Project report

Abstract: This research focuses on improving passenger satisfaction in the airline industry. Using data analytics and machine learning, we can gain insights and boost passenger experiences. This project evaluates several machine learning models (KNN, Naive Bayes, Decision Trees, SVM, Random Forest, AdaBoost, and XGBoost) to predict and improve passenger satisfaction. Extensive testing and analysis will provide airlines with valuable information for enhancing their services and making customer satisfaction a top priority.

Introduction: In the highly competitive airline industry, providing excellent customer experiences is crucial for airlines. Happy passengers are more loyal, which improves brand image and leads to greater revenue. Airlines have traditionally relied on surveys and passenger feedback, but machine learning advancements provide innovative ways to enhance satisfaction. This research explores how machine learning can be utilized to analyze passenger data and improve overall contentment.
In the last work i mention before in the proposal he used some models like KNN,support vector machine,Random Forest, AdaBoost and gradient boosted decision trees, he depend on the error percentage in training and test only but i try to make the time taken by the model under consideration
But finally both of us found that Random Forest showed the best result.

Methods: My approach consists of several steps:Data Preparation: I clean and format the data to make it compatible with our machine learning models. I use techniques to extract meaningful features from the data. Model Selection and Training: I evaluate different machine learning algorithms to find the ones that work best for predicting passenger satisfaction. I train the models using our data, adjusting their parameters to improve their performance. I test the models to measure their accuracy and make sure they generalize well.

What machine learning techniques have I tried and why?

1. **K Nearest Neighbors (KNN)**
2. **Naive Bayes Classifier**
3. **Decision Tree Classifier**
4. **Support Vector Machine (SVM)**
5. **Random Forest**
6. **AdaBoost**
7. **XGBoost**

Each machine learning model excels at handling specific types of data and prediction challenges. By testing out various models, I can assess their strengths and limitations. This helps me determine the best model for predicting airline passenger satisfaction based on the specific data you have available.

Experiments: During our testing phase, we thoroughly assess chosen machine learning models i preferred before using the prepared data. I evaluate each model's performance using metrics like accuracy, precision, recall, and F1 score. and also analyze errors to find common problems and areas for improvement. For reference, we include traditional statistical models or heuristics. By systematically testing and refining, we gain insights into the strengths and weaknesses of each model and their abilities in forecasting passenger satisfaction.

Results

Let's provide a more detailed explanation of what happened with each model during modeling

In the modeling stage, each machine learning model underwent training and evaluation processes using the prepared data. Here's a summary of the specific actions taken during this phase:

K Nearest Neighbors (KNN):

The KNN model was trained and tested on the dataset. Here are the key findings:

Training Set Performance:

- Accuracy: The KNN model achieved an accuracy of approximately 92.2% on the training set, indicating good performance in learning from the data.

Testing Set Performance:

- Accuracy: On the testing set, the KNN model achieved an accuracy of 92.29%, suggesting its ability to generalize well to unseen data.
- ROC Area under Curve (ROC AUC): The model's ROC AUC score was approximately 0.9149, indicating its effectiveness in distinguishing between satisfied and unsatisfied passengers.
- Precision, Recall, and F1-score: Detailed metrics showed balanced performance across precision, recall, and F1-score for both classes.
- Execution Time: The KNN model took approximately 210.12 seconds to train and evaluate.

Naive Bayes (NB):

The NB classifier was trained and tested on the dataset. Here are the key findings:

Training Set Performance:

- Accuracy: The NB classifier achieved an accuracy of approximately 86.5% on the training set.

Testing Set Performance:

- Accuracy: On the testing set, the NB classifier achieved an accuracy of 92.7%.
- ROC Area under Curve (ROC AUC): The model's ROC AUC score was approximately 0.859.
- Precision, Recall, and F1-score: Detailed metrics showed balanced performance across precision, recall, and F1-score for both classes.
- Execution Time: The NB classifier took approximately 0.157 seconds to train and evaluate.

Decision Tree (DT):

The DT classifier was trained and evaluated on the dataset. Here are the key findings:

Training Set Performance:

- Accuracy: The DT classifier achieved an accuracy of approximately 94.6% on the training set.

Testing Set Performance:

- Accuracy: On the testing set, the DT classifier achieved an accuracy of 94.6%.
- ROC Area under Curve (ROC AUC): The model's ROC AUC score was approximately 0.943.
- Precision, Recall, and F1-score: Detailed metrics showed balanced performance across precision, recall, and F1-score for both classes.
- Execution Time: The DT classifier took approximately 1.055 seconds to train and evaluate.

Support Vector Machine (SVM):

The SVM classifier was trained and evaluated on the dataset. Here are the key findings:

Training Set Performance:

- Accuracy: The SVM classifier achieved an accuracy of approximately 86.5% on the training set.

Testing Set Performance:

- Accuracy: On the testing set, the SVM classifier achieved an accuracy of 95.6%.
- ROC Area under Curve (ROC AUC): The model's ROC AUC score was approximately 0.953.
- Precision, Recall, and F1-score: Detailed metrics showed balanced performance across precision, recall, and F1-score for both classes.
- Execution Time: The SVM classifier took approximately 392.71 seconds to train and evaluate.

Random Forest (RF):

The RF classifier was trained and evaluated on the dataset. Here are the key findings:

Training Set Performance:

- Accuracy: The RF classifier achieved an accuracy of approximately 96.2% on the training set.

Testing Set Performance:

- Accuracy: On the testing set, the RF classifier achieved an accuracy of 96.2%.
- ROC Area under Curve (ROC AUC): The model's ROC AUC score was approximately 0.959.
- Precision, Recall, and F1-score: Detailed metrics showed balanced performance across precision, recall, and F1-score for both classes.
- Execution Time: The RF classifier took approximately 28.38 seconds to train and evaluate.

AdaBoost (Adaptive Boosting):

The AdaBoost classifier was trained and evaluated on the dataset. Here are the key findings:

Training Set Performance:

- Accuracy: The AdaBoost classifier achieved an accuracy of approximately 92.8% on the training set.

Testing Set Performance:

- Accuracy: On the testing set, the AdaBoost classifier achieved an accuracy of 92.8%.
- ROC Area under Curve (ROC AUC): The model's ROC AUC score was approximately 0.926.
- Precision, Recall, and F1-score: Detailed metrics showed balanced performance across precision, recall, and F1-score for both classes.
- Execution Time: The AdaBoost classifier took approximately 20.40 seconds to train and evaluate.

XGBoost (Extreme Gradient Boosting):

The XGBoost classifier was trained and evaluated on the dataset. Here are the key findings:

Training Set Performance:

- Accuracy: The XGBoost classifier achieved an accuracy of approximately 96.1% on the training set.

Testing Set Performance:

- Accuracy: On the testing set, the XGBoost classifier achieved an accuracy of 96.1%.
- ROC Area under Curve (ROC AUC): The model's ROC AUC score was approximately 0.959.
- Precision, Recall, and F1-score: Detailed metrics showed balanced performance across precision, recall, and F1-score for both classes.
- Execution Time: The XGBoost classifier took approximately 174.37 seconds to train and evaluate.

Conclusion

We observe , Random Forest and XGBoost have performed equally well on producing high ROC_AUC scores (99%). But **Random Forest** has taken a lesser amount of time compared to time taken by AdaBoost. So, we will stick to Random Forest as the best model.

References:

Data:https://www.kaggle.com/datasets/teejmahal20/airline-passenger-satisfaction/data

Previous work: https://www.kaggle.com/code/frixinglife/airline-passenger-satisfaction-part-1

https://www.kaggle.com/code/frixinglife/airline-passenger-satisfaction-part-2

Paper:https://www.researchgate.net/publication/350552031_Predicting_Airline_Passenger_Satisfaction_with_Classification_Algorithms