CSE343/ECE343 Machine Learning, Monsoon 2023
End-Sem Project Evaluation

Aditya Girdhar 2021005

Aditya Raj 2021512

Mehar Khurana 2021541

Nalish Jain 2021543

Abstract

In this report, we explore two primary machine learning models: Linear Regression and Random Forest Regression, to predict meal sales at IIIT Delhi's dining facilities accurately. We also investigate the impact of feature engineering, including one-hot encoding of menu items, on model performance. Our analysis showcases the importance of feature engineering and the choice of encoding techniques in improving model accuracy.

1. Introduction

In educational institutions like IIIT Delhi, predicting daily meal sales is a crucial task for mess management. Accurate forecasts enable the optimization of food supply, minimize wastage, and ensure efficient resource allocation. In this context, machine learning offers valuable tools to develop predictive models.

This research project focuses on building and evaluating machine learning models to predict meal sales in the IIIT Delhi mess. We analyze a comprehensive dataset that includes features such as date, meal type, payment methods (Paytm+Cash), and coupon usage. To enhance our models' predictive power, we engineer additional features, including the day of the week, academic semester type, availability of specific menu items and the number of coupons solds in the mess.

We employ four machine learning models: Linear Regression, Random Forest Regression, Support Vector Regressor and Neural Networks to forecast meal sales accurately. These models enable us to understand the relationship between various factors and meal sales, aiding in better decision-making for mess management.

Additionally, we investigate the impact of different encoding techniques, particularly one-hot encoding for menu items, on model performance. This exploration allows us to identify the most effective strategies for feature representation and improve prediction accuracy.

2. Literature Review

In this section, we go through some of the work done in predicting food demand in various places.

2.1. Predicting food demand in food courts

In this study, the authors addressed the challenges of fluctuations and unpredictability in food demand within public food courts and proposed the use of three decision tree methods (CART, CHAID, and Microsoft Decision Trees) to predict consumption demand for specific menu items on selected dates. They analyzed a two-year dataset from food courts at Hacettepe University, achieving prediction accuracies up to 0.83 in R2. The study demonstrated the suitability of decision tree methodology for food consumption prediction in such contexts, emphasizing the potential benefits for optimizing the balance between food supply and demand. The authors discussed the relevance of data mining techniques in solving resource optimization problems and highlighted the limited application of these methods in the food consumption field compared to other domains. They also compared the performance of the three decision tree algorithms, with CHAID yielding the most accurate predictions on average, followed by Microsoft Decision Trees and CART.

2.2. Predicting restaurant sales

This article explored the utilization of machine learning for predicting restaurant sales through the use of Azure Machine Learning tools. The article underscored the importance of precise sales forecasts for efficient restaurant management and categorized machine learning into various types, with a specific focus on Supervised Machine Learning. The step-by-step guide detailed procedures for data import, cleaning, model training, and evaluation, ultimately identifying Decision Forest Regression as the most suitable algorithm. This article contributed to the existing body of literature by providing a practical demonstration of how machine learning techniques and Azure tools can enhance restaurant sales forecasting.

3. Dataset

3.1. Original Features

1. **Date**: This is the date the sample was recorded on.

2. **MealType**: This refers to the meal the data sample is recorded for, e.g., 'Breakfast'.

3. **Paytm+Cash Amount**: This is the total sale amount recorded during a meal.

4. **Coupons**: This refers to the total number of Coupons scanned during a given meal.

3.2. Engineered Features

1. **Weekday**: This feature is derived from the date, with 0 representing Mondays, 1 representing Tuesdays, and so on. The idea behind including this feature is that the daily foot traffic would be heavily dependent on what day of the week the data is recorded on.

2. **Month**: This feature is derived from the date, with 1 representing January, 2 representing February, and so on.

3. **Paytm+Cash**: This denotes the total number of plates ordered using cash or PayTM. This data is derived using the 'Paytm+Cash Amount', or the total sale amount recorded, and the price of one plate for the corresponding MealType.

4. **SemType**: This feature denotes whether an Academic semester is going on at the time of recording the current meal sales. It can take two values, 'Acad' when the Winter and Monsoon semesters are underway, and 'Vacation' when a vacation is ongoing. The sales would be diminished during vacations when most students are not in the hostels.

5. **Holiday**: This denotes the number of consecutive days off during a holiday break. Weekends take this value as 1 (instead of 2). We believe that adding this feature would help in predicting sales, as a higher value for this feature would result in reduced sales since students are more likely to go back home during longer breaks.

6. **CouponsMand**: This denotes whether coupons were mandatory(1) or not(0). In case of special cases when coupons were mandatory for a section of students we assigned fractional values representing that section of students.

7. **MenuItem**: This is the one-hot encoding of the food items being offered in the corresponding meals. We believe that the items being offered would influence the sales of meals. Implementation: created a dictionary to store month-wise weekly menu. Assigned unique indices to the menu items and converted the menu to a 122-long vector, storing 122 unique food items. A set bit (1) in this vector signifies that the corresponding item was offered in the meal.

8. **Coupon Counts**: These are the cumulative number of coupons for each of the 4 categories (15, 20, 25, 30), which denote the number of days for which a student bought the coupons, for a given month. These counts are reset at the end of every month. The idea behind this feature is that a high number of coupons bought would increase the likelihood of coupons being used in any meal for a given month.

3.3. Visualization

Figure 1: Distribution over different features

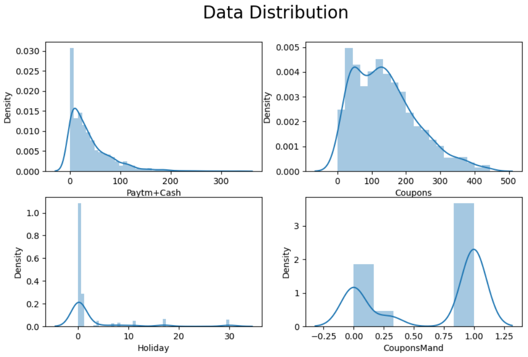

Figure 1 shows the density distribution of different features. Figure 2 shows the correlation heatmap of the features in the dataset. Holiday correlates negatively with sales (PayTM+cash and coupons).

Figure 2: Heatmap for correlation

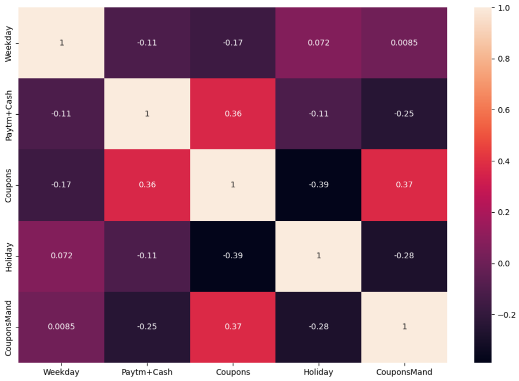

Figure 3 shows the sales of two food items, Matar Kulcha and Dal Makhni. The graphs also have labels for various holiday breaks, (i.e. when students tend to go back home), correlating with the dips in sales. This correlation confirms the accuracy of the data collected.

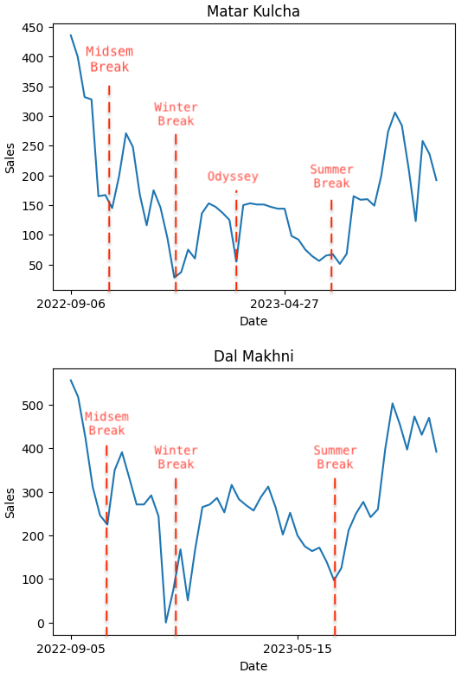

Figure 3: Sales for two food items over the year

4. Methodology

4.1. Linear Regression

Initially, we employed a Linear Regression model to forecast the mess's daily foot traffic, utilizing a set of features that encompassed ’Month,’ ’Weekday,’ ’MealType,’ ’SemType,’ ’Holiday,’ and ’CouponsMand.’ To ensure robust model evaluation, we partitioned the data into training and test sets using a 92% to 8% split, roughly representing 30 days of data (with the last month serving as the test dataset). We defined three target variables ’Paytm+Cash’, ’Coupons’ and ’Both’ which represents the sum of the first two target variables. Notably, we encoded ’Month,’ ’Weekday,’ ’MealType,’ and ’SemType’ using both one-hot encoding and numerical encoding to convert categorical variables into a format suitable for modeling. Subsequently, after evaluating the initial model's performance, we introduced a new feature, ”Menu Item,” into our dataset with both initial numerical and one-hot encoded features. This additional feature was meticulously one-hot encoded to incorporate the specific menu choices into our predictive framework, refining the model for more accurate estimates of mess attendance.

4.2. Random Forest Regression

Utilized the Random Forest Regressor from the sklearn library to forecast the mess foot traffic. Feature selection was similar to the linear regression model - ’Day’, ’Month,’ ’Weekday,’ ’MealType,’ ’SemType,’ ’Holiday,’ and ’CouponsMand’. Our analysis involved four datasets: the first and third excluded the ’MenuItem’ feature. We tried both one-hot encoding and numerical encoding to represent the categorical features, including the ’MenuItem’ feature, due to the insights gained in the Mid-Semester report. For the latter two datasets, we introduced the ’Coupon Count’ feature, which was further segmented into ’15_coupon_count’, ’20_coupon_count’, ’25_coupon_count’ and ’30_coupon_count’.

The train-test split constituted data from Sept’22 to end of Aug’23 for training and Sept’23 for testing. To ensure optimal training of the model, we ran a grid search by splitting the training set into training and validation (80:20) to optimize the parameters ’n_estimators’ (number of decision trees formed), ’max_depth’ (of each tree), ’min_samples_split’ (minimum samples to split an internal node), ’min_samples_leaf’ (number of samples at leaf node) and ’max_features’ (number of features considered during a split).

4.3. Support Vector Regression

To enhance predictive accuracy, we employed the Support Vector Regressor from the scikit-learn (sklearn) library for forecasting mess foot traffic. The initial feature selection mirrored that of the linear regression model, encompassing ’Day,’ ’Month,’ ’Weekday,’ ’MealType,’ ’SemType,’ ’Holiday,’ and ’CouponsMand.’ Our analysis involved four datasets: the first and third excluded the ’MenuItem’ feature. We utilized one-hot encoding to represent the categorical features.

For the latter two datasets, we introduced the ’Coupon Count’ feature, which was further segmented into ’15_coupon_count’, ’20_coupon_count’, ’25_coupon_count’ and ’30_coupon_count’. o ensure ro- bust model evaluation, we partitioned the data into training and test sets using a 92% to 8% split, roughly represent- ing 30 days of data.

The models underwent thorough hyperparameter tuning until improvements in accuracy plateaued.

5. Result and Analysis

5.1. Linear Regression

An initial predictive model was developed for the purpose of forecasting daily traffic within IIIT Delhi's dining facilities, colloquially referred to as a ”mess.”

Our first model’s predictive capacity was constructed through the utilization of a Linear Regression framework. In the first phase of our analysis, we started by testing our model using the raw features without any additional encoding. The primary goal of this phase was to establish a baseline for our model’s performance.

Building upon the insights gained from the initial model, we decided to explore the potential benefits of one-hot encoding the categorical features.

The results of our experiments in the midsem report clearly indicated the benefits of one-hot encoding categorical features in our forecasting model. The inclusion of the one-hot encoded ’MenuItem’ along with ’Coupon Count’ feature was a very strategic refinement, giving the predictive framework an increased capacity to capture nuanced variations in mess attendance attributable to specific culinary offerings, as the decision of students to avail dining facilities often hinges on what’s being served that day and the number of coupons sold in the mess in that month.

The outcomes of all conducted experiments have been presented comprehensively within Table 1 and Table 2. Evidently, a discernible pattern emerges, manifesting as a progressive improvement in model performance. This improvement is notably characterized by a declining trend in Root Mean Squared Error (RMSE) values and a simultaneous increasing trend in R^2 values as one travels down the table. If we have to use only one out of ’MenuItem’ and ’Coupon Count’, ’MenuItem’ is a better option since it almost gives maximum performance (almost as much as when using both).

Although linear regression gives very solid results, we strongly believe that we can do better by trying out non-linear models to capture more complex patterns in the data, as can be seen in further models we tried

5.2. Random Forest Regression

The results of the model on the above mentioned four data sets are summarized in Table 4:

The categorical features were numerically encoded for Model 1. For Model 2, MenuItem was one-hot encoded and other categorical features were numerically encoded. For Model 3, categorical features were one-hot encoded. For Model 4, both MenuItem and other categorical features were one-hot encoded.

The trends are harder to distinguish for the Random Forest Regression model, however, they are still present. Addressing the issue of one-hot encoding, it is clear that with the inclusion of this technique, the difference between the model’s performance on the training and testing data increases (especially with the MenuItem feature) for the Coupons and Total target variables. This may be due to the curse of dimensionality, where the increased dimensionality of the data leads to a more sparse representation. This leads to the model being unable to find meaningful patterns and overfitting to the training data. Despite this however, it was found upon testing, that one-hot encoding the MenuItem gave better results than numerically encoding it. This may be because of the way Random Forest Regressor splits numerical values in a feature (by considering intervals of values) resulting in not meaningful splits. Other categorical features interchangably performed better or worse with numerical and one-hot encoding. This may be due to their relatively smaller number and cardinality.

Table 1: LR Model Training Performance Metrics

Model	RMSE	R^2
Dataset without ’MenuItem’, (Model 1)		
Paytm+Cash	26.31	0.62
Coupons	50.38	0.7
Both	59.71	0.72
Dataset with ’MenuItem’, (Model 2)		
Paytm+Cash	22.83	0.720
Coupons	45.93	0.76
Both	47.24	0.81
Dataset with ’Coupon Count’, (Model 3)		
Paytm+Cash	22.62	0.53
Coupons	45.89	0.72
Both	54.71	0.71
Dataset with ’MenuItem’ and ’Coupon Count’, (Model 4)		
Paytm+Cash	18.57	0.72
Coupons	37.96	0.81
Both	41.83	0.83

Table 2: LR Model Testing Performance Metrics

Model	RMSE	R^2
Dataset without ’MenuItem’, (Model 1)		
Paytm+Cash	24.12	0.62
Coupons	57.09	0.64
Both	69.68	0.63
Dataset with ’MenuItem’, (Model 2)		
Paytm+Cash	19.83	0.74
Coupons	43.90	0.78
Both	47.78	0.83
Dataset with ’Coupon Count’, (Model 3)		
Paytm+Cash	22.47	0.51
Coupons	48.64	0.66
Both	58.11	0.59
Dataset with ’MenuItem’ and ’Coupon Count’, (Model 4)		
Paytm+Cash	19.99	0.61
Coupons	39.7	0.77
Both	45.06	0.72

Attempts were made to increase the training error (bias)

Table 3: RFR Model Training Performance Metrics

Model	RMSE	R^2
Dataset without 'MenuItem', (Model 1)		
Paytm+Cash	10.81	0.93
Coupons	27.59	0.88
Both	36.15	0.84
Dataset with 'MenuItem', (Model 2)		
Paytm+Cash	10.82	0.93
Coupons	18.85	0.95
Both	22.98	0.95
Dataset with 'Coupon Count', (Model 3)		
Paytm+Cash	10.09	0.91
Coupons	28.78	0.87
Both	35.91	0.85
Dataset with 'MenuItem' and 'Coupon Count', (Model 4)		
Paytm+Cash	13.30	0.84
Coupons	25.17	0.89
Both	24.13	0.89

Table 4: RFR Model Testing Performance Metrics

Model	RMSE	R^2
Dataset without 'MenuItem', (Model 1)		
Paytm+Cash	11.55	0.91
Coupons	54.12	0.65
Both	59.16	0.74
Dataset with 'MenuItem', (Model 2)		
Paytm+Cash	11.80	0.903
Coupons	52.25	0.679
Both	53.82	0.788
Dataset with 'Coupon Count', (Model 3)		
Paytm+Cash	10.34	0.92
Coupons	54.85	0.64
Both	61.26	0.72
Dataset with 'MenuItem' and 'Coupon Count', (Model 4)		
Paytm+Cash	12.08	0.89
Coupons	60.96	0.56
Both	63.16	0.71

and thereby decrease the testing error (variance), however, it was found that an increase in the training error corresponded to an increase in the testing error as well. Many values of the regularizing parameters (max_depth, min_samples_split, min_samples_leaf, max_features) were tested, but the best test set results were as reported.

From the trend of the RMSE values of the predictions for the Paytm+Cash target variable before and after the addition of the 'Coupon Count' features, we can surmise that it does indeed improve the model overall. However, it doesn't seem to impact the performance of the Coupons and Total target variables very much. MenuItem seems to improve Coupons and Total to some degree. For the discrepancy between the RMSE values of Paytm+Cash and Coupon predictions, it can be reasoned that the relationship between the Paytm+Cash target variable and the utilized parameters is relatively simple, and is easily captured by the Decision Trees in the Random Forest Regressor. However, the relationship between the Coupons target variable (and thus Total) and features like Coupon Count and MenuItem is seemingly more complex, and the feature-value splitting mechanism of a Decision Tree simply isn't able to capture this relationship and ends up having to memorize the training data. This can be confirmed by the fact that Coupon Count seems to improve the Support Vector Regressor with Gaussian Kernel dramatically and has a much better score on the Coupons target variable highlighting its higher expressivity. However, it performs worse than Random Forest Regressor on the Paytm+Cash target variable, indicating a simpler relationship between it and the corresponding features.

5.3. Support Vector Regression

The results of the model on the above mentioned four data sets are summarized in Table 6:

Table 5: SVR Model Training Performance Metrics

Model	RMSE	R^2
Dataset without 'MenuItem', (Model 1)		
Paytm+Cash	19.62	0.77
Coupons	34.43	0.86
Both	37.09	0.89
Dataset with 'MenuItem', (Model 2)		
Paytm+Cash	18.60	0.80
Coupons	32.85	0.87
Both	33.55	0.91
Dataset with 'Coupon Count', (Model 3)		
Paytm+Cash	21.15	0.65
Coupons	38.34	0.804
Both	53.81	0.72
Dataset with 'MenuItem' and 'Coupon Count', (Model 4)		
Paytm+Cash	18.56	0.73
Coupons	27.91	0.89
Both	33.2	0.9

The Support Vector Regressor (SVR) models yielded optimal results for the 'Coupons' and 'Both' scenarios when applied to the dataset incorporating both one-hot-encoded 'MenuItem' and 'Coupon Count.' However, the inclusion of 'Coupon Count' led to a decline in the predictive performance for the 'Paytm+Cash' scenario. This observation aligns with the logical expectation that 'Paytm+Cash' may not be significantly influenced by variations in 'Coupon Count.'

Table 6: SVR Model Testing Performance Metrics

Model	RMSE	R^2
Dataset without ’MenuItem’, (Model 1)		
Paytm+Cash	17.67	0.79
Coupons	37.23	0.84
Both	41.23	0.87
Dataset with ’MenuItem’, (Model 2)		
Paytm+Cash	18.10	0.78
Coupons	35.88	0.86
Both	38	0.89
Dataset with ’Coupon Count’, (Model 3)		
Paytm+Cash	22	0.53
Coupons	45.7	0.71
Both	57.55	0.59
Dataset with ’MenuItem’ and ’Coupon Count’, (Model 4)		
Paytm+Cash	18.01	0.68
Coupons	29.83	0.87
Both	32.31	0.87

Table 7: Neural Network Training Performance Metrics

Model	RMSE	R^2
Dataset without ’MenuItem’, (Model 1)		
Paytm+Cash	15.1	0.85
Coupons	31.6	0.87
Both	31.9	0.91
Dataset with ’MenuItem’, (Model 2)		
Paytm+Cash	15.37	0.85
Coupons	32.0	0.87
Both	31.2	0.92
Dataset with ’Coupon Count’, (Model 3)		
Paytm+Cash	10.2	0.90
Coupons	23.7	0.92
Both	27.1	0.92
Dataset with ’MenuItem’ and ’Coupon Count’, (Model 4)		
Paytm+Cash	10.1	0.91
Coupons	23.0	0.93
Both	26.2	0.92

Table 8: Neural Network Testing Performance Metrics

Model	RMSE	R^2
Dataset without ’MenuItem’, (Model 1)		
Paytm+Cash	19	0.76
Coupons	38.8	0.83
Both	42.4	0.86
Dataset with ’MenuItem’, (Model 2)		
Paytm+Cash	20.50	0.72
Coupons	37.9	0.84
Both	43.9	0.85
Dataset with ’Coupon Count’, (Model 3)		
Paytm+Cash	15.25	0.77
Coupons	36.5	0.80
Both	42.4	0.78
Dataset with ’MenuItem’ and ’Coupon Count’, (Model 4)		
Paytm+Cash	15.76	0.76
Coupons	39.5	0.78
Both	45.4	0.75

Interestingly, the root mean square error (RMSE) substantially increased when solely considering ’Coupon Count,’ suggesting that the predictive power of this feature is enhanced when combined with ’MenuItem.’ This underscores the complementary nature of ’Coupon Count’ and ’MenuItem’ in providing more comprehensive information for predicting the target variables, ’Coupon Count’ and ’MenuItem.’

Support Vector Regressor (SVR) outperforming linear regression and random forest could be attributed to its inherent capacity to handle non-linear relationships and capture complex patterns in the data. Unlike linear regression, which assumes a linear relationship between the features and the target variable, SVR can model non-linear dependencies through the use of kernel functions. This flexibility is particularly advantageous when dealing with intricate and non-linear patterns that may exist in real-world datasets.

Additionally, SVR is less prone to overfitting compared to random forest, especially when dealing with limited data. Random forest can sometimes overfit noisy data, whereas SVR, with appropriate parameter tuning, tends to generalize well to unseen data. The regularization properties of SVR help prevent it from fitting the training data too closely, making it more robust when applied to new and unseen samples.

5.4. Neural Networks

The results of the model on the above-mentioned four data sets with neural networks are summarized in Table 2:

The Neural Network model generally performs better than the Support Vector Regressor when ’Coupon Counts’ are considered. However, it fails to predict the targets well when using the Menu Items. We believe that this is due to the high dimensionality of this data. Neural networks are known to overfit on data that has high cardinality but less number of samples.

We use dropout layers along with L2 Regularization to reduce overfitting and largely succeed in doing so – reducing the discrepancy between train and test RMSE values from around (5, 20) in sklearn’s MLPRegressor to around (10, 15) using tf.keras’ Sequential networks. This proves the utility of such methods when using training data with a small number of samples.

6. Conclusion

6.1. Learnings from the Project

In the course of this project, we gained several valuable insights. Firstly, we learned about the importance of feature engineering in machine learning, something that can make or break certain models. Our experience also highlighted the significance of exploring different ways of encoding categorical variables, particularly through one-hot encoding, to improve model performance in the case of Linear Regression and Support Vector Regression. Additionally, the choice of the machine learning model was also important, with Random Forest Regression outperforming Support Vector Regression for the Paytm+Cash target variable and the latter performing better for the Coupons target variable. Other models like Linear Regression and Multi Layer Perceptron occupied a middle ground between the previous two in terms of metrics. In the case of Neural Networks, we were able to demonstrate the utility of techniques like L2 regularization and Dropout layers used to reduce overfitting. We also recognized the value of domain knowledge in feature engineering, especially when considering factors like academic calendars, holidays, menu items, and number of people buying coupons. Continuous data refinement and the balance between model interpretability and accuracy were key considerations.

6.2. Member Contribution

1. **Aditya Girdhar:** Data collection, Feature engineering, Linear Regression model, Result and Analysis, Abstract and Introduction.
2. **Aditya Raj:** Data collection, Feature engineering, Random Forest model, Neural Network model, Methodology, Result and Analysis and Conclusion.
3. **Mehar Khurana:** Feature engineering('MenuItem'), Random Forest model, Neural Network model, description and visualization.
4. **Nalish Jain:** Data collection, Feature engineering, Linear Regression model, Support Vector Regressor, Coupon Data collection and processing, Literature Review and methodology.

Overall all members contributed equally in the project.

7. Timeline

We were able to follow our proposed timeline.

References

1. https //www.sciencedirect.com/science/article/pii/S1877050910005004
2. https //medium.com/analytics-vidhya/restaurant-sales-prediction-using-machine-learning-24928a2e3206