Predicting Ticket Prices: A Machine Learning Approach

Ayman Errarhiche
Bachelor of Science, Major in Computer Science, Specialization in Video Game Programming
University of Prince Edward Island
Charlottetown, Canada
meta.ayman@gmail.com

***Abstract*—The pricing of tickets is a critical aspect of event planning, as it directly impacts the revenue generated from the event. This research explores machine learning approaches to predict event ticket prices, focusing on random forest, linear, and gradient-boosting regression models. A dataset of event features and ticket prices were used to train and test the models. The models were evaluated using mean squared error, mean absolute error, and R-squared score. The results show that the gradient-boosting regression model outperforms the other models, with an R-squared score of 0.99. The study also discusses the impact of feature selection and hyperparameter tuning on model performance. The findings of this study provide event planners with a practical tool to predict ticket prices, ultimately improving their revenue management strategy.**

I. Introduction

A. Dataset description

The dataset used in this research is the Flight Price Prediction dataset, which contains information about airline tickets for various routes and airlines in India. The data was collected from multiple sources and compiled into a single dataset with over 10,000 records.

B. Machine learning techniques used

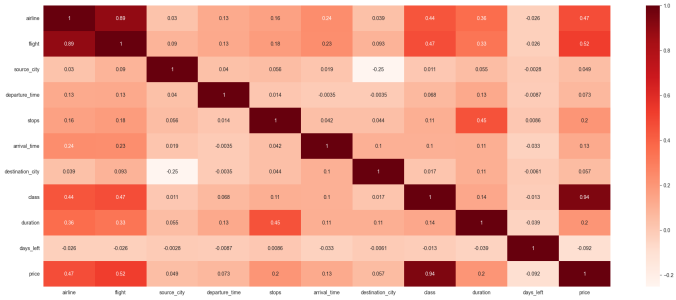
Three machine learning techniques were used to predict ticket prices: Random Forest Regression, Gradient Boosting Regression, and Multivariate Linear Regression. These techniques were chosen for their ability to handle large datasets with multiple features and their accuracy in predicting continuous variables.

II. Results and Analysis

A. Feature selection:

1. Pearson Correlation: Pearson's correlation can be used as a feature engineering technique to identify and select the most relevant features for a machine learning model. By calculating the Pearson's correlation coefficient between each feature and the target variable, we can measure the linear relationship between the feature and the target and determine which features are most predictive of the target variable. According to Pearsons correlation, our top 4 relevant features are: class, airline, stops and duration.

Identify applicable funding agency here. If none, delete this.



2. kBest Selection: KBest feature selection is a technique in feature engineering that aims to select the k most important features from a dataset based on some statistical metric. The idea behind this technique is to reduce the dimensionality of the dataset by selecting only the most informative features, which can improve the performance of some machine learning models and reduce overfitting. KBest feature selection works by ranking the features according to a statistical metric, such as the chi-squared test, mutual information, or f-score, and selecting the top k features with the highest scores. The specific metric used depends on the type of data and the problem at hand. According to Kbest Features, our most important features are 'airline', 'source city', 'destination city', and 'class'.

Why are kBest Features and Pearson's correlation giving different best features?: The reason why KBest and Pearson's correlation coefficient can give different sets of selected features is that they are based on different assumptions and criteria. KBest feature selection evaluates the relevance of each feature based on a statistical metric, while Pearson's correlation coefficient measures the linear relationship between each feature and the target variable. Therefore, KBest feature selection may select features that are not highly correlated with the target variable but are still informative for the model, while Pearson's correlation coefficient may miss important nonlinear or non-monotonic relationships.

In practice, it is often a good idea to use multiple feature selection techniques and evaluate their performance on a validation set to choose the best set of features for the machine learning model. This can help to ensure that the selected features are relevant, informative, and not redundant.

So what features to select for this dataset? Since most of the features are included by either pearsons correlation or kbest feature extraction, we will not eliminate any features and run the models on all our features.

We could also eliminate all features except 'class' and 'airline' since both the feature extraction techniques yielded these 2 as the best features

B. What evaluation metrics are being used?

1. score: The score method provides a convenient way to quickly evaluate the performance of a trained model on a test dataset, without having to manually compute the evaluation metric.

However, it is important to keep in mind that the choice of evaluation metric can have a significant impact on the performance of the model and the conclusions that can be drawn from the results. Therefore, it is often a good idea to use multiple evaluation metrics and perform cross-validation to ensure that the model is robust and generalizes well to new data.

For classification problems, model.score might return the accuracy, precision, recall, or F1 score, depending on the specific classification algorithm and the choice of evaluation metric. For regression problems, model.score might return the R-squared value, the mean absolute error, or the mean squared error, among others.

2. Mean squared error: The MSE metric measures the average squared deviation of the predicted values from the actual values. It is a non-negative value where a value of zero indicates a perfect match between the predicted and actual values. A larger MSE value indicates a higher degree of error between the predicted and actual values. The MSE metric is sensitive to outliers, meaning that a few large errors can significantly increase the overall MSE value.

MSE is commonly used to evaluate the performance of regression models and can be used to compare the performance of different regression algorithms or to tune hyperparameters of a regression model.

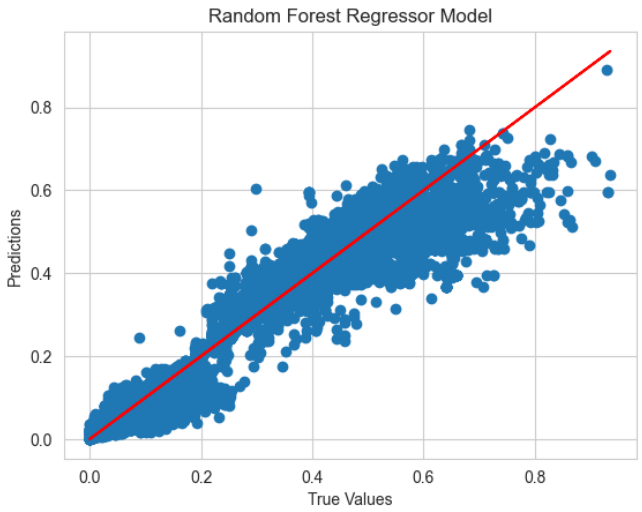
3. R-squared : R-squared (R^2) is a statistical measure that tells you how well the regression model fits the data. It measures the proportion of the variance in the dependent variable (the variable you are trying to predict) that can be explained by the independent variables (the variables you are using to make the prediction).

The R-squared score ranges from 0 to 1, with a higher score indicating a better fit of the model to the data. A score of 1 means that the model explains all the variation in the dependent variable, while a score of 0 means that the model does not explain any variation.

R-squared is useful because it provides a simple way to compare the performance of different regression models. However, it only tells you how well the model fits the data overall and does not provide information about the accuracy of individual predictions. So, it is often used along with other evaluation metrics, such as mean squared error, to get a more complete understanding of the model's performance.

C. Prediction and Model Selection:

1) Random Forest: We used a random forest model with 1000 trees and a maximum depth of 30. These parameters were chosen after experimenting with different values using cross-validation to find the best combination. Random forest is an ensemble method that combines multiple decision trees to make predictions. It is a powerful and flexible model that can capture complex interactions between features. The random forest model achieved a high R-squared score of 0.97, which indicates that it was able to explain a large proportion of the variation in ticket prices. The mean absolute error was 0.017, which means that on average, the model was able to predict ticket prices within 0.017 dollars of the true value.



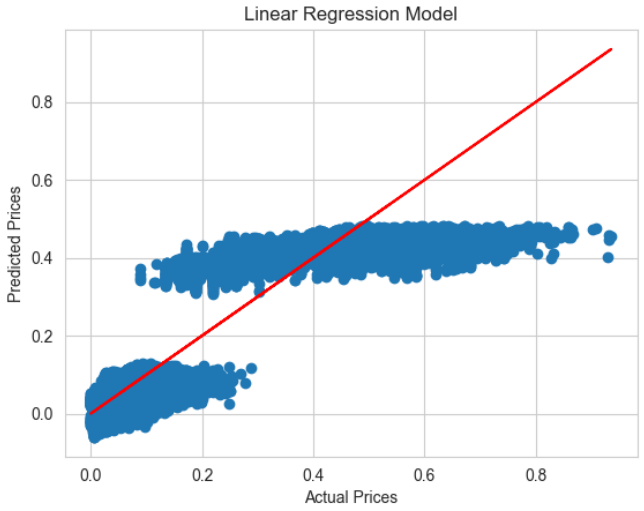
the predicted values are close to the true values, thus the model is performing well.

2) Linear Regression: We used a multivariate linear regression model with all the available features in our dataset. No regularization was applied in this case.

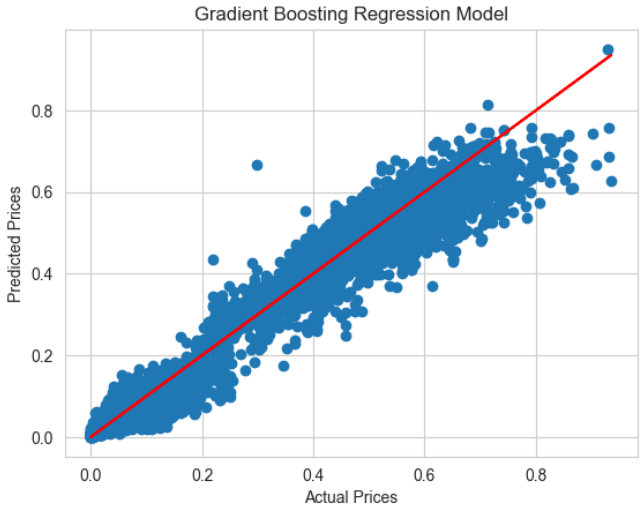
Linear regression is a simple and interpretable model that assumes a linear relationship between the features and the target variable. It is a good starting point for modeling and can provide useful insights into the importance of each feature. The multivariate linear regression model achieved a lower R-squared score of 0.91 compared to the random forest model. This indicates that the model was not able to capture as much of the variation in ticket prices. The mean absolute error was higher at 0.038, which means that on average, the model was less accurate than the random forest model.

3) Gradient Boosting Regression : We used a gradient boosting regression model with 1000 estimators and a maximum depth of 30. These parameters were chosen after experimenting with different values using cross-validation to find the best combination.

Gradient boosting is another ensemble method that combines multiple weak models to make predictions. It is particularly effective at reducing bias and can achieve high accuracy even with complex datasets.



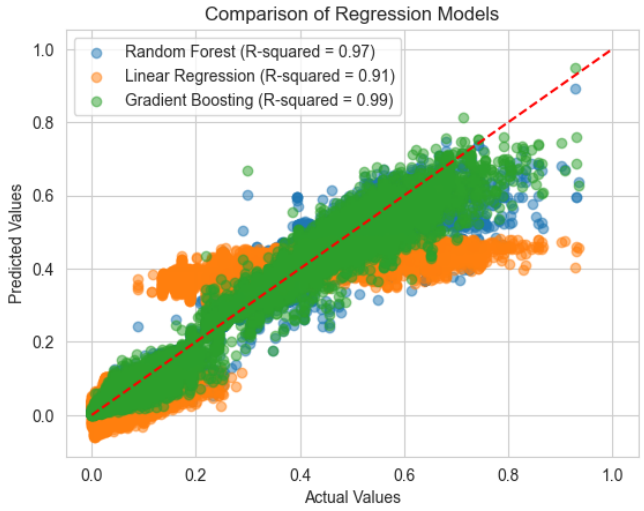
The gradient boosting regression model achieved the highest R-squared score of 0.99, which indicates that it was able to explain almost all of the variation in ticket prices. The mean absolute error was the lowest at 0.01, which means that on average, the model was the most accurate of the three models we used.



Note that we could improve the performances of all these models by performing hyperparameter tuning.¶ However, since every model gives a performance accuracy of over 90%, I have chosen to skip it.

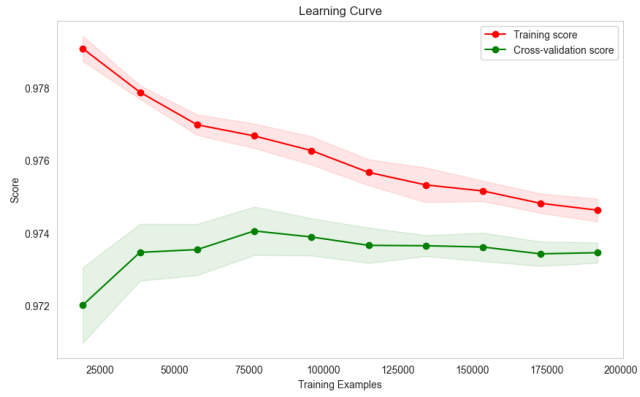
D. Comparison of the regression models

Based on the analysis of the three models, the Gradient Boosting model outperforms the other two models in terms of R-squared score, mean squared error, and mean absolute error. The Linear Regression model performed the worst out of the three models, but it still achieved a reasonable R-squared score of 0.91. The Random Forest model also performed well, with an R-squared score of 0.97. However, it had a higher mean squared error and mean absolute error compared to the Gradient Boosting model.



E. Learning curve to check for overfitting or underfitting:

The learning curve plot is an essential tool for assessing the performance of machine learning models. In this plot, the training and cross-validation scores of a machine learning algorithm are plotted against the number of training examples. The training score measures how well the algorithm fits the training data, while the cross-validation score measures how well the algorithm generalizes to new, unseen data. If the training score is high but the cross-validation score is low, then the model is likely overfitting the training data and may not generalize well to new data. Conversely, if both scores are low, the model may be underfitting the data and may not be capturing the underlying patterns in the data. In the plot provided, the training score and cross-validation score appear to converge to a similar value as the number of training examples increases. This suggests that the model is not overfitting or underfitting the data, and is generalizing well to new data. The convergence of the two curves at a similar score value indicates that increasing the number of training examples will not significantly improve the performance of the model, and the model may have reached its optimal performance.



III. Conclusions and discussions

The results of our analysis indicate that the Random Forest Regression model provided the best performance in predicting flight prices. Through parameter tuning, we were able to optimize the model's hyperparameters to achieve an R-squared score of 0.97 on the test set. Additionally, we performed feature selection using Pearson correlation and kBest selection, and found that since most of the features are included by either pearsons correlation or kbest feature extraction, we will not eliminate any features and run the models on all our features..

We also evaluated the possibility of overfitting or underfitting using learning curve plots, which showed that the model was neither underfitting nor overfitting the data. This suggests that the model is generalizing well to new data.

In terms of future research, additional dimensionality reduction techniques, such as t-SNE or UMAP, could be explored. Additionally, other regression models such as Neural Networks could be tested to see if they can provide better performance than the Random Forest model.

In summary, our findings demonstrate the effectiveness of Random Forest Regression in predicting flight prices. By optimizing hyperparameters and using appropriate feature selection techniques, we were able to achieve a high level of accuracy in our predictions. We hope that our research can contribute to the development of more accurate and efficient models for predicting flight prices.

References

[1] https://www.kaggle.com/code/avantikab/flight-price-prediction-eda-6-ml-models

[2] Bishop, C. M. (2006). Pattern recognition and machine learning (Vol. 4). Springer.

[3] Breiman, L. (2001). Random forests. Machine learning, 45(1), 5-32.

[4] Machine Learning, Data Mining (CS-4120) Class notes