UDC 004.94

D. Tarasonok, Y. Oliinyk, T. Likhouzova

MODELS FOR FORECASTING FLIGHT DELAYS

Abstract: The problem of improving the operation of airports and air carriers is considered. It is proposed to use machine learning models and technologies to predict flight delays. Several different quality measures are used to evaluate the effectiveness of the proposed models, which diversely reflect the expediency of using these models in the context of the needs of each task.

Keywords: machine learning, forecasting, regression problem, classification problem.

Introduction

All his existence man tried to fly like a bird. The Wright brothers were able to lift a heavier-than-air device into the air for the first time at the end of 1903: it was a very primitive aircraft, and it was in the air for only 12 seconds.

Nowadays, humanity cannot imagine its existence without air travel: it includes cargo transportation, public transportation, and military use. According to statistics, more than 10,000 planes are constantly in the air. Such a large industry simply needs to analyze trends in detail to improve logistics and increase profits.

For instance, accurate forecasting of flight delays can provide valuable guidance to logistics companies for more precise planning of their transportation activities. In this industry, this is one of the key elements in generating profit.

Two tasks were set for predicting flight delays: quantitative forecasting (in minutes) of the departure delay and qualitative forecasting (whether the departure will be delayed or not). It is worth noting that it is the departure delay that will be predicted, not the arrival delay. To accomplish the tasks, we used Python 3 packages: scikit-learn [1], pandas [2], NumPy [3], Matplotlib [4], and Keras [5].

Data description

Flights are characterized by the following features: airport of departure and airport of destination, flight number (a specific route identifier), registration number (a unique number for each existing aircraft) of the aircraft operating the flight, scheduled and actual time of departure and arrival.

Delays can be affected by a large number of factors, one of which is the weather conditions at the time of departure. The weather report is transmitted in the aviation industry according to a special METAR standard (METeorological Aerodrome Report). This report includes a very large amount of information, which will be listed briefly: time this report was

 ISSN 1560-8956

taken, air temperature, dew point temperature, relative humidity, wind direction, wind speed, precipitation information, pressure, visibility, wind gust speed, cloud level, icing.

Since METAR data is not actually available in advance, you can either use the data that is valid at the time of the delay prediction or use TAF [10] (Terminal Aerodrome Forecasts), which is a weather forecast for the airport. It is valid for 30 hours and is issued 4 times a day at 6-hour intervals; it includes wind direction and speed, visibility, precipitation, and cloud cover.

When it comes to aircraft, model information is characterized by the following features: manufacturer, model name, model code, aircraft type, engine type, number of engines, and its turbulence category. One model code can include a large number of different modifications of this model. Currently, aircraft are manufactured with the following types of engines: electric, jet, piston, rocket, and turboprop (or turbocharged). The turbulence category is a category that actually indicates the weight and size parameters of an aircraft. It can have the following values: L (light), L/M (light-mid), M (medium), H (heavy), and J (Airbus A380 only).

It is also worth defining what exactly is considered a departure delay. It can be measured quantitatively (in terms of time, as the difference between the scheduled departure time and the actual departure time) or qualitatively. It is not possible to clearly define which departures are considered delayed. For the purpose of this study, a value of 15 minutes is assumed, i.e. all departures with a delay of 15 minutes or more are considered delayed. This parameter can be adjusted depending on the interest of the person for whom the analysis is being performed.

Materials and methods

Data loading and preprocessing

The process of downloading data for this task consisted of several stages. For three years, we collected complete information about air flights from the FlightRadar24 service [11]. This data includes information about altitude, coordinates, departure and destination airports, airlines, aircraft models, and flight numbers - that is, complete information about every aircraft in the sky. Of these, 562,894 flight numbers and 278,508 aircraft registration numbers were exported.

After downloading all the flights, information about all the aircrafts was also collected, resulting in an extremely large database of aircrafts, and there are no such databases in public access. The disadvantage is that the entry requirement for the dataset is at least one flight in the last 3 years, which is why not all aircraft were included in the dataset, but the coverage is still very high.

The next step was to download historical METAR data, using the Iowa State University [12], which provides unlimited weather information, and then combine each flight with the weather conditions in effect at the time.

We also downloaded information about aircraft models from the ICAO website [13] and about airports, regions, and countries [14].

Since working with such a large dataset is quite a challenge, it was decided to reduce it. For the task of predicting delays, the dataset was filtered by the departure airport: the largest airport in the world by passenger traffic, Hartsfield–Jackson Atlanta International Airport, was chosen. It was also chosen because it has rather unstable weather conditions, which should strongly impact flight delays. The source file contains more than 1 million flights.

Next, we removed columns that had no data (i.e., all rows had missing values). Then, we calculated the number of flights adjacent to each departure - the count of flights occurring within a 5-minute interval before and after the scheduled departure time of each flight.

The next step involved merging this data with the aircraft dataset using the registration number. A left join was performed because not all flights contain detailed information about the aircraft. Rows with missing values in the flight delay column were then removed, as this is the target variable for the data analysis. The flight delay time was divided by 60 to convert the values from seconds to minutes for easier interpretation of results. Fig. 1 visualizes the distribution of aircraft models in the overall flight structure.

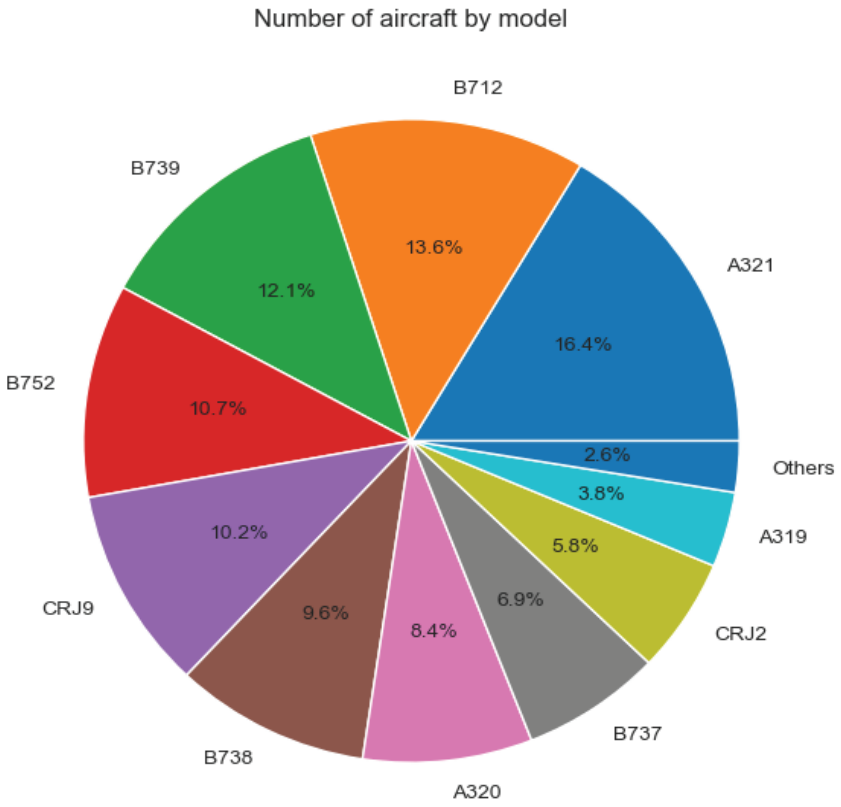

Fig. 1. Share of aircraft models in the overall structure of flights

ISSN 1560-8956

Modeling methods

Three methods were chosen to solve the problem of predicting flight delays, each of which can be used for both regression and classification. The first method is a histogram-based gradient-boosting tree [15]. Its main advantage is that, according to the documentation of the scikit-learn package [16], it is the only one that directly supports the use of missing values. With its help, we will build a regression model and check which features have the greatest impact on flight delays, and then leave only them, and then there will be few rows with missing values, which will make it possible to use other methods, such as stochastic gradient descent [17] and neural networks.

The stochastic gradient descent method was chosen because it is recommended [18] for use in the scikit-learn documentation in the article on model selection. Its main advantage is its high computational efficiency with very large datasets while maintaining a fairly high accuracy.

For the described task, we will also use a column transformer. Its idea is to scale numeric values in the range from -1 to 1 using StandardScaler from the scikit-learn package, and to convert categorical variables into a boolean array using OneHotEncoder from the same package.

Neural networks were chosen because they usually have slightly higher accuracy and can be easily saved to a file for further use in predictions, but they have a very serious drawback - the need for a very long training time.

An additional benefit of the first two methods is maintaining class imbalance by using weights for each class. If the number of delayed flights and non-delayed flights is very different, this will not cause classification problems.

Another disadvantage of the last two methods is that they do not support rows with missing values, and therefore such rows must be deleted.

Results and discussion

As already mentioned, the dataset contains a large number of missing values, which significantly limits the list of available methods. Because of this, it is a good idea to use a regression model that supports missing values to see which features have the greatest impact on the target variable and extract only those. To do this, we built a correlation matrix and saw that there is no strong correlation with any feature.

It is also worth noting that there are a large number of categorical variables (aircraft model, airline, some METAR information, etc.), such variables are not reflected in the correlation matrix, but they can also strongly affect the target variable, so we can derive median flight delays for a number of such categorical variables (Fig. 2, Fig. 3). The median is chosen due to the fact that flight delay values are distributed asymmetrically with a large

number of outliers, and average values may not accurately represent the situation. As you can see, departure delays depend quite strongly on the airline and somewhat less on the aircraft model.

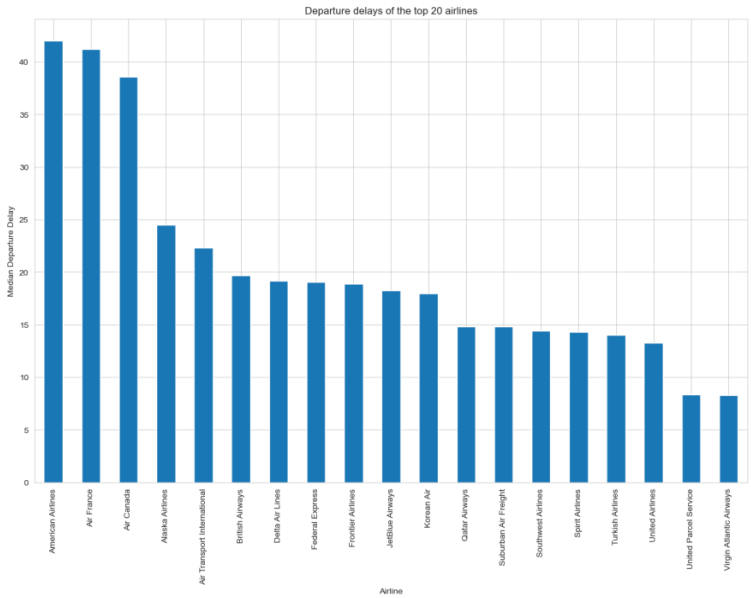

Fig. 2. Median values of departure delays for the largest airlines

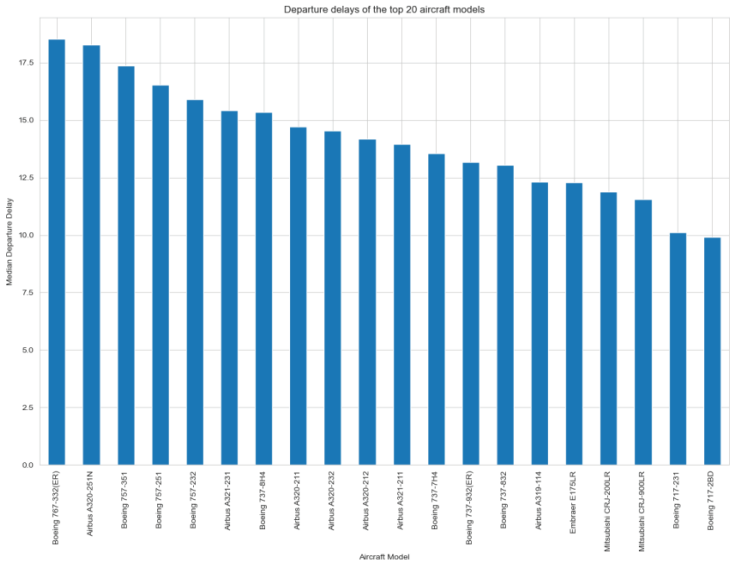

Fig. 3. Median values of departure delays for the most used aircraft models

Some other features were also discarded: flight number, aircraft registration number (they can be used, but then the model will be less flexible and much more complex), departure airport (since models are built for only one airport) and destination airport (since it is

 ISSN 1560-8956

theoretically quite difficult to establish a relationship between the flight departure delay and the destination airport, but adding this feature will make the model much more complex), the actual times of departure and arrival, and arrival delay (since they are unknown at the time of flight planning), the text of the aircraft model, the country of aircraft registration, as well as some minor information from the METAR report.

Next, we build a regression model to calculate the importance of each feature. As mentioned earlier, this model is a histogram-based gradient boosting tree. Due to the long training time of the model (about 8 minutes), parameter tuning is not feasible here, and therefore we used the default parameters. The model may yield slightly lower accuracy, but this is unlikely to significantly affect which features it prioritizes. After training the model, we evaluate its accuracy (Fig. 4). For this, three metrics were used: mean absolute error, root mean square error, and median absolute error.

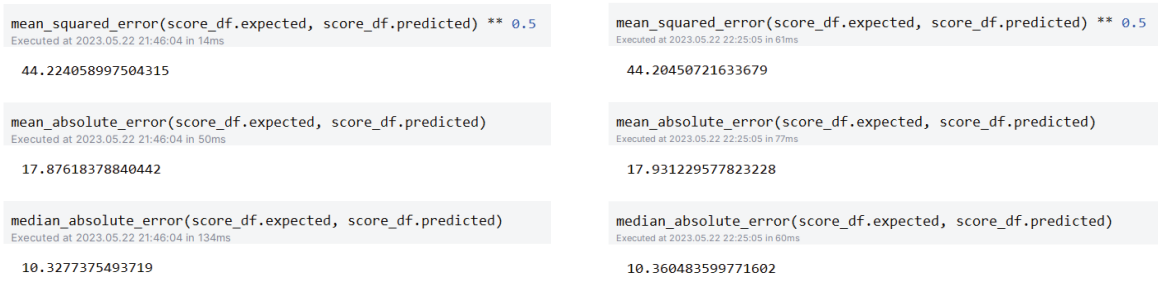

Fig. 4. Evaluation of the histogram-based gradient boosting tree before and after the removal of the least important features from the dataset

We can see that the accuracy is quite low, although the median error of 10 minutes does not seem that significant. Let's use the permutation_importance function of the scikit-learn package to determine the importance of each of the features and display them on the plot. We can see that some features do not affect the flight delay at all, and that the model of the aircraft and the airline operating the flight have the greatest impact on the target variable. Let's keep only the 11 most influential features and repeat all these steps (Fig. 5).

It's also worth looking at the accuracy metrics to see if removing some features has worsened the accuracy of our model - the accuracy hasn't dropped much, so these changes will be beneficial, as the model will now learn faster and it will be possible to remove rows with missing values and use other models.

Now let's move on to the classification problem. As already mentioned, we will classify as follows: if a flight is delayed by less than 15 minutes, it is not delayed, and if it is delayed by more than 15 minutes, it is delayed. It should also be noted that the case where a flight that was not delayed is classified as delayed is preferable to the opposite, as the opposite

situation can lead to problems with further planning, and therefore it is better to take a pessimistic forecast than an optimistic one. In other words, the type I error is more desirable than the type II error.

First, let's use a histogram-based gradient boosting tree to handle missing values. We achieved an accuracy of 63.16%, which is quite a good result. Let's display the confusion matrix to examine in more detail how the model performs (Fig. 6).

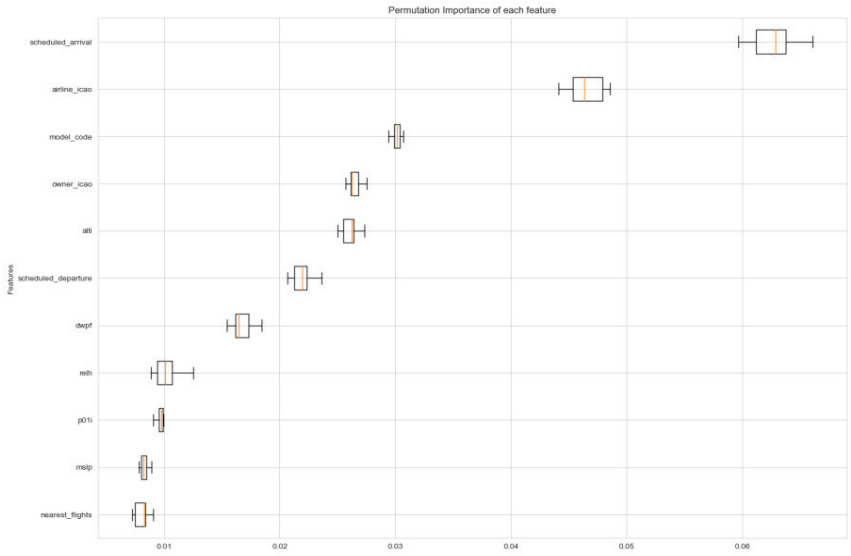

Fig. 5. Plot of the importance of each feature
in the dataset after removing the least important features

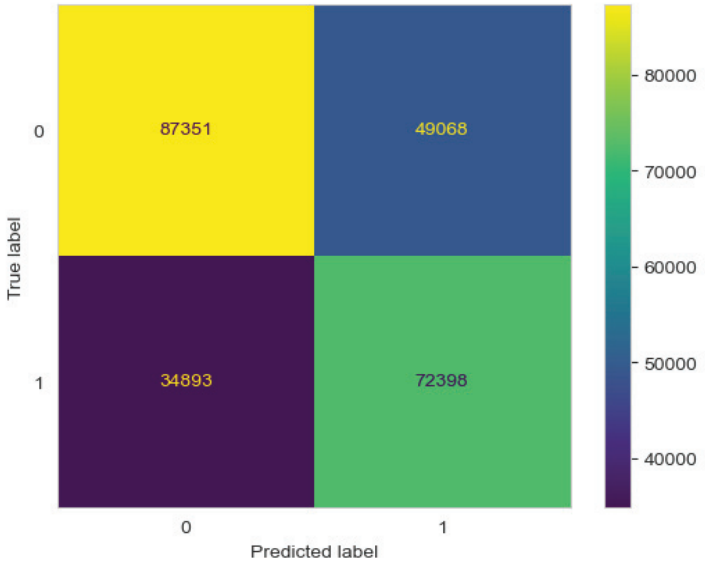

Fig. 6. Confusion matrix of the histogram-based
gradient boosting classification tree

ISSN 1560-8956

As you can see, the model more often incorrectly assigns the status that the flight will be delayed. Overall, this is a good thing.

The next step is to drop rows with missing values for further analysis. After this action, we can proceed to stochastic gradient descent. For this method, we will first perform classification, and then regression. The accuracy achieved by the classifier was 63.15%, which is very close to the previous classification method. However, the training time was reduced from 4.5 minutes to 10 seconds, and the prediction time for the test sample from 12 seconds to 1 second, while only about 22% of the dataset was dropped. Considering the large size of the dataset, this is not a significant loss. Once again, we'll display the classifier's performance using the confusion matrix (Fig. 7).

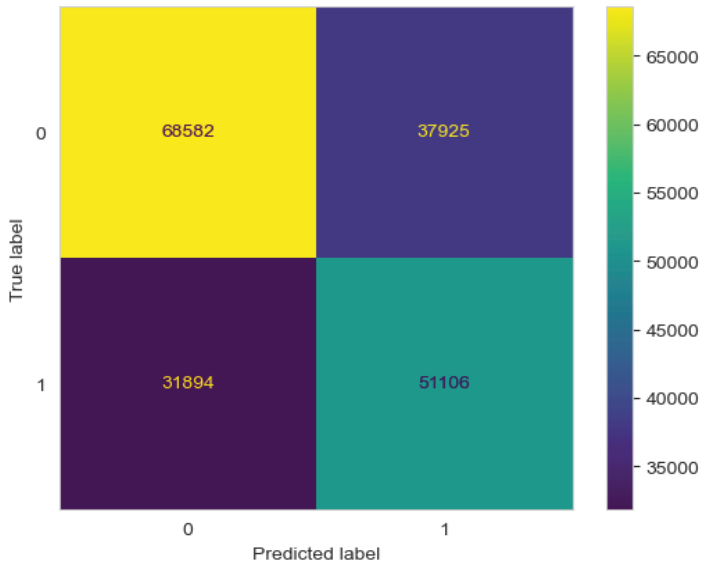

Fig. 7. Confusion matrix of the stochastic gradient descent classification model

As we can see, the situation remains unchanged. Most of the samples are classified correctly, while the model is more likely to classify non-delayed departures as delayed, which generally suits us.

Since this method operates quite rapidly, it's already pertinent to use cross-validation with grid search. The selection of the following parameters was performed: the loss function, the constant that multiplies the regularization term, and the penalty. The best result is produced by the model with these parameters: regularization constant – 0.0001, loss function – log_loss, and penalty l2.

The next step is to use neural networks. The neural networks for regression and classification analyses will have the same structure: a feed-forward neural network with four

dense layers: 256, 128, 64 neurons in the first three layers and 1 in the last layer. We will use the standard activation function: ReLU [19], and the optimizer is Adam [20]. The only difference between the neural network for regression analysis and the neural network for classification analysis is the neuron activation function in the last layer: in the first case, it is a linear function (the output is a real number), and in the second case, it is a sigmoid that gives a value in the interval (0; 1), which will correspond to the labels of our classes. The loss function will also differ: the mean absolute error in the first case and the binary cross-entropy in the second.

Let's set the batch size to 32. This size will avoid overtraining and reduce the training time, which, in turn, will take place over 30 epochs (iterations). Also, after each epoch, the neural network will evaluate the accuracy of the test data, so that we can objectively evaluate the accuracy of the model at each epoch.

First of all, let's try this approach for regression analysis. After completing training, we can display the model’s mean absolute error history on the plot (Fig. 8).

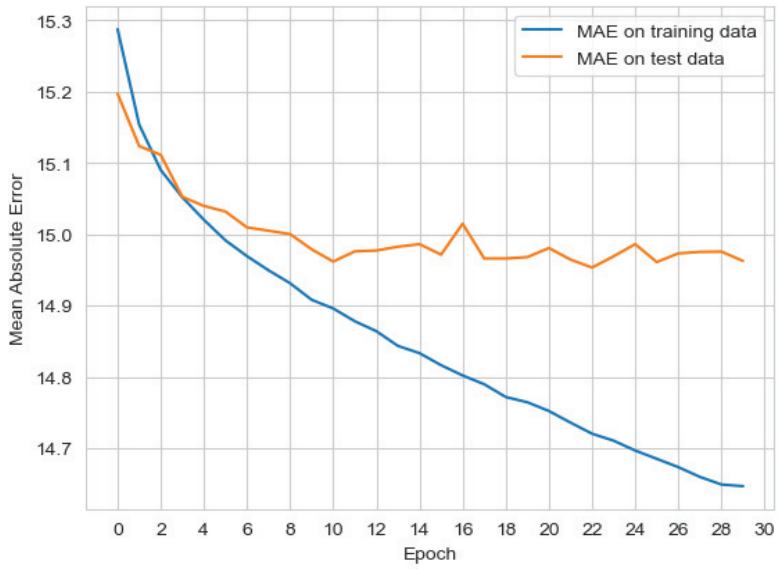

Fig. 8. Dependence of the mean absolute error
of the neural network for the regression problem on the epoch

We can see that the accuracy reached a plateau after about epoch 10. We can also see that the error has become smaller compared to both previous regression models but is still quite large. We also note that the accuracy does not increase significantly with each epoch, which means that it takes a very long time to train a better model. We can also calculate all the other metrics that were calculated for the previous models (Fig. 9).

ISSN 1560-8956

```
mean_squared_error(score_df.expected, score_df.predicted) ** 0.5
Executed at 2023.05.23 00:36:34 in 178ms

44.48343094268268

median_absolute_error(score_df.expected, score_df.predicted)
Executed at 2023.05.23 00:36:35 in 121ms

5.4194559097290025
```

Fig. 9. Evaluation of the neural network for the regression problem

We can see that the root mean square error has not changed much, but the median absolute error has decreased very significantly – almost twice.

The next step is to train the neural network for classification. We will classify according to the initial rules (that is, a delay of more than 15 minutes is considered a delay). Again, let's display the training results on the plot (Fig. 10).

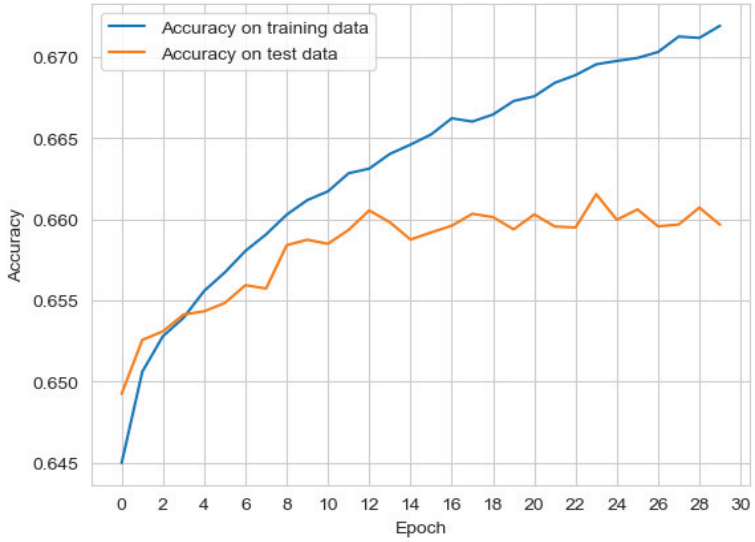

Fig. 10. Neural network accuracy dependence on epoch for classification task

We can see that after the 12th epoch, the accuracy plateaued and didn't change further. Also, it is worth noting that the accuracy slightly increased compared to the previous classification models. Now, we can display the results using the confusion matrix (Fig. 11).

Despite a certain increase in the model's accuracy, the results of the neural network's operation have actually worsened. The neural network classifies a delayed flight as not delayed (i.e., commits a type II error) twice as often as the reverse, which, as previously noted, is unlikely to satisfy the end user.

The next step is to compare the performance of all models, but this is currently not possible: a histogram-based gradient boosting trees were trained on a dataset with missing

values, while stochastic gradient descent and neural networks were without them, so such a comparison would be incorrect. The problem can be solved very easily: we need to build new regression and classification models using histogram-based gradient boosting trees.

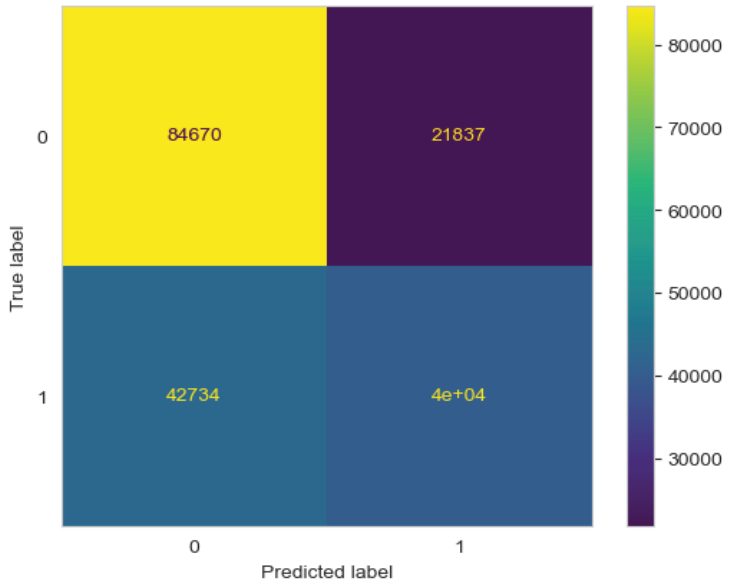

Fig. 11. Neural network confusion matrix for the classification problem

We will compare the regression models by all of the following metrics: root mean square error, mean absolute error, and median absolute error (Fig. 12).

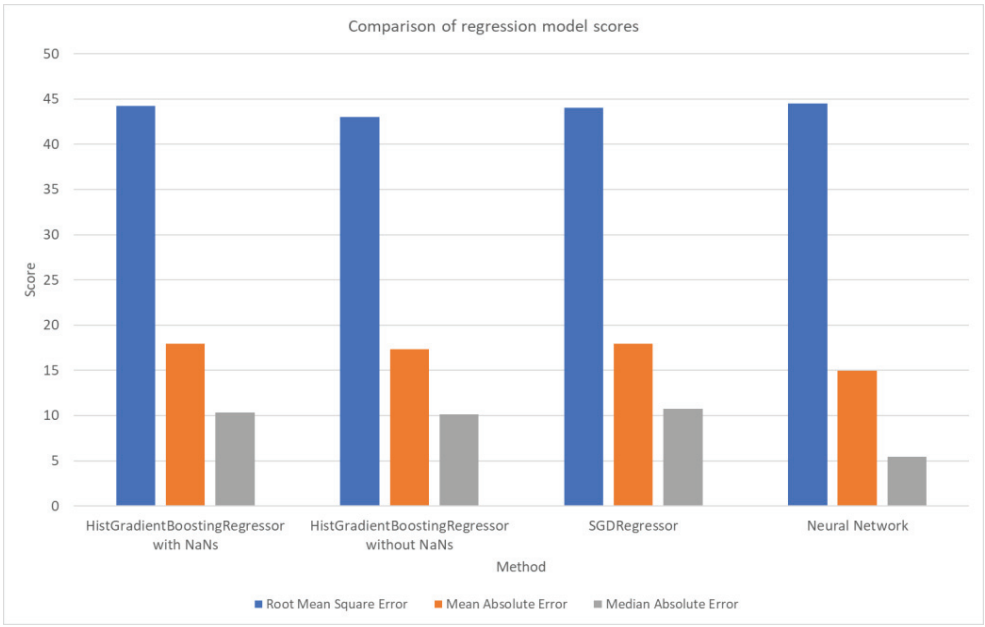

Fig. 12. Comparison of regression models

ISSN 1560-8956

Summarizing the results of the regression models, we can draw the following conclusions: the development of such models managed to reduce the median absolute error from 10.5 to 5 minutes, which is an excellent result in predicting flight delays. Nevertheless, the mean absolute error was reduced only slightly, by only 16%, and the root mean square error remained constant for all models. This indicates that the models are good at predicting small delay values, while large delays are predicted much worse. In general, this result is satisfactory, because most often flight delays are not too long, and large delays are the exception to the rule, not the rule, and therefore most often we will have to predict rather small delays, which the models handle well. Obviously, the best model is the neural network: it improved the mean absolute error by 16% and the median absolute error by 50%.

Now let's move on to evaluating the classification models. We use three metrics for this evaluation: accuracy, precision, and recall. For this paper, recall is the most important metric because the user will be better off if a flight that was predicted to be delayed does not get delayed, rather than vice versa because it is always better to prepare for the worst. Let's calculate all of these estimates for the models used and show them in the graph (Fig. 13).

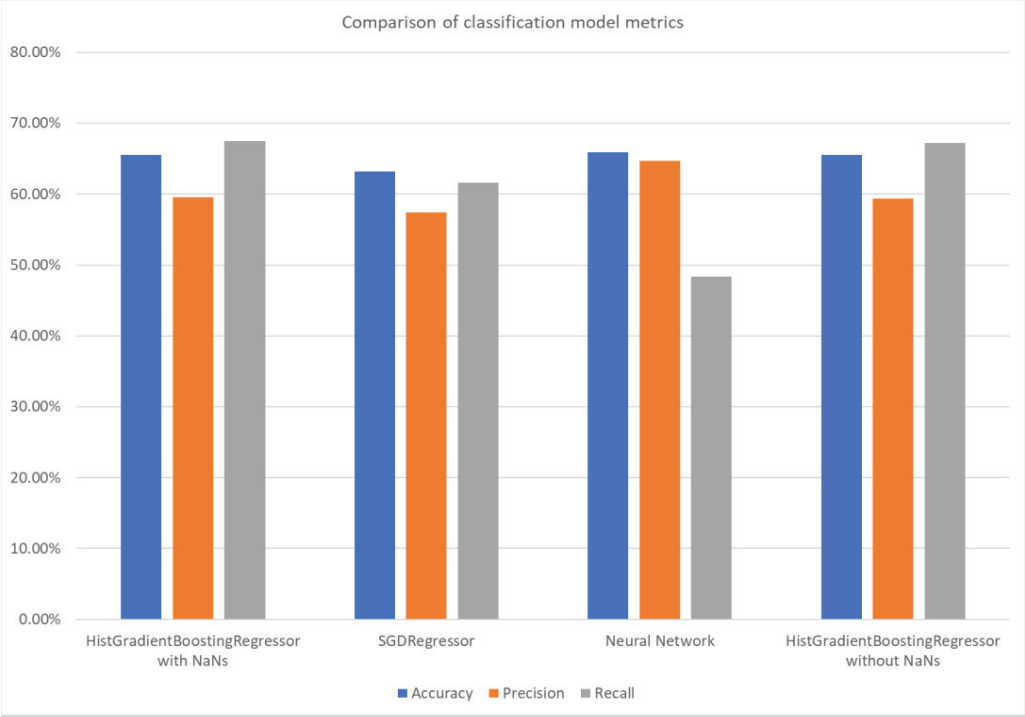

Fig. 13. Comparison of classification models

As you can see, the worst recall, surprisingly enough, is in the neural network, while the highest recall is in the histogram-based gradient boosting tree. It is also important to look at the accuracy of the model: it is more or less the same in all of them, while the highest is

again in the histogram-based gradient boosting tree. A simple conclusion can be drawn: the combination of high accuracy and high recall makes this model the best among all the models presented in this paper. Nevertheless, it is important to remember that in a different scenario, when the user needs the model to classify late flights as being on time, a neural network should be used, because it is this model that has achieved the highest precision.

Conclusion

There are 5 regression models and 5 classification models of three different types were built in this paper to predict flight delays at Atlanta Airport, USA. Among the regression models, the neural network showed the best result, and among the classification models, a histogram-based gradient boosting tree if the user prefers the type II error, and a neural network if the user prefers the type II error.

The models created can be used in many aviation-related industries: airlines will be able to plan their flight times more accurately, and logistics companies will be able to reduce or increase the time intervals between different parts of cargo delivery, and passengers, thanks to the conducted analysis of the influence of factors, will be able to clearly see which airlines to use and which not to. This partially applies to aircraft owners as well, as it has been found that the delay of a flight greatly depends on the model of the aircraft. With this knowledge, one can adjust their aircraft fleet.

REFERENCES

1. Scikit-learn: machine learning in Python. URL: https://scikit-learn.org (from: 29.05.2023).
2. Pandas – Python Data Analysis Library. URL: https://pandas.pydata.org (from: 29.05.2023).
3. NumPy. URL: https://numpy.org (from: 29.05.2023).
4. Matplotlib – Visualization with Python. URL: https://matplotlib.org (from: 29.05.2023).
5. Keras: Deep Learning for humans. URL: https://keras.io (from: 29.05.2023).
6. Shao, W.; Prabowo, A.; (...); Salim, F.D. Predicting flight delay with spatiotemporal trajectory convolutional network and airport situational awareness map // Neurocomputing, 2022, vol.472, pp.280-293; DOI: 10.1016/j.neucom.2021.04.136
7. Zoutendijk, M. and Mitici, M. Probabilistic Flight Delay Predictions Using Machine Learning and Applications to the Flight-to-Gate Assignment Problem // Aerospace, 2021, vol. 8 (6); DOI: 10.3390/aerospace8060152
8. Khan, W.A.; Ma, H.L.; (...); Wen, X. Hierarchical integrated machine learning model for predicting flight departure delays and duration in series // Transportation research part c-emerging technologies, 2021, vol.129; DOI: 10.1016/j.trc.2021.103225
9. Schosser, D. and Schonberger, J. On the performance of machine learning based flight delay prediction-investigating the impact of short-term features // Promet-traffic & transportation, 2022, vol. 34 (6) , pp.825-838

10. Terminal Aerodrome Forecast (TAF). URL: https://flightcrewguide.com/wiki/ meteorology/ terminal-aerodrome-forecast-taf (from: 29.05.2023).

11. Flightradar24: Live Flight Tracker – Real-Time Flight Tracker Map. URL: https://www. flightradar24.com (from: 29.05.2023).

12. Iowa Environmental Mesonet. URL: https://mesonet.agron.iastate.edu/request/ download.phtml (from: 29.05.2023).

13. Aircraft Type Designators. URL: https://www.icao.int/publications/doc8643/ pages/search.aspx (from: 29.05.2023).

14. Open data @ OurAirports. URL: https://ourairports.com/data (from: 29.05.2023).

15. Histogram Boosting Gradient Classifier. URL: https://www.analyticsvidhya.com/ blog/ 2022/01/histogram-boosting-gradient-classifier (from: 29.05.2023).

16. Imputation of missing values. URL: https://scikit-learn.org/stable/modules/ impute.html (from: 29.05.2023).

17. Stochastic Gradient Descent (v.2). URL: https://leon.bottou.org/projects/sgd (from: 29.05.2023).

18. Choosing the right estimator. URL: https://scikit-learn.org/stable/tutorial/machine_learning_ map/index.html (from: 29.05.2023).

19. A Gentle Introduction to the Rectified Linear Unit (ReLU). URL: https://machinelearningmastery.com/rectified-linear-activation-function-for-deep-learning-neural-networks (from: 29.05.2023).

20. Gentle Introduction to the Adam Optimization Algorithm for Deep Learning. URL: https://machinelearningmastery.com/adam-optimization-algorithm-for-deep-learning (from: 29.05.2023).