RESTAURANT REVENUE ANALYSIS

https://github.com/AdityaPatel1068/DMA

Problem Statement

In the ever-evolving restaurant industry, making informed decisions is crucial to success. When a company is planning to open new restaurants, they need a comprehensive analysis of the factors influencing restaurant revenue. Our specific mandate involves the execution of an in-depth analysis, employing sophisticated Machine Learning techniques, Regression analysis, and the Neural network methodology. The ultimate aim is to yield actionable insights that are instrumental in the optimization of revenue stream.

Analysis and Key Findings

Variables selection:

In our analysis, we carefully examined each of the provided features to determine their correlation with restaurant revenue. The dataset includes the following attributes:

1. Restaurants App:
    a. Relation: Negative
    b. Influence: Strong
    c. Strategic Plan: Optimizing the restaurant's services and presence on the app can significantly contribute to the growth of restaurant revenue.
2. Restaurants App:
    a. Relation: Negative
    b. Influence: Strong
    c. Strategic Plan: Optimizing the restaurant's services and presence on the app can significantly contribute to the growth of restaurant revenue.
3. Distance from Town Center:
    a. Relation: Negative
    b. Influence: Strong
    c. Strategic Plan: Minimizing the distance from the town center is imperative. It increases foot traffic, which, in turn, positively impacts the monthly revenue of the restaurant.

Importance of Factors:

The importance of these factors in influencing restaurant revenue cannot be overstated. Here's a summary of their significance:

1. **Location (Town Population, Distance from Town Center):** These factors play a pivotal role. Targeting areas with high population density and minimizing the distance from the town center are essential for revenue optimization.
2. **Customer Experience (Outdoor Dining):** Providing an appealing outdoor dining experience is crucial for boosting monthly revenue. Customers value this aspect, and it has a strong positive influence.
3. **Technology and App Presence (Restaurants App):** Optimizing services and the restaurant's presence on apps is essential in today's digital age. While it has a negative relation, it can significantly impact revenue if managed effectively.
4. **Service Quality (Take Out Service):** While it has a positive relation, the influence is moderate. Enhancing takeout services can still be beneficial but may not be as influential as other factors.
5. **Income Area (Service Area Income):** While positively correlated with monthly revenue, it doesn't have as strong an impact as population and location. However, targeting areas with higher income can be considered as part of revenue optimization.

Regression Model:

In this section, we'll delve into the fundamental components of our analysis, which employed both Linear Regression and Neural Network models. Our dataset, comprising a mere 100 rows, posed unique challenges due to its limited size and simplicity.

Data Size Consideration:

- It's essential to address that the dataset contained only 100 rows.
- To evaluate the models, we divided the dataset into an 80-20 split for training and testing.
- *A crucial point to note is that in scenarios with very limited data, models can tend to overfit, especially if the dataset is small and straightforward*.

Desired outcome:

Required output is Maximizing Restaurant Revenue through Informed Decision-Making. By implementing these data-driven strategies and utilizing both regression analysis and neural network methodologies, the restaurant chain can make informed decisions, optimize revenue, and stay competitive.

Success Matrix:

Determining the success of your analysis and models is, without a doubt, critical for evaluating the effectiveness of our tactics and decision-making. Here are some crucial success measures to consider in the context of our problem statement:

- Mean Squared Error (MSE)
- R-squared (R^2)
- Correlation
- P- Value

Methodology:

Data Collection:

The dataset for this analysis was procured internally. The data encapsulates key metrics like Restaurant unique identifier, Take Out Service, Area Income, availability of Out-Door Dining, part of an App network, Distance from Town Center and monthly revenue. The Data set was uploaded on the GitHub repository for easy access through various python libraries.

Data Cleaning:

The primary aim of our preprocessing stage was to ensure data accuracy and clarity.

Column Selection: We initiated the process by carefully selecting and retaining columns that were deemed relevant to the analysis, eliminating any unnecessary or redundant data.

Handling missing values: Impute missing data using techniques like mean, median, or interpolation.

Handling Outliers: Identifying and handling outliers using IQR and replacing them by mean value.

Data Transformation:

Transforming data in order to fit into predefined constraints and scale numerical features to a standard range by standardization.

Feature scaling: Scale numerical features to a standard range by standardization. Here we divide the dataset into target and feature columns.
Later we fit and transfer the featured column of the data set to build the model.

Data Splitting: Data splitting is a technique commonly used in machine learning and data analysis to divide a dataset into distinct subsets for different reasons. The most common application is to divide a dataset into training and testing sets in order to evaluate the performance of a machine learning model. Here, we are considering 80 percent as training data and the remaining 20 percent as testing data.

Model Selection:

Model selection is an important phase in the machine learning workflow in which you select the optimal method or model for your particular problem. A model that works well with one sort of data may not work well with another.
Here, we choose Machine Learning models Linear Regression and Neural Networks to build the model and know the accuracy.

Model Training:

We conducted training on the chosen model using the training dataset, focusing on learning data patterns and relationships. Given the limited dataset size, it was important to prevent overfitting. To address this, we carefully selected the number of epochs, narrowing it down to a range between 437 and 500. We also employed a batch size of 5. These adjustments were made to minimize the loss function and enhance the predictive capabilities of the neural network.

Tools and Techniques:

Software Tools: Jupiter notebook, Excel file, GitHub and VsCode

Techniques: We use Python libraries like sklearn, matplotlib, numpy, keras, and requests.
To develop the model and evaluate the results.

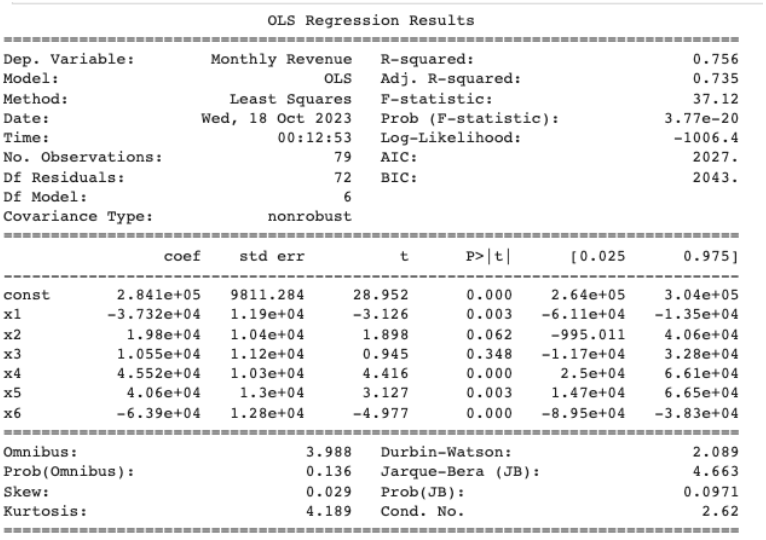
Model Evaluation:

Metrics Used

1. Mean Squared Error (MSE): MSE measures the average squared difference between the actual and predicted values. Lower MSE values indicate a better fit to the data.
2. R-squared (R^2): R^2 represents the proportion of the variance in the dependent variable that is explained by the independent variables. An R^2 value closer to 1 indicates a better model fit.
3. Correlation P-Value: The p-value associated with the correlation coefficient helps determine the statistical significance of the relationships between variables. A low p-value suggests a strong correlation.

Linear Regression Evaluation

1. The MSE is 8,297,763,755.27, which indicates the average squared difference between the actual and predicted values. Lower MSE values are better. In this case, the MSE is relatively high.
2. The R^2 value is 0.544, which represents the proportion of the variance in the dependent variable that is explained by the independent variables. An R^2 value closer to 1 indicates a better fit. In this case, an R^2 of 0.544 suggests that linear regression explains a moderate amount of variance in your data.



OLS Regression Results

Dep. Variable:	Monthly Revenue	R-squared:	0.756
Model:	OLS	Adj. R-squared:	0.735
Method:	Least Squares	F-statistic:	37.12
Date:	Wed, 18 Oct 2023	Prob (F-statistic):	3.77e-20
Time:	00:12:53	Log-Likelihood:	-1006.4
No. Observations:	79	AIC:	2027.
Df Residuals:	72	BIC:	2043.
Df Model:	6		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	2.841e+05	9811.284	28.952	0.000	2.64e+05	3.04e+05
x1	-3.732e+04	1.19e+04	-3.126	0.003	-6.11e+04	-1.35e+04
x2	1.98e+04	1.04e+04	1.898	0.062	-995.011	4.06e+04
x3	1.055e+04	1.12e+04	0.945	0.348	-1.17e+04	3.28e+04
x4	4.552e+04	1.03e+04	4.416	0.000	2.5e+04	6.61e+04
x5	4.06e+04	1.3e+04	3.127	0.003	1.47e+04	6.65e+04
x6	-6.39e+04	1.28e+04	-4.977	0.000	-8.95e+04	-3.83e+04

Omnibus:	3.988	Durbin-Watson:	2.089
Prob(Omnibus):	0.136	Jarque-Bera (JB):	4.663
Skew:	0.029	Prob(JB):	0.0971
Kurtosis:	4.189	Cond. No.	2.62

Neural Network Evaluation

For the neural network model, trained on a dataset consisting of only 100 rows and 6 columns, the evaluation raised several noteworthy considerations.

1. The MSE is higher at 11,184,944,438.82. This suggests that the neural network model is less accurate on your dataset compared to linear regression.

2. The R^2 value is lower at 0.386, indicating that the neural network model explains less of the variance in the data compared to linear regression.

- Epochs and Batch Size:
  The model was trained with 500 epochs and a batch size of 5.
  During the training process, we observed that the mean squared error (MSE) approached minima point. However, the exact nature of this minima—whether it was a global minima or local minima—remained uncertain.

- Overfitting Concern:
  The use of neural networks, particularly on a small dataset, inherently poses a risk of overfitting. Overfitting occurs when a model learns to fit the training data too closely, capturing noise rather than genuine patterns.
  In our case, given the limited dataset size, the model indeed had a high potential for overfitting. A more complex model could easily adapt to the small dataset, causing it to perform poorly on unseen data.

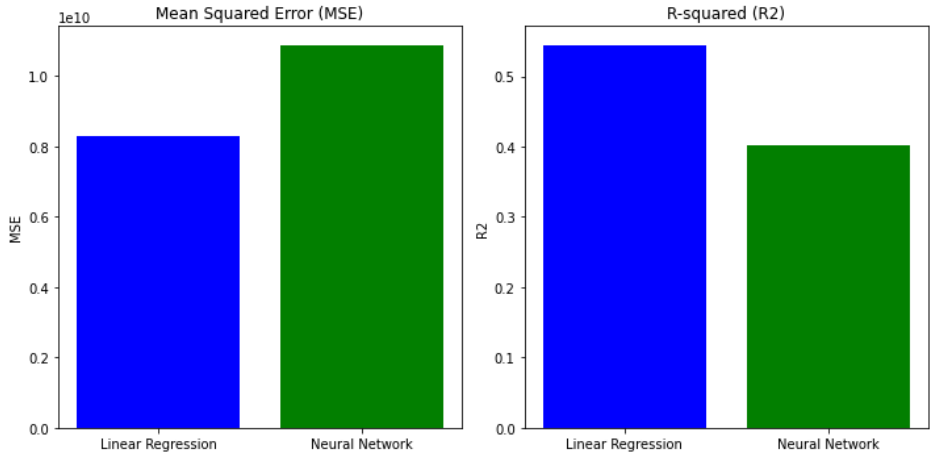
- Model Robustness:
  To address overfitting, we carefully adjusted the number of epochs and batch size. While this mitigated overfitting to some extent, it is important to note that even with these adjustments, the model still had a high likelihood of overfitting.
  The trade-off between model complexity and dataset size is a challenge in such scenarios. Simplistic models may underfit, while complex models may overfitting.

```
Model: "sequential_1"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 dense_3 (Dense)             (None, 64)                448

 dense_4 (Dense)             (None, 32)                2080

 dense_5 (Dense)             (None, 1)                 33

=================================================================
Total params: 2561 (10.00 KB)
Trainable params: 2561 (10.00 KB)
Non-trainable params: 0 (0.00 Byte)
_________________________________________________________________
```



Overfitting and Underfitting

In the evaluation of our models, particularly the neural network, we encountered a notable challenge related to overfitting. Overfitting occurs when a model becomes excessively tailored to the training data, capturing not just meaningful patterns but also noise. Several factors contributed to the overfitting observed in our analysis:

Simple Data:

The simplicity of the dataset played a significant role in overfitting. When the data is straightforward and lacks complexity, it can be challenging to develop a model that generalizes well to unseen data. Our dataset contained a limited number of features, which could make the model more prone to overfitting.

Small Data Size:

With only 100 rows, the dataset's size was relatively small. Small datasets are more susceptible to overfitting because there is insufficient data to capture the full range of variability in the target variable. Models may learn to memorize the training data instead of learning underlying patterns.

Column Value Ranges:

A noteworthy characteristic of our dataset was the varied value ranges within columns. While some columns had values between 0 and 1, others exhibited a much broader range from 1000 to 100,000. This diversity in value scales can make it challenging for the model to learn a consistent representation of the data.

Experiment results:

In our analysis, both the neural network and linear regression models were applied to predict restaurant prices using unseen raw data. These models were tested on a subset of restaurant data, with the aim of understanding their predictive capabilities. The evaluation of these models provided insights into their performance in making price predictions, offering valuable insights for decision-making in the restaurant industry.

Restaruants	App	Take Out Servce	Area Income	Town Pop	OutDoor Dining	Distance from Town Center	Revenue with Linear regression	Revenue with Neural networks
PPI	0	1	$56,000	29,000	1	9	154017.941560	29168.218750
TRE	0	0	$110,000	115,000	0	1	398779.116627	386294.937500
GGT	1	1	$67,000	252,000	0	3	365408.652827	348224.218750
MND	0	0	$105,000	71,000	0	2	327279.894986	288490.406250
WRT	0	1	$67,000	150,000	1	1	549942.658990	523541.343750
GFR	0	1	$74,000	120,000	0	7	197322.649469	175177.031250
WWW	1	1	$130,000	75,000	1	2	410709.133116	314757.781250
QWE	0	0	$51,000	100,000	0	4	236309.735477	179476.875000
FGR	1	0	$45,000	47,600	0	3	171534.572003	152448.843750
SSC	0	1	$72,000	91,000	1	11	128441.842383	20471.687500
SAE	1	1	$85,600	68,000	1	4	310227.451432	119573.617188