Project Title: Flight delay prediction based on machine learning

Problem Statement

Nearly every airline passenger has experienced the uncertainty and stress associated flight delays. On a regular basis, the national media highlights the growing trend of flight delays and the impact these delays pose for the travelling public. In 2007, U.S government had endured 31–40 billion dollars downsides due to flight delays. In 2017, 76% of the flights arrived on time. Where, in comparison to 2016, the percentage of on time flights decreased by 8.5% [1]. Almost half of all flight delays and cancellations are caused by poor weather. While weather may only minimally impact other modes of transportation, it is the single most prevalent cause of flight delays. KnowDelay™, it is a technology-based service that accurately predicts weather-related delays up to five days in advance of travel and identifies alternative itineraries that are less likely to be delayed. Company research has shown that the likelihood of a passenger delay during poor weather can be reduced from over 50 percent to approximately 15 percent with the use of KnowDelay. Over the past few decades, Machine Learning (ML) delivers an accurate and quick predict outcome and it has become a powerful tool in health settings, offering personalized clinical care for stroke patients. Machine learning methods are flexible prediction algorithms with potential advantages over conventional regression and scoring system. XGBoost has been widely recognized in the number of machine learning and data mining challenges. In this study, we will explore flight delay prediction dataset using XGBoost and compare with various other machine learning models and techniques to predict status of flight delay.

Data Source

The data we are using is from KnowDelay. It contained 34 airports records. Each row in the data provides relevant information about the flight arrive and delay information. The data types are mixed and include strings, integer, and floats. Each row represents an hour record. The dependent variables are Temp, Dew, T.D.Spread, Wind.Direction, Wind speed, Wind Gust, Altimeter, Vis, Ceriling and Avg.Gate.Arrival.Delay. Using ATL airport as an example, the dataset contains data associated with 8760 hours information. The mean of Temp is 66.29 years, on time gate arrival percentage is 79.66%. Average gate arrival delay is 10.47 minutes. The delay ratio in this dataset is 3.4%. The brief data summary is in the Table 1. Figure 1 show the delay distribution by months. It clearly shown that there is not significant different across the months. The other data summary graphs and tables were in Appendix. We observed that delay is similar across hours each day except the early hour (5 am – 8 am) due to there is not many flights during that time each day in the Appendix Figure 2.

Table 1: Summary statistics of the study airport (ATL)

Features	**Statistics (**mean ± std**)**
Temp	66.2900 ± 17.87
Dew	52.1900 ± 16.65
T.D.Spread	14.1000 ± 9.2
Wind.Direction	167.3300 ± 99.47
Wind.Speed	9.2000 ± 5.07

Wind.Gust	23.4500 ± 4.9
Altimeter	29.9800 ± 0.74
Vis	9.4600 ± 1.82
Ceiling	13655.6400 ± 11785.23
On.Time.Gate.Arrivals	79.6600 ± 22.98
Avg.Gate.Arrival.Delay	10.4700 ± 17.64

Note: ATL mean ATL airport.

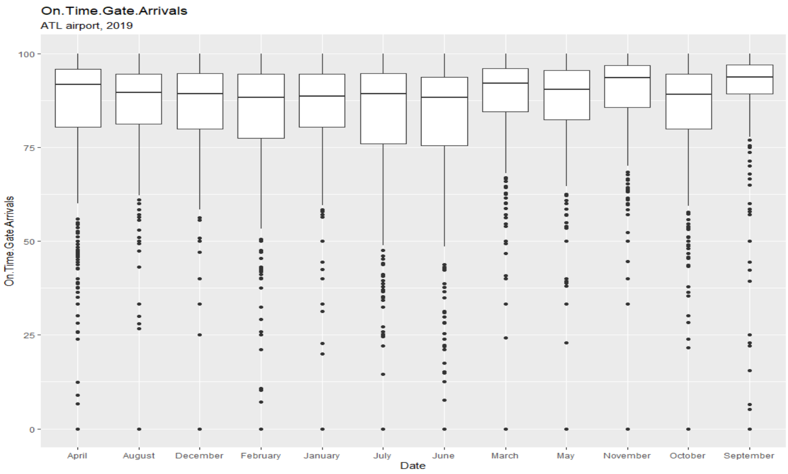

Figure 1. The flight on time gate arrival percentage distribution by month (ATL).

Methodology

EXtreme Gradient Boosting (XGBoost) is a machine learning technique with the remarkable features of processing the missing data efficiently and flexibly and assembling weak prediction models to build an accurate one [2]. As an open source package, XGBoost has been widely recognized in a number of machine learning and data mining challenges, for example, 17 solutions used XGBoost among the 29 challenge winning solutions published at Kaggle's blog in 2015 and the top-10 winning teams used XGBoost in KDD Cup 2015. novel machine learning techniques have demonstrated improved predictive performance compared to traditional prediction methods. In this project. We attempted to compare the performance of machine learning (XGboost) model with several others traditional prediction models (Random Forest, Logistic Regression, GradientBoost, AdaBoost, GaussianNB, KNN, MLP) [3,4,5]. Methodology flowchart of this project is shown in Figure 2, and the set-ups of all these methods will be introduced in detail in their respective parts.

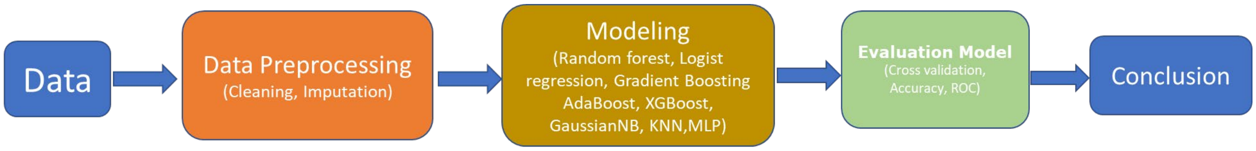
Data
Data Preprocessing
(Cleaning, Imputation)
Modeling
(Random forest, Logist regression, Gradient Boosting AdaBoost, XGBoost, GaussianNB, KNN,MLP)
Evaluation Model
(Cross validation, Accuracy, ROC)
Conclusion

Figure 2: Methodology Flowchart

Data Preprocessing

Missing value and Data Imputation

Based on the Figure 3. We have found that there are 32 missing values in the Temp, Dew, T.D.Spead, Wind.Speed, and another 7924 unknown values for Wind gust. The missing value were filled based on the following rule. Wind Gust use 0 if missing, visibility use 10 if blank, Ceiling use 10000 if blank. The records will be records if it missing all the feature information. Total missing rate is 8.6% in this dataset. It is mainly due to the wind gust. Based on the Figure 3. We can see that the missing variables did not show clear association with other variables and therefore we can assume this missingness is MCAR (missing completely at random). The correlation map as shown in Figure 4.

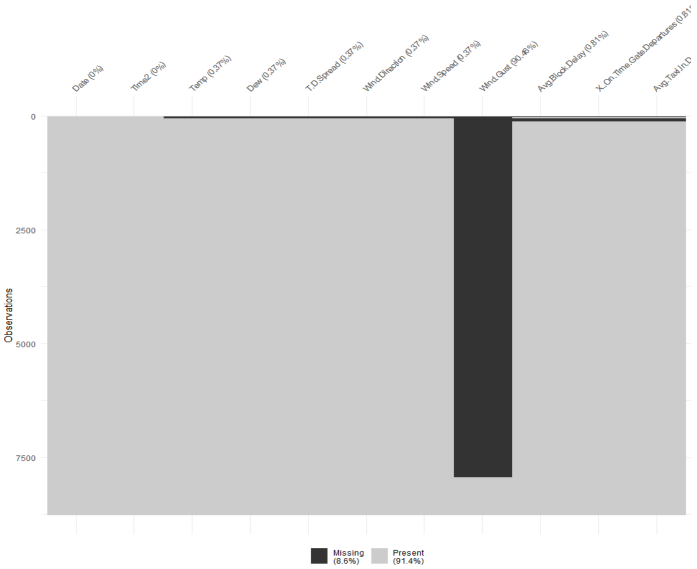

Figure 3. Missing rate in the features (ATL).

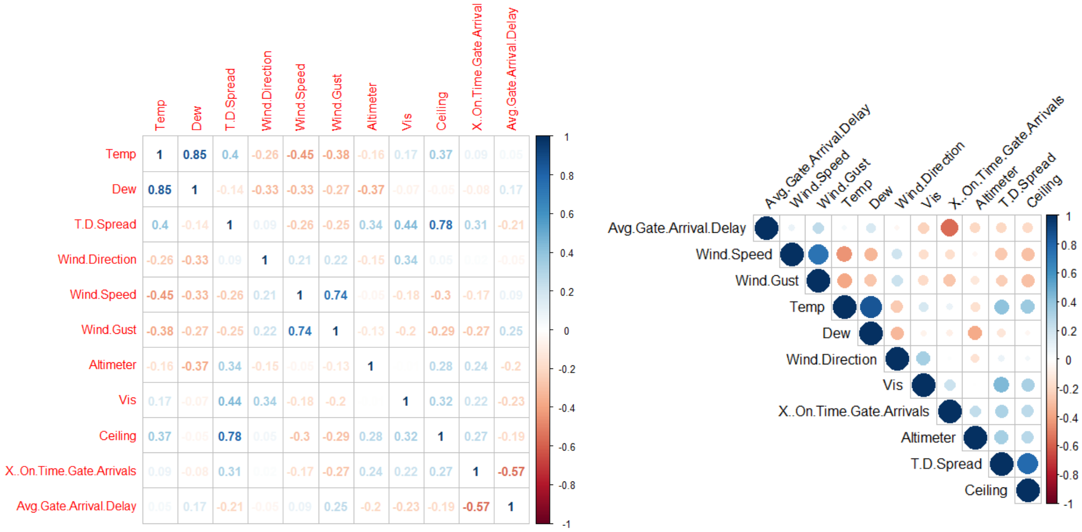

Figure 4. The correlation between the features (ATL).

Experimental Setup

Our study contained three stages of implementation: (1) Data summary and processing (**ETL process**) on Local Docker environment (1 terabyte space, 16 GB RAM, and 8 processors, 6 GB GPU). (2) Modeling and Cross Validation (**Modeling**) on a local cluster (1 terabyte space, 16 GB RAM, and 8 processors, 6 GB GPUs). (3) Prediction using several machine learning methods on a local cluster (1 terabyte space, 16 GB RAM, and 8 processors, 6 GB GPUs). Data output in the first stage is then used as the input for the model training in the second stage. We also used Python and packages such as Pandas and Scikit-learn for model testing, hyperparameter tuning and model evaluation. Using ATL airport as an example for the initial analysis. And then we are using all other airports data for the evaluation.

Evaluation

The dataset will be split into 70% training set and 30% test set. Hyperparameter tuning was done on 5-fold CV of the training set and the final evaluation of model performance was done on the test set. According to the highly imbalance of this dataset in Figure 5, A problem with imbalanced classification is that there are too few examples of the minority class for a model to effectively learn the decision boundary. Synthetic Minority Oversampling Technique (SMOTE) is a very popular oversampling method that was proposed to improve random oversampling with low-dimensional data. So, we will apply SMOTE (Synthetic Minority Oversampling Technique) to oversample the dataset and then the models can learn more efficiently [6].

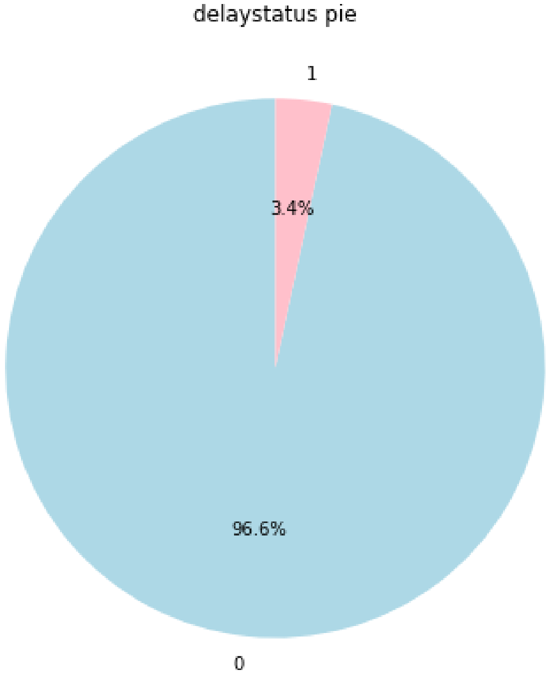

Figure 5. The flight delay distribution (ATL).

Multiple machine learning models (Random Forest, Logistic Regression, GradientBoost, AdaBoost, XGBoost, GaussianNB, KNN, MLP) in the study were evaluated and compared using the Area under the Receiver Operating Characteristic curve (ROC) on the test set. The ROC curve is the true positive rate against the false positive rate at various threshold settings. ROC provides a single measure of the diagnostic ability of a binary classifier. Apart from the primary metric of Roc_auc_score, accuracy, MAPE (mean absolute percentage error), precision, recall and F1-score were also used during the model testing stage to provide a full picture of model performance.

Results

The results are based on the 5-fold cross validation. For phase1 model comparison results, we are using 9 features (Temp, Dew, T.D.Spread, Wind.Direction, Wind speed, Wind Gust, Altimeter, Vis, Ceriling) to predict the flight delay. Eight predictive models including Random Forest, Logistic Regression, GradientBoost, AdaBoost, XGBoost, GaussianNB, KNN, MLP and XGBoost algorithm model were constructed by Python software. Based on the ROC curve graph in Figure 6, we found that GaussianNB is the worst method to predict the flight delay in this dataset. Random Forest and XGBoost are performing best compared with the rest of methods. Based on the Accuracy and MAPE in the Table 2. XGBoost method has best accuracy 0.9825 and lowest MAPE value 0.0107. The random forest had a resulting MAPE of 0.0230 and an accuracy score of 0.9689. The precision, recall and F1-score are shown in the Appendix table 1.

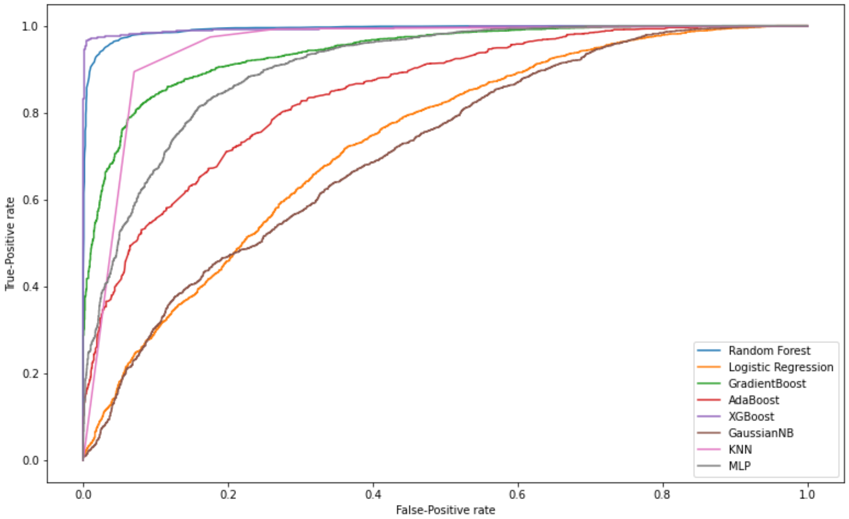

Figure 6. ROC trend by eight machine learning methods (ATL) using 9 features.

Table 2: Accuracy and RMSE for the models using 9 features (ATL).

Methods	Accuracy	Roc_auc_score	MAPE

Random Forest	0.9689	0.9956	0.0230
Logistic Regression	0.6787	0.7397	0.2493
GradientBoost	0.8690	0.9416	0.1009
AdaBoost	0.7632	0.8500	0.1866
XGBoost	0.9825	0.9944	0.0107
GaussianNB	0.6426	0.7178	0.2558
KNN	0.9164	0.9577	0.0793
MLP	0.8509	0.9226	0.1233

For phase2 model comparison results, Table 3 and Figure 7 compares the model performance of 8 models using 4 features (Wind Speed, Wind Gust, Visibility, Ceiling) to predict the flight delay. The results showed good discriminatory power with accuracy score of 0.7398, 0.6113, 0.7401, 0.6855, 0.7909, 0.5995, 0.7138, 0.6763, respectively (table 3). The XGBoost algorithm model showed the highest accuracy score and lowest MAPE value. Logistic Regression has lowest accuracy score and highest value for MAPE.

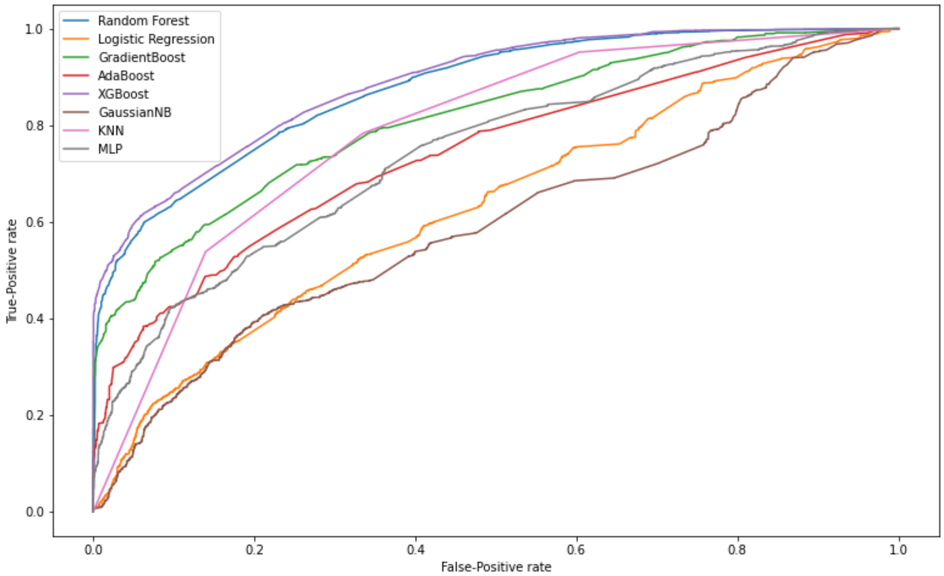

Figure 7. ROC trend by eight machine learning methods (ATL) using 4 features.

Table 3: Accuracy and RMSE for the models using 4 features (ATL).

Methods	Accuracy	Roc_auc_score	MAPE
Random Forest	0.7398	0.8786	0.2352
Logistic Regression	0.6113	0.6447	0.2380

GradientBoost	0.7401	0.8267	0.1683
AdaBoost	0.6855	0.7586	0.2122
XGBoost	0.7909	0.887	0.1545
GaussianNB	0.5995	0.6023	0.2374
KNN	0.7138	0.8038	0.2261
MLP	0.6763	0.7555	0.2361

All other airports data were used to evaluate the results using XGBoost model. The accuracy, Roc_auc_score, and MAPE were shown in table4. For the accuracy, PDX airport is the highest predictive accuracy with 0.9905 and lowest MAPE. EWR airport is the lowest with 0.9169 and the highest MAPE. The results have confirmed XGBoost model is the consistent and best model comparing with the other models.

Table 4: Accuracy and MAPE for the airport using XGBoost model

Airport	Accuracy	Roc_auc_score	MAPE
ATL	0.9825	0.9944	0.0107
BOS	0.9620	0.9910	0.0270
BWI	0.9768	0.9921	0.0141
DEN	0.9626	0.9916	0.0263
DFW	0.9637	0.9913	0.0227
DTW	0.9847	0.9964	0.0095
IAH	0.9694	0.9904	0.0181
JFK	0.9750	0.9923	0.0164
MSP	0.9772	0.9946	0.0145
PHL	0.9655	0.9921	0.0221
SEA	0.9886	0.9976	0.0076
SLC	0.9797	0.9934	0.0119
CLE	0.9655	0.9881	0.0216
CVG	0.9689	0.9882	0.0178
DCA	0.9804	0.9972	0.0130
MCI	0.9729	0.9910	0.0151
MDW	0.9779	0.9941	0.0137

PDX	0.9905	0.9977	0.0058
PIT	0.9708	0.9891	0.0169
RDU	0.9709	0.9903	0.0172
TPA	0.9642	0.9862	0.0213
CLT	0.9794	0.9941	0.0124
EWR	0.9169	0.9757	0.0585
FLL	0.9804	0.9928	0.0109
IAD	0.9525	0.9784	0.0288
LAS	0.9817	0.9962	0.0129
LAX	0.9855	0.9976	0.0089
LGA	0.9369	0.9808	0.0419
MCO	0.9798	0.9952	0.0124
MIA	0.9820	0.9917	0.0101
PHX	0.9740	0.9926	0.0160
SAN	0.9883	0.9971	0.0075
SFO	0.9498	0.9862	0.0348

Discussion

This study demonstrated that the use of machine learning models can accurately predict the flight delay. various machine learning algorithms have been investigated for prediction performance. The AUCs, accuracy score and MAPE we developed have demonstrated the benefit of using a XGboost model as opposed to the other machine learning for prediction of flight delay. Our results have showed that XGBoost has outperformed the other methods based on all these three criterions. But there are several limits in XGBoost methods. For instance, the features selected were according to experiences but not algorithm; the representativeness of features may not clear in flight delay and some important dynamic features were not included. We also compared the eight models using two different kinds of features. Using all 9 features are outperforming the model which is only using 4 features. Based on this result, we suggest using all 9 features to do the prediction. We also use all other airport data to evaluate the XGBoost models. XGBoost model give the consistent and the best predicted accuracy.

The strength of this study was mainly that it was to predict the flight delay using the XGBoost model. And compared to traditional regression analysis. We must acknowledge some other limitations of our study and it may provide a base for potential improvement: firstly, because the data come from only one database and incidence of flight delay is very small; secondly, further exploration for the database was not performed, which may lead to the abandonment of some key variables; Thirdly, the proposed model was not designed to be validated by developing set from the other database. Even so, we believe that the proposed model may contribute to further our understanding of the flight delay.

Conclusion

In this project, we investigate an interesting topic of predicting the flight delay by implementing eight machine learning models (Random Forest, Logistic Regression, GradientBoost, AdaBoost, GaussianNB, KNN, MLP, XGBoost). Based on the cross-validation results. XGboost generally performs the best on predicting all awards. Using machine learning technique by XGboost, more significant prediction model can be built. We also apply this technique on other airports data sets and get the similar accuracy. Therefore, by proposed model XGboost has greater accuracy in forecasting flight delay compared to other machine learning models.

Reference

1. Maryam FY, Seeyed RK. Flight delay prediction based on deep learning and Levenberg-Marquart algorithm, Jounrnal of Big Data, 2020;7:106.

2. Chen T, Guestrin C. XGBoost: a scalable tree boosting system. In: Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining-KDD 2016, San Francisco, CA, USA; 2016. p.785–94.

3. Breiman L. Random forests. Mach Learn. 2001;45(1):5–32.

4. Friedman JH. Greedy function approximation: a gradient boosting machine. Ann Stat. 2000;29(5):1189–232.

5. Tibshirani R. Regression shrinkage and selection via the lasso. J R Stat Soc Ser B Methodol. 1996;58(1):267–288.

6. Blagus, R., Lusa, L. SMOTE for high-dimensional class-imbalanced data. BMC Bioinformatics 14, 106 (2013). https://doi.org/10.1186/1471-2105-14-106

Appendix

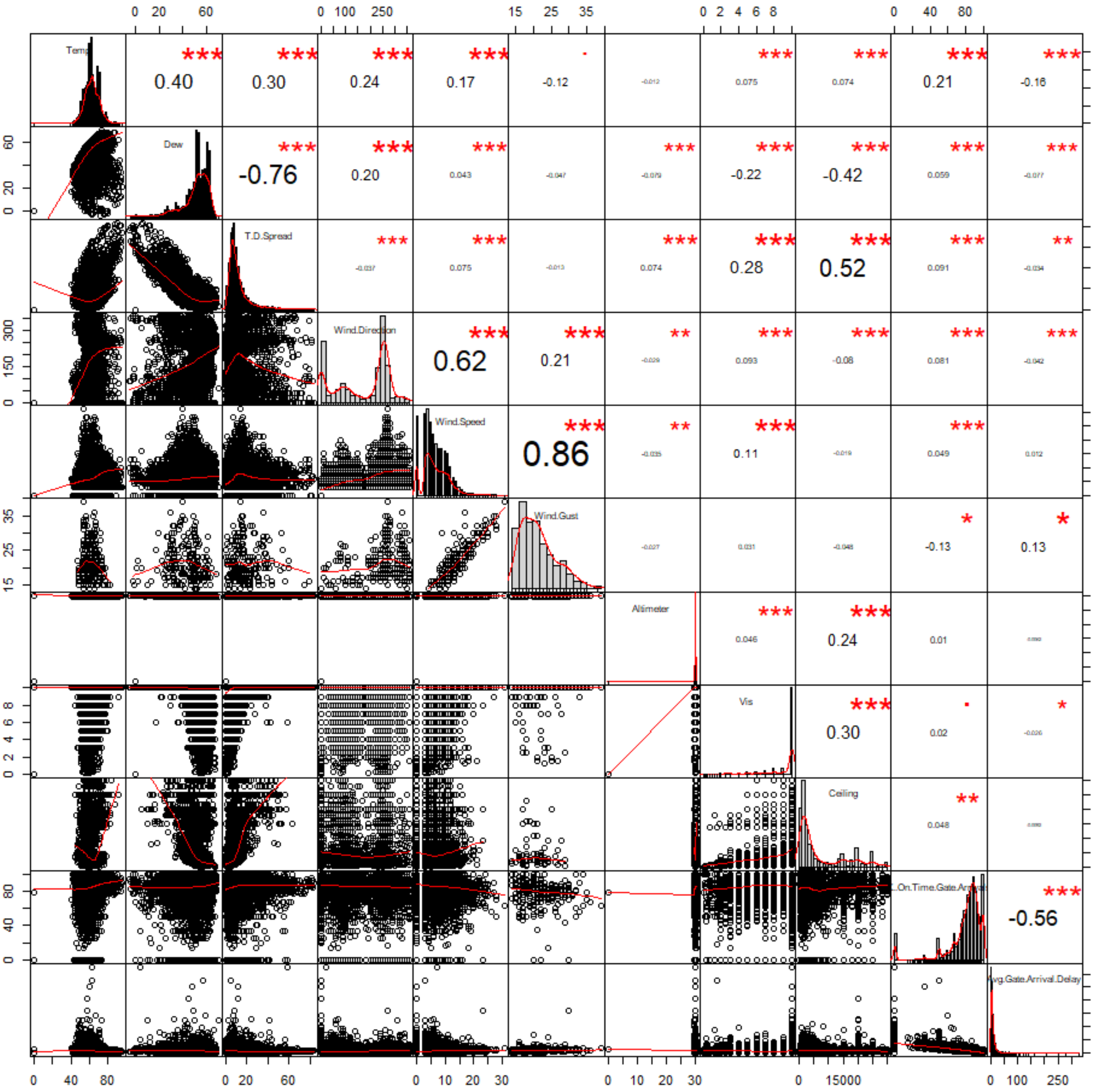

Figure 1. The KDE density graph between the features (Temp, Dew, T.D.Spread, Wind.Direction, Wind speed, Wind Gust, Altimeter, Vis, Ceriling, On.Time.Gate.Arrivals, Avg.Gate.Arrival.Delay).

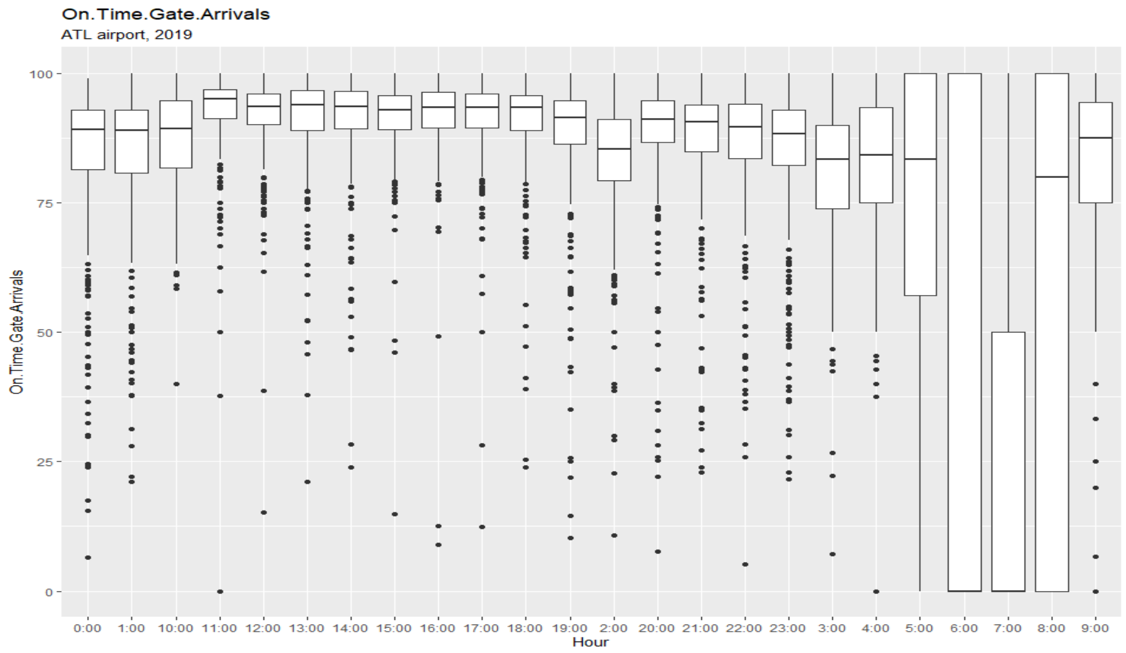

Figure 2. The flight on time gate arrival percentage distribution by Hour (ATL).

Table 1: The precision, recall and F1-score for the model (ATL)

Model		precision	recall	f1-score
Random Forest	0	0.96	0.96	0.96
	1	0.96	0.96	0.96
Logistic Regression	0	0.68	0.65	0.66
	1	0.66	0.7	0.68
GradientBoost	0	0.88	0.85	0.86
	1	0.86	0.88	0.87
AdaBoost	0	0.79	0.72	0.75
	1	0.74	0.81	0.77
XGBoost	0	0.97	0.99	0.98
	1	0.99	0.97	0.98
GaussianNB	0	0.62	0.7	0.66
	1	0.66	0.57	0.61
KNN	0	0.97	0.82	0.89
	1	0.85	0.97	0.91
MLP	0	0.85	0.79	0.82
	1	0.81	0.86	0.83