) underscores the importance of predicting Customers Acquiring Costs (CAC) as a critical metric for evaluating marketing performance [4]. The study explores how data science and machine learning can be leveraged to analyze marketing strategies and forecast CAC based on various features. Notably, the paper identifies neural networks as the most effective model for CAC prediction, highlighting their superior performance compared to other machine learning techniques. This study

aligns closely with the objectives of the capstone project, providing a framework for developing effective predictive models. By adopting the neural network model, the project can leverage advanced data analytics to enhance the accuracy of lead acquisition cost predictions.

Incorporating insights and methodologies from these studies will significantly benefit the capstone project. The integration of neural network models, comparisons of machine learning techniques, and advanced data analytics will enhance the precision and effectiveness of lead acquisition cost predictions. Each study provides significant insights and contributions, ranging from advanced neural network techniques to practical machine learning algorithms. These insights will be instrumental in developing a robust prediction model for lead acquisition costs and optimizing strategies for Nuun Digital.

3. Methodology

This section details the comprehensive approach undertaken to address the regression problem at hand. The methodology encompasses the entire workflow, from dataset development to performance evaluation, ensuring each stage is meticulously optimized to enhance the effectiveness of our final selected model.

a. Approach

The approach to developing a lead acquisition cost prediction model follows a systematic process designed to ensure accuracy and effectiveness. The process begins with data collection, where relevant and comprehensive data is gathered to support the modeling effort. This data undergoes data normalization, which involves scaling and encoding features to prepare them for analysis. Subsequently, the data is split into training, validation, and test sets. The model is initially trained on the training set and optimized using the validation set. The model's performance is then evaluated with metrics such as Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE) on the test set. If necessary, the model is subject to fine-tuning to improve predictive accuracy. The best-performing model is selected and saved for deployment. This saved model is employed for optimization tasks, supporting strategic resource allocation and decision-making. Additionally, the processed data and model outputs are integrated into a dashboard, offering a user-friendly interface for real-time insights and visualization to aid in informed decision-making for lead acquisition strategies.

The figure illustrates the process for building and deploying a lead acquisition cost prediction model, from data collection to cost per lead optimization and dashboard integration.

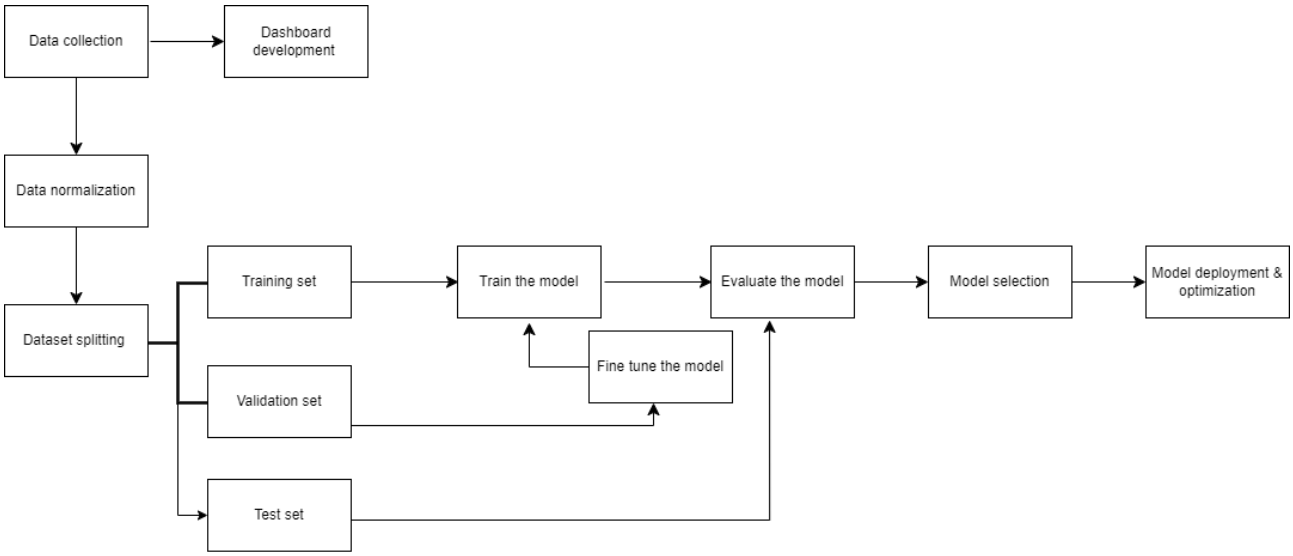

Figure 2- Workflow for Lead Acquisition Cost Prediction Model Development

b. Data Collection

The dataset utilized for this project was sourced from Nuun Digital and comprises 14,737 rows of data, capturing a wide array of metrics pertinent to digital marketing campaigns. The data was compiled and stored in an Excel sheet, which facilitated efficient management and analysis. This comprehensive dataset includes various attributes categorized into several key domains:

1. Campaign and Platform Information:
    - Social Media/Platform: Indicates the platform where the ad was displayed (e.g., Facebook, Instagram).
    - Industry: Represents the sector or industry relevant to the ad campaign (e.g., Healthcare, Finance).
    - Location, City, State/Province: Geographic details specifying where the ad was targeted.
2. Campaign Performance Metrics:
    - Impressions: The total number of times the ad was displayed to users.
    - Clicks: The number of times users clicked on the ad.
    - Ad Cost: The amount spent on the ad campaign.
    - CTR (%): Click-Through Rate, calculated as (Clicks/Impressions) * 100.
    - CPC ($): Cost Per Click, calculated as Ad Cost/Clicks.
3. Engagement and Action Metrics:

- Certain Action: Indicates any specific actions taken by users after interacting with the ad (e.g., signing up, purchasing).
- Sales: The total number of sales generated from the ad.
- Revenue: The total revenue earned from the ad.
- Engagements: The total number of user interactions with the ad (likes, shares, comments).
- Engagement Type: Type of engagement (e.g., Like, Share).

4. Financial Metrics:
    - Total Marketing Spend: Total expenditure on marketing efforts.
    - Total Number of Leads Generated: The total number of potential leads acquired.
    - CPA ($): Cost Per Acquisition, calculated as Ad Cost/Total Number of Leads Generated.
    - CPS ($): Cost Per Sale, calculated as Ad Cost/Sales.
    - CPL ($): Cost Per Lead, calculated as Ad Cost/Total Number of Leads Generated.
    - A/S: A financial metric related to the ad's performance (specific calculation details may vary).
    - ROAS: Return On Ad Spend, calculated as Revenue/Ad Cost.
    - ROI (%): Return on Investment, calculated as (Revenue - Ad Cost)/Ad Cost * 100.
5. Additional Metrics:
    - Content/Ad Format: The format of the ad content (e.g., image, video).
    - Device Type: Type of device used by the audience (e.g., Mobile, Desktop).
    - Time of Day: The time when the ad was displayed.
    - Temperature: Weather condition at the time of the ad campaign (this might be used for contextual analysis).

Some of these metrics have been utilized as Key Performance Indicators (KPIs) in the dashboard, providing real-time insights into campaign performance. KPIs such as CTR, CPC, CPA, and ROAS were selected for their relevance in assessing the effectiveness of marketing strategies and optimizing lead acquisition efforts [5]. This approach ensures that the dashboard effectively supports decision-making by highlighting critical performance metrics and trends. By analyzing these categories, the project aims to refine lead acquisition cost predictions and optimize marketing strategies for Nuun Digital.

c. Data Preprocessing

Data preprocessing is essential for preparing the dataset from Nuun Digital for machine learning modeling. In this project, two key preprocessing steps were employed: data normalization and data splitting.

Data Normalization

In this project, data normalization was achieved through categorical encoding using the `LabelEncoder` technique. This method was employed to transform categorical variables into numerical format, making them suitable for machine learning algorithms. Specifically, columns such as 'Social Media/Platform', 'Industry', 'Location', and others were encoded to convert their text values into integer labels. This transformation is crucial as many machine learning models require numerical input to perform calculations and learn patterns effectively. By applying `LabelEncoder`, categorical data is systematically converted into a format that preserves the original relationships while enabling the model to process and analyze the data accurately. This normalization step ensures that the model can interpret and utilize all features, leading to more robust and reliable predictions. A screenshot demonstrating this encoding technique is provided below.

```
# Encode categorical columns
categorical_columns = ['Social Media/Platform', 'Industry', 'Location', 'City', 'State/Province', 'Engagement Type', 'Device Type', 'Content/Ad Format']
label_encoders = {}
for col in categorical_columns:
    le = LabelEncoder()
    if col in df.columns:
        df[col] = le.fit_transform(df[col].astype(str))
        label_encoders[col] = le
```

Figure 3- Application of Label Encoding for Normalizing Categorical Data

Data Splitting

The dataset was divided into three subsets: training, validation, and test sets. This split ensures that the model can be trained, validated, and tested effectively. The training set is used to fit the model and learn from the data. The validation set is utilized for tuning model parameters and evaluating performance during training, allowing iterative improvements. The test set, kept separate from training and validation, provides an unbiased assessment of the model's performance on unseen data. Typically, the data is split into approximately 70% for training, 15% for validation, and 15% for testing. This approach helps ensure that the model is robust and generalizes well to new, unseen data.

d. Model Selection

In the pursuit of optimizing lead acquisition cost predictions, the selection of an appropriate model is critical. Based on the literature review, three prominent models have been identified as effective for solving similar problems: Linear Regression, Decision Trees, and Neural Networks. Each of these models brings distinct advantages and capabilities to the table.

Linear Regression

Linear Regression is a fundamental statistical method used to model the relationship between a dependent variable and one or more independent variables. This model assumes a linear relationship and estimates the coefficients that best fit the data. Linear regression is straightforward and easy to implement, making it a good starting point for many predictive tasks. However, its simplicity may limit its effectiveness in capturing complex, non-linear relationships present in the data. Despite this, linear regression can still provide valuable baseline results and insights, especially when the relationships between features and outcomes are approximately linear.

The `LinearRegression` model from `sklearn.linear_model` establishes a linear relationship between the target variable and one or more predictors by fitting a linear equation to the data. It minimizes the residual sum of squares between observed and predicted values, offering clear insights into feature impacts through its coefficients. This model is computationally efficient and highly interpretable, making it suitable for large datasets. Below is a screenshot illustrating the implementation of this model.

```
# Initialize and fit the linear regression model
model_1 = LinearRegression()
model_1.fit(X_train, y_train)
```

Figure 4- Implementation of the Linear Regression model from sklearn.linear_model

Decision Trees

Decision Trees are widely recognized for their clarity and ease of interpretation. They work by dividing the data into subsets based on specific feature values, forming a tree-like structure where each branch represents a decision rule and each leaf denotes a predicted outcome. This approach effectively models hierarchical decision-making processes, providing a straightforward visualization of how decisions are made. Although Decision Trees may be susceptible to overfitting, they are valuable for revealing feature importance and decision criteria, offering valuable insights into the factors driving lead acquisition costs.

The Decision Tree Regressor from sklearn.tree is utilized to model the lead acquisition cost. This model constructs a decision tree by recursively splitting the data based on feature values to create a structure that captures the relationships between features and target outcomes. Each node in the tree represents a decision criterion, and the branches represent possible outcomes of those decisions. The model is trained using the DecisionTreeRegressor class with a specified random seed to ensure reproducibility. The trained model can provide insights into the key decision rules influencing lead costs. Below is a screenshot of the Decision Tree Regressor implementation.

```
# Initialize and train the decision tree regressor
model_2 = DecisionTreeRegressor(random_state=42)
model_2.fit(X_train, y_train)
```

Figure 5- Implementation of the Linear Regression model from sklearn.linear_model

Neural Networks

Neural Networks are a class of machine learning models inspired by the human brain's structure and functioning. They excel in capturing complex, non-linear relationships within data. Neural networks are particularly useful for problems involving large datasets with intricate patterns, making them well-suited for predicting lead acquisition costs. Their ability to learn and adapt through multiple layers of processing allows them to model intricate dependencies and interactions between features. The advanced pattern recognition and error correction capabilities of neural networks can enhance the precision of cost predictions by uncovering hidden relationships in the data.

The neural network model, built using Keras, features a sequential architecture comprising three layers. The first layer is a dense layer with 64 units and ReLU activation, followed by a second dense layer with 32 units and ReLU activation. The final layer outputs a single value. The model is compiled using the Adam optimizer and mean squared error loss function, making it well-suited for regression tasks. The model summary reveals a total of 3,329 parameters, all of which are trainable. Below is a screenshot of the model layers, providing a detailed view of the network's structure and parameter count.

```python
# Build the neural network model
model = keras.Sequential([
    keras.layers.Dense(64, activation='relu', input_shape=(X_train.shape[1],)),
    keras.layers.Dense(32, activation='relu'),
    keras.layers.Dense(1)
])

# Compile the model
model.compile(optimizer='adam', loss='mean_squared_error')
```

Figure 6- Implementation of the Linear Regression model from sklearn.linear_model

Layer (type)	Output Shape	Param #
dense (Dense)	(None, 64)	1,216
dense_1 (Dense)	(None, 32)	2,080
dense_2 (Dense)	(None, 1)	33

Total params: 3,329 (13.00 KB)
Trainable params: 3,329 (13.00 KB)

Figure 7- Implementation of the Linear Regression model from sklearn.linear_model

The choice between the models is determined based on their performance metrics following training. Each model—Linear Regression, Decision Tree Regressor, and Neural Network—offers different strengths and complexities. After training, accuracy metrics such as Mean Absolute Error (MAE), Mean Squared Error (MSE), and R-squared (R^2) are evaluated to assess their predictive performance. The model that demonstrates the highest accuracy and best aligns with the project's objectives is selected. This approach ensures that the final choice is driven by empirical evidence and reflects the model's effectiveness in capturing and predicting lead acquisition costs.

e. Feature Analysis and CPL Optimization

Feature Analysis

For feature analysis, SHAP (SHapley Additive exPlanations) values were utilized to interpret the chosen model's predictions and understand the importance of each feature. The process involved converting the training data back into a DataFrame to retain column names and then initializing the SHAP explainer with the trained model. SHAP values were calculated for the training set, and a summary plot was generated to visualize feature importance.

The SHAP summary plot displays the importance of each feature in the model's predictions. Each point on the plot represents a SHAP value for a particular feature and instance in the training set. Features are plotted on the y-axis, and the SHAP values are on the x-axis. The color of each

point indicates the value of the feature: red dots represent higher feature values, while blue dots represent lower feature values. The spread of points along the x-axis for each feature reflects its impact on the model's output. Features with a wider spread (i.e., more variation in SHAP values) are more influential in the model's predictions. Higher absolute SHAP values (either positive or negative) indicate greater influence. The plot provides an intuitive visualization of which features drive the model's decisions and how their values affect predictions. A screenshot of the SHAP code used for this analysis is provided below.

```
import shap

# Convert X_train back to a DataFrame to retain column names
X_train_df = pd.DataFrame(X_train, columns=features)

# Initialize the SHAP explainer with the trained model
explainer = shap.Explainer(model, X_train_df)

# Calculate SHAP values for the training set
shap_values = explainer(X_train_df)

# Plot the feature importance with real feature names
shap.summary_plot(shap_values, X_train_df, feature_names=X_train_df.columns)
```

Figure 8- SHAP Code for Feature Analysis

CPL Optimization

To analyze the impact of changes in specific features on the Cost Per Lead (CPL), a function was developed to predict CPL after modifying feature values. This function, `predict_cpl_with_feature_change`, operates by copying the original feature set, updating the desired feature to a new value, and then scaling the adjusted feature data. The chosen model is used to predict the CPL for this modified dataset.

For example, setting the 'Clicks' feature to 100 and predicting the resultant CPL allows for the calculation of the difference between this predicted CPL and the average CPL from the dataset. This approach helps in understanding how varying individual feature values affect CPL and can guide optimization strategies. It can guide decision-making and strategy adjustments—for instance, if increasing the number of clicks significantly changes the CPL, it might inform how resources are allocated. A screenshot of the code used for this prediction process is provided below.

```python
from tensorflow.keras.models import Sequential

# Calculate average CPL
average_cpl = y.mean()

def predict_cpl_with_feature_change(feature_name, new_value):
    # Make a copy of the original features DataFrame
    X_copy = X.copy()

    # Set the feature to the new value
    X_copy[feature_name] = new_value

    # Apply scaling to the modified feature DataFrame
    X_scaled = scaler.transform(X_copy)

    # Predict CPL using the scaled feature DataFrame
    predicted_cpl = model.predict(X_scaled)

    return predicted_cpl

# Example usage
feature_to_change = 'Clicks'  # The feature you want to change
new_feature_value = 100       # New value for the feature

# Predict CPL with the new feature value
predicted_cpl = predict_cpl_with_feature_change(feature_to_change, new_feature_value)

# Calculate the average CPL
average_cpl = y.mean()

# Calculate the difference
difference = predicted_cpl.mean() - average_cpl

print(f'Predicted CPL with {feature_to_change} set to {new_feature_value}: {predicted_cpl.mean()}')
print(f'Average CPL: {average_cpl}')
print(f'Difference between new CPL and average CPL: {difference}')
```

Figure 9- The code used for predicting CPL with changes in feature values and analyzing the impact on CPL

To optimize the Cost Per Lead (CPL) and enhance marketing efficiency, a detailed analysis was performed to determine the impact of changes in individual features on the CPL. The process begins by defining a target CPL reduction, which is calculated by subtracting the desired reduction from the average CPL. The code then implements a function, `calculate_feature_changes`, which systematically adjusts each continuous feature in the dataset and evaluates the resulting changes in CPL. Continuous features were chosen for this analysis because they are either directly controllable or can be adjusted to a certain extent, making them suitable for this type of optimization. By incrementally modifying feature values and observing their effect on CPL, the function calculates how much each feature needs to be adjusted to achieve the target reduction. If a feature's change does not result in a significant CPL variation, it is noted as ` No significant change detected`. The output of this analysis provides valuable insights into which features most influence CPL, guiding strategic adjustments to optimize marketing efforts.

This approach allows you to assess how changing individual features affects the CPL, providing insights into which features are most influential. It also helps identify how much each feature needs to be adjusted to achieve a target reduction in CPL, guiding decision-making for marketing strategies and budget allocation. Furthermore, by using the model's predictions to guide feature adjustments, the recommendations are based on actual data rather than assumptions. A screenshot of the code used for this analysis is provided below.

```
# Define the target CPL reduction
target_reduction = 2
target_cpl = average_cpl - target_reduction

# Define a function to calculate the average change required for each feature
def calculate_feature_changes(target_cpl, features, X, model, scaler):
    # Calculate the baseline CPL
    baseline_cpl = model.predict(scaler.transform(X)).mean()

    # Placeholder for feature changes
    feature_changes = {}

    for feature in features:
        # Create a copy of the dataset
        X_copy = X.copy()

        # Increase the feature value slightly
        increment = 1e-1  # Larger increment for better sensitivity
        X_copy[feature] = X_copy[feature] + increment

        # Predict CPL with the adjusted feature
        baseline_predicted_cpl = model.predict(scaler.transform(X))
        adjusted_predicted_cpl = model.predict(scaler.transform(X_copy))

        # Calculate the change in CPL
        change_in_cpl = adjusted_predicted_cpl.mean() - baseline_predicted_cpl.mean()

        if change_in_cpl != 0:
            # Calculate the required change in feature to achieve the target reduction
            feature_changes[feature] = (target_reduction / change_in_cpl) * increment
        else:
            # Handle cases where no change in CPL is observed
            feature_changes[feature] = np.nan  # or a large value if preferred

    return feature_changes

# Continuous features to be optimized (excluding categorical features)
continuous_features = [col for col in X.columns if col not in categorical_columns]

# Calculate feature changes
feature_changes = calculate_feature_changes(target_cpl, continuous_features, X, model, scaler)

# Print results
print(f'Average required changes for each feature to achieve target CPL reduction:')
for feature, change in feature_changes.items():
    if np.isnan(change):
        print(f'{feature}: No significant change detected')
    else:
        print(f'{feature}: Change by {change}')

print(f'Target CPL: {target_cpl}')
print(f'Average CPL: {average_cpl}')
print(f'Difference from Average CPL: {target_reduction}')
```

Figure 10- Code implementation for calculating feature changes needed to achieve a target reduction in Cost Per Lead (CPL)

f. Implementation Tools

Various tools and libraries have been employed to ensure effective data processing, model implementation, and visualization. Each tool plays a specific role in the machine learning pipeline, contributing to the overall success of the project.

- Power BI: Power BI was used for creating interactive dashboards that offer real-time data visualization and KPI tracking. This tool facilitates informed decision-making and strategy optimization by providing an intuitive interface for exploring and analyzing data.
- Pandas & NumPy: These libraries are essential for data manipulation and preprocessing. Pandas handles structured data efficiently, enabling tasks such as data cleaning, transformation, and analysis. NumPy supports numerical operations, crucial for mathematical computations and handling large arrays.
- Scikit-learn: Scikit-learn provides comprehensive tools for training, evaluating, and fine-tuning machine learning models. It includes algorithms for classification, regression, and clustering, along with utilities for model selection and performance evaluation. This library is central to the machine learning workflow.
- SHAP: SHAP (SHapley Additive exPlanations) was employed to analyze feature importance within the models. It helps in understanding the model's behavior by showing how individual features impact predictions, thereby highlighting the most influential factors in decision-making.
- Jupyter Notebook: Jupyter Notebook was utilized for developing and documenting code. It supports interactive data analysis and model development, allowing for iterative coding, visualization, and documentation in a user-friendly environment.

4. Results

a. Model Validation and Selection

In the model validation and selection phase, the linear regression model demonstrated consistent performance across both validation and test datasets. The Mean Absolute Error (MAE) for the validation set was 10.49, while for the test set, it was slightly lower at 10.33. This close alignment in MAE values indicates that the model's accuracy in predicting average errors is stable across different data subsets. Additionally, the R-squared (R^2) values were 0.69 for the validation set and 0.68 for the test set, reflecting that the model explains approximately 68% to 69% of the variance in the target variable. These results underscore the model's robustness and its ability to generalize effectively to unseen data. The minor differences in MAE and R^2 between the validation and test sets are normal and expected, highlighting the model's reliable performance.

On the other hand, the decision tree model exhibited remarkable consistency and performance. The Mean Absolute Error (MAE) was 0.76 for the validation set and slightly higher at 0.80 for the test set, demonstrating that the model's accuracy in predicting average errors remains stable across different data subsets. The R-squared (R^2) values were exceptionally high, with 0.997 for the validation set and 0.997 for the test set, indicating that the model explains nearly 99% of the variance in the target variable. These metrics reflect the model's excellent fit to the data and its robust generalization capability. The minimal differences between validation and test metrics suggest that overfitting is not a significant concern, affirming that the model performs reliably across both seen and unseen data.

In evaluating the neural network model, the final training and test losses, alongside the performance metrics, reveal strong results. The final training loss was 0.65, while the final validation loss was slightly higher at 0.77. The test loss was notably lower at 0.45, suggesting good generalization without significant overfitting or underfitting. The model achieved a Mean Absolute Error (MAE) of 0.74, a Mean Squared Error (MSE) of 0.90, and a Root Mean Squared Error (RMSE) of 0.95, indicating that its predictions are close to the actual values with relatively small errors. The R-squared (R^2) value of 0.999 reflects an excellent fit, as the model explains nearly all of the variance in the target variable. These results underscore the model's robust performance and accuracy. For a visual representation of the model's training progress, refer to the image below, which depicts the training and validation loss over epochs.

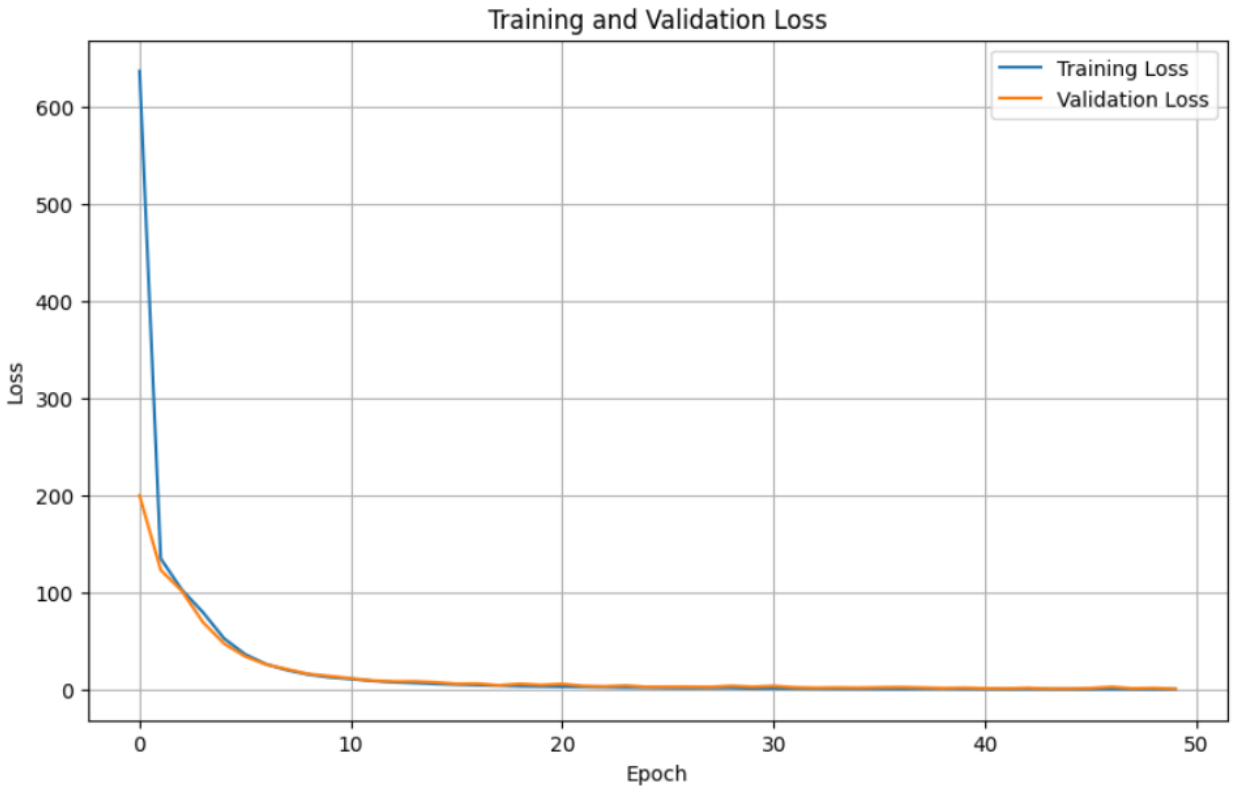

Figure 11- Training and Test Loss Over Epochs

Comparing the performance metrics of the linear regression, decision tree, and neural network models highlights distinct differences in accuracy and model fit. The linear regression model shows the highest error rates and the lowest R-squared value, indicating relatively poorer performance compared to the other models. In contrast, both the decision tree and neural network models exhibit similar and high levels of accuracy, with high R-squared values reflecting an excellent fit. This suggests that the decision tree and neural network models are more effective at capturing the variance in the target variable and delivering precise predictions. The image below illustrates this comparison in detail, highlighting the performance metrics side by side.

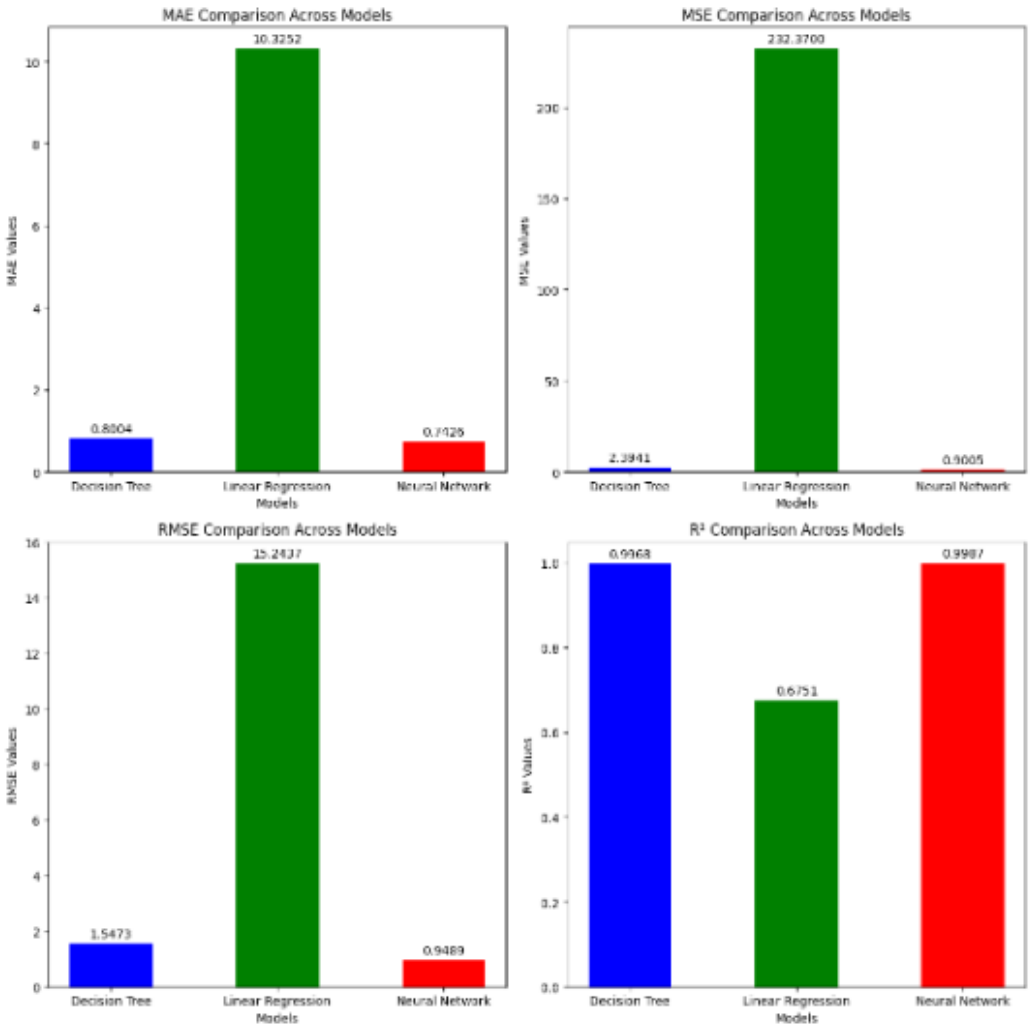

Figure 12- Performance comparison of Decision Tree, Linear Regression, and Neural Network models across key metrics

Given the results, it was advantageous to proceed with the neural network model. The neural network slightly outperformed the decision tree in terms of error rates and R-squared values, indicating a superior fit and more accurate predictions for the regression task at hand. Because decision trees are better suited for classification tasks, it was more appropriate to choose neural networks for this regression problem. The neural network's higher R-squared value and lower error metrics demonstrate its effectiveness in capturing the nuances of the data and making precise predictions.

b. Feature Impact Analysis Using SHAP Values

After selecting the neural network model, SHAP values were used to evaluate the impact of various features on Cost Per Lead (CPL) predictions. The SHAP summary plot, as shown in Figure

13, provides insights into the significance of each feature by illustrating their contributions to the model's predictions.

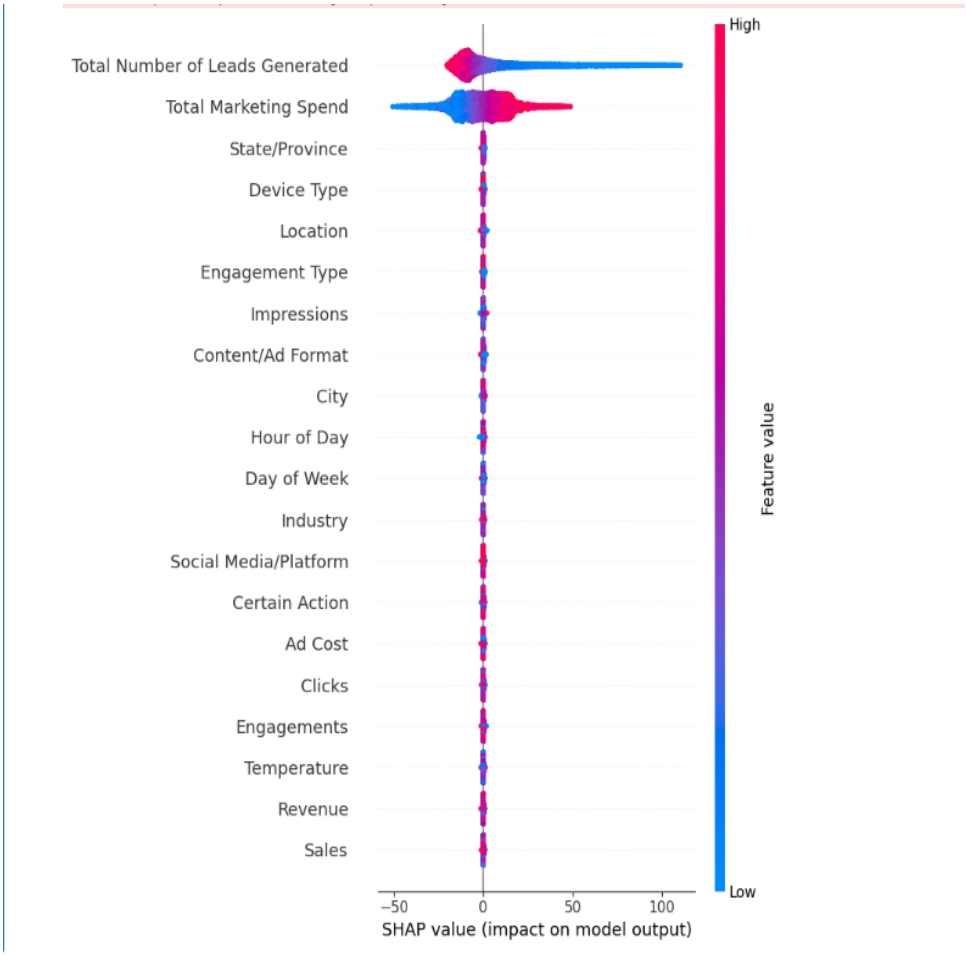

Figure 13- SHAP summary plot showing feature importance in CPL prediction

In the plot, features are ranked according to their influence on CPL. Each dot represents a SHAP value for a specific feature and instance, with dots on the right indicating a positive impact on CPL and dots on the left reflecting a negative impact. The color of the dots further enriches the analysis: red dots signify higher feature values, while blue dots denote lower values.

As depicted in the figure, the features with the most substantial impact on CPL are the total number of leads and total marketing spend, which aligns with logical expectations given their direct relevance to CPL. Conversely, features such as clicks, location, impressions, and sales show only minimal impacts. This nuanced understanding of feature importance is particularly valuable, and the use of neural networks proves advantageous here due to their ability to effectively capture and model complex relationships within the data.

c. Optimizing CPL

The first task in optimizing Cost Per Lead (CPL) was to analyze the impact of changes in specific features. To achieve this, a function was developed to predict CPL after modifying feature values. For instance, when the number of clicks was set to 100, the predicted CPL was 28.46, compared to an average CPL of 28.29. The difference between the new CPL and the average CPL was 0.17. An illustration of this analysis is provided in the image below.

```
Predicted CPL with Clicks set to 100: 28.461742401123047
Average CPL: 28.29186894677107
Difference between new CPL and average CPL: 0.16987345435197554
```

Figure 14- Impact of adjusting clicks on the predicted Cost Per Lead (CPL), with a comparison to the average CPL

This approach is valuable for performing sensitivity analysis by altering one feature and observing its impact on CPL. It facilitates decision-making and strategic adjustments, such as reallocating resources based on how significantly changes in features like clicks affect CPL. Also, with the model demonstrating a very high R-squared value of 0.999 and low error metrics (MAE, MSE, RMSE), the predictions are both accurate and reliable. This robust performance ensures that the analysis of feature impacts is grounded in precise and dependable results, supporting effective optimization and strategy formulation.

The second task involved determining how much each feature needs to be adjusted to achieve a target reduction in Cost Per Lead (CPL). This approach provides insights into how adjustments to individual features can impact CPL, highlighting which features are most influential in achieving the target reduction. By optimizing these features, you can refine marketing strategies and budget allocations effectively. The use of the model's predictions ensures that these adjustments are based on actual data, leading to more accurate and actionable recommendations. An example of these adjustments is provided below as an image.

```
Average required changes for each feature to achieve target CPL reduction:
Impressions: No significant change detected
Clicks: Change by -8738.133333333333
Ad Cost: No significant change detected
Certain Action: Change by -2912.711111111111
Sales: Change by -3744.9142857142856
Revenue: Change by 52428.8
Total Marketing Spend: Change by 388.3614814814815
Total Number of Leads Generated: Change by -9.213390738950883
Engagements: Change by -104857.6
Temperature: Change by -4194.304
Target CPL: 26.29186894677107
Average CPL: 28.29186894677107
Difference from Average CPL: 2
```

Figure 15- Required adjustments for each feature to achieve the target reduction in Cost Per Lead (CPL)

d. Power BI Dashboard for Marketing KPIs

The Power BI dashboard offers a comprehensive and interactive view of marketing performance through various Key Performance Indicators (KPIs) and metrics. It features essential data such as clicks, leads, sales, and profits, along with critical KPIs including Cost Per Click (CPC), Cost Per Lead (CPL), Cost Per Sale (CPS), Return on Investment (ROI), and Ad Cost to Sale ratio.

Users can explore KPIs across different industries, providing insights into how marketing effectiveness varies by sector, which aids in tailoring strategies and optimizing campaigns. Additionally, the dashboard includes comparative matrices for metrics like impressions, revenue, clicks, ad costs, sales, and leads across various social media platforms. This allows for the assessment of platform performance and informed decisions on resource allocation.

By visualizing these KPIs, users can evaluate marketing spend efficiency and make strategic adjustments to improve ROI. The ability to analyze data by industry and platform helps refine marketing strategies and optimize budgets, ensuring resources are directed towards the most effective channels.

Overall, the dashboard's features offer valuable insights and actionable data, enhancing strategic decision-making and marketing effectiveness. A screenshot of the dashboard, highlighting its layout and functionality, is shown in Figure 16.

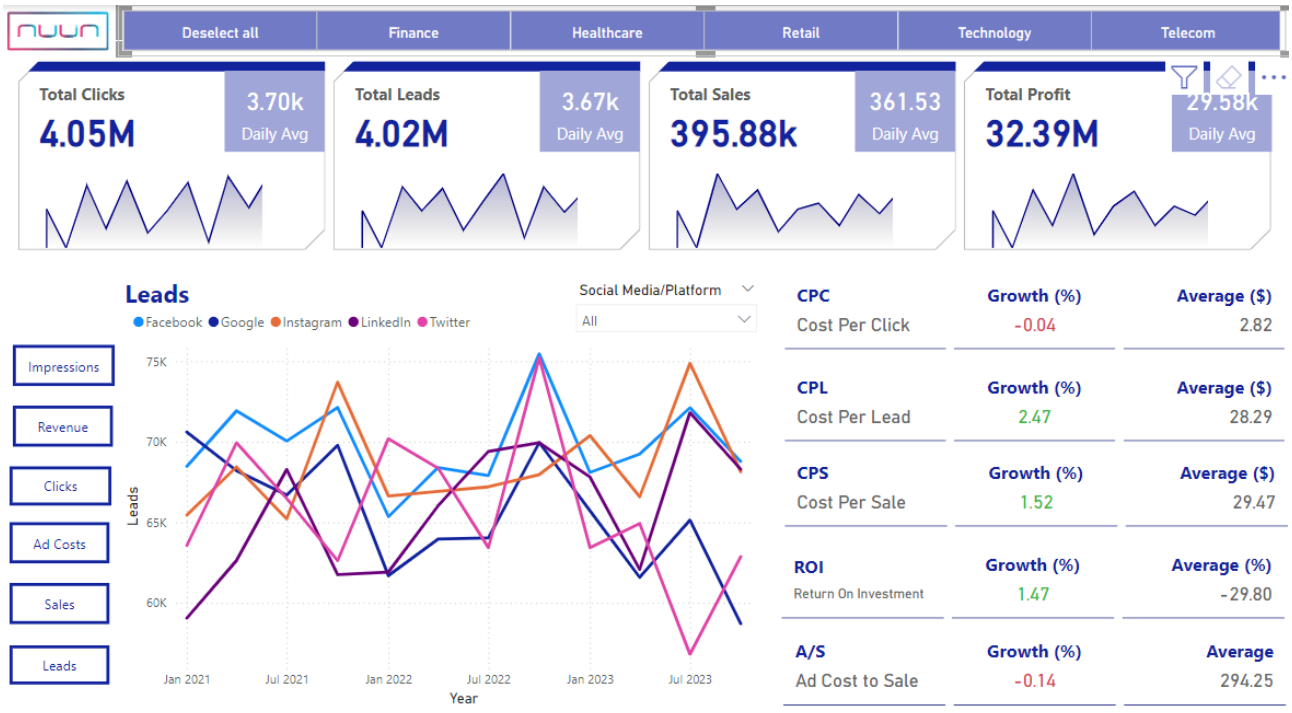

Figure 16- Power BI Dashboard showcasing marketing KPIs

5. Discussion

The analysis of the models employed in this project has highlighted both their strengths and limitations in predicting Cost Per Lead (CPL). Among the evaluated models, the neural network emerged as the most effective tool, demonstrating superior performance metrics, including a notably high R-squared value and low error metrics. This exceptional performance underscores the neural network's ability to capture and model complex, non-linear relationships within the data, making it particularly adept at predicting CPL. The neural network's robustness in delivering accurate predictions positions it as a valuable asset for optimizing marketing strategies.

The feature impact analysis, utilizing SHAP values, provided critical insights into how various features affect CPL. The analysis revealed that features such as the total number of leads and total marketing spend have a substantial and direct influence on CPL. These findings emphasize the importance of focusing on these key features to effectively optimize CPL. Conversely, other features demonstrated minimal effects on CPL, underscoring the need to prioritize adjustments in the most impactful areas. However, a notable limitation was identified: the current dataset lacks strong correlations with CPL. This limitation suggests that the model's accuracy and the quality of insights could be significantly improved with more robust data. To address this, it is essential for the company to continuously expand and enhance its data sources. Incorporating additional data with a strong relationship to CPL will refine the model's performance and yield more actionable predictions. Therefore, ongoing efforts to enrich the dataset are crucial for maintaining and improving the model's effectiveness.

The optimization analysis highlighted that even minor adjustments to features can lead to significant changes in CPL. This sensitivity analysis underscores the importance of precise feature management and targeted adjustments. By focusing on the most influential features and implementing data-driven changes, the company can achieve notable reductions in CPL. The model's high accuracy and reliability ensure that these adjustments are based on solid, actionable data, facilitating more effective decision-making.

A critical aspect of maintaining the model's efficacy is the regular updating of data. As market conditions and customer behaviors evolve, the factors influencing CPL may change over time. Features that impact CPL today might not be as influential in the future, and vice versa. To keep the model relevant and effective, it is essential to continuously update the dataset, optimize the

model, and refresh the dashboard. This ongoing process will ensure that the insights derived remain accurate and actionable, supporting strategic adjustments and improving marketing performance.

Objective criticism of the outcomes reveals that while the project met several key objectives, there were areas for improvement. Notably, the project did not directly compare CPL metrics with industry standards and benchmarks due to a lack of resources on this matter. Such a comparison would have provided valuable context regarding Nuun Digital’s competitive position and insights into how its CPL measures up against industry norms. Additionally, the focus was limited to lead acquisition cost (cost per lead) rather than customer acquisition cost because of data constraints. Addressing these gaps by incorporating industry comparisons and expanding data sources will enhance the analysis's comprehensiveness and provide a clearer picture of the company's market position.

Overall, the project has made significant strides in optimizing CPL through advanced machine learning techniques and actionable insights. However, ongoing efforts to expand data sources and incorporate industry comparisons will further refine the model's effectiveness and provide more valuable recommendations for the company's marketing strategy.

6. Conclusion

This project has achieved significant advancements in optimizing Cost Per Lead (CPL) through the application of machine learning techniques. By harnessing the power of predictive modeling, the analysis has yielded critical insights and actionable recommendations that can substantially enhance CPL management. The comprehensive evaluation of various models demonstrated the neural network's exceptional capability to predict CPL with high accuracy, reflecting its ability to capture complex, non-linear relationships in the data.

The findings underscore the importance of ongoing data updates and model refinement to ensure that the insights remain relevant and actionable. Continuous data enrichment and periodic adjustments to the model will be pivotal in maintaining the accuracy of predictions and adapting to shifts in market conditions. This proactive approach will support the development of more effective marketing strategies and facilitate more efficient lead acquisition processes.

The commitment to refining the model and integrating new data is not merely a recommendation but a necessity for sustained improvement. By embracing these practices, the company will be better equipped to respond to evolving market dynamics, optimize marketing expenditures, and achieve superior outcomes in lead acquisition. Ultimately, this project highlights the transformative potential of machine learning in driving strategic decisions and operational efficiencies, paving the way for more informed, data-driven marketing strategies that can adapt to the ever-changing landscape of customer acquisition.

7. References

1. Olausson, M. (2018). Prediction of conversion rates in online marketing: A study of the application of logistic regression for predicting conversion rates in online marketing.
2. Chaudhari, P., Dingankar, R., & Chaudhari, R. (2013). Prediction of CPC using neural networks for minimization of cost. *International Journal of Computer Theory and Engineering*, *5*(4), 457-463.
3. Darapaneni, N., Paduri, A. R., Shukla, S., Dwivedi, K., Sharma, S., Saha, S., Kumar, S., & Gupta, A. (2023). Cost prediction in acquiring customers using machine learning. In *Proceedings of the International Conference on Recent Trends in Data Science and its Applications* (pp. 447-456).
4. Prasad, G. L. V., Bunga, A., Kishore, S., Nanda, A. D. S., Tadepalli, S. K., & Saini, Y. (2024). Machine learning based cost prediction for acquiring new customers. In *Proceedings of the International Conference on Recent Trends in Data Science and its Applications*. IEEE Xplore.
5. Bondarenko, S., Laburtseva, O., Sadchenko, O., & Vira. (2019). Modern lead generation in internet marketing for the development of enterprise potential. *International Journal of Innovative Technology and Exploring Engineering (IJITEE)*, 8(12), 2278-3075.