PILOT'S PERFORMANCE ASSESSMENT THROUGH DATA MINING

Konstantin Metodiev

Space Research and Technology Institute, Bulgarian Academy of Sciences
e-mail: komet@space.bas.bg

Key words: *Flight Simulator, Machine Learning, Cessna 172*

Abstract: *The report outlines an approach to evaluating the effectiveness of a light aircraft pilot in flight simulator during ILS landing procedure. In the course of flight task fulfilment, data from eye tracking measurements is being gathered. The data are further utilized for machine learning of several dichotomous classification models. The models' accuracy is assessed to facilitate the automatic ranking of pilots while executing the aforementioned task in particular.*

ОЦЕНКА НА РАБОТАТА НА ПИЛОТА ЧРЕЗ ИЗВЛИЧАНЕ НА ЗАКОНОМЕРНОСТИ ОТ ДАННИ

Константин Методиев

Институт за космически изследвания и технологии, Българска академия на науките
e-mail: komet@space.bas.bg

Ключови думи: *симулатор, машинно обучение, Сесна 172*

Резюме: *В доклада е представен метод за оценка работата на пилот на лек самолет в полетен симулатор при изпълнение на заход за кацане по ILS. По време на изпълнение на полетната задача се събира статистически материал от данни от окулография. Събраните данни са използвани за машинно обучение на няколко дихотомни модела за класификация. Точността на моделите е оценена, за да се използват за машинно класиране на пилоти при изпълнение на посочената задача.*

Introduction

In air transport and general aviation, data mining is used to enhance flight safety and assess pilot's performance in numerous ways. Airlines gather extensive amounts of flight information from aircraft sensors and cockpit audio recordings. Machine learning algorithms examine this data to identify patterns pointing to pilot behaviors, including compliance with procedures, reaction times, and management of unusual situations, for example repeated deviations from standard operating procedures to name but a few. What is more, machine learning models are capable of detecting typical pilot behaviors or system interactions that may suggest fatigue, stress, or insufficient training. These irregularities are subsequently examined to enhance pilot training programs or operational procedures. Data mining facilitates the comparison of pilots' performance to that of peers or predetermined benchmarks. This can enable individualized instruction by highlighting areas where a pilot excels or needs more training. In addition, data mining helps to analyze contributing factors, such as pilot reactions, in mishaps or near-misses. Better training programs centered on particular abilities or decision-making procedures may result from this realization. Last but not the least, airliners, using time-stamped records, employ predictive models to foresee possible safety risks, thus allowing for proactive responses before problems worsen. In general, these data-oriented strategies offer a more unbiased, thorough perspective on pilot performance, enhancing safety and efficiency in flight operations.

Current research involves a simple investigation of pilots' performance within a flight simulator environment utilizing eye tracker data and data mining algorithms, see for example study case [1].

Materials and methods

The proposed research employs decision tree, logistic regression, random forest, and k-nearest neighbor classification algorithms implemented in Orange, [2] data mining toolkit, in order to assess pilot's performance, Fig. 1.

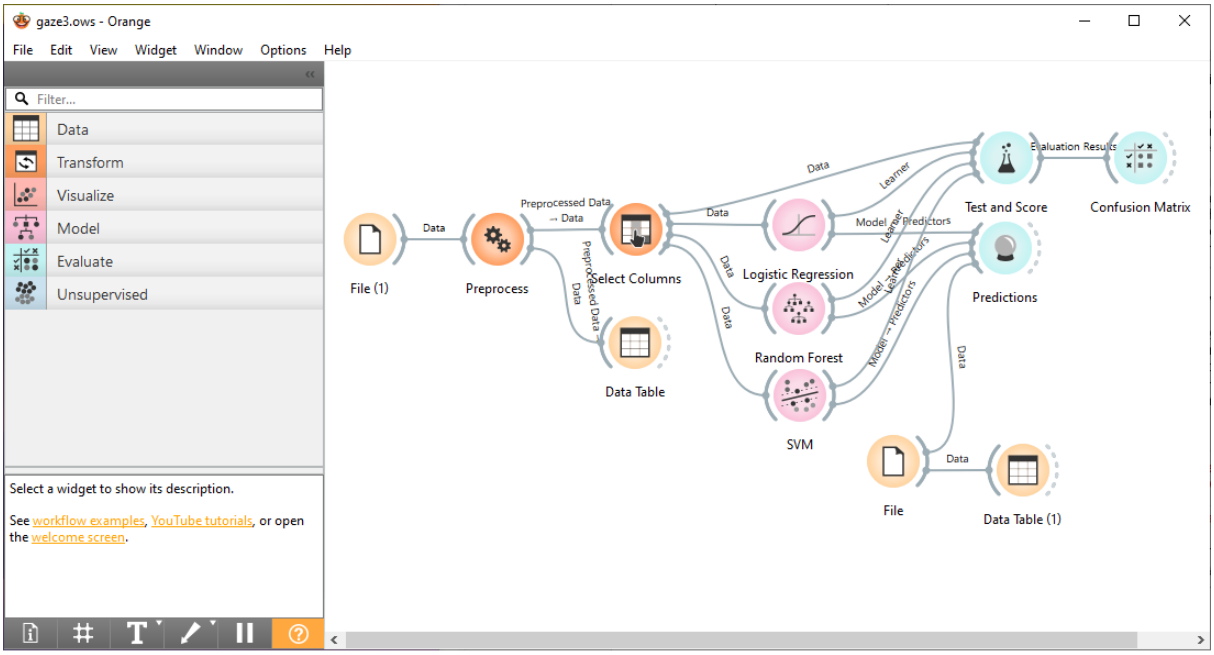

Fig. 1. Block diagram for validating models and making predictions in Orange IDE

The **logistic regression** is widely used classification method in machine learning. It's especially helpful for predicting binary results, such as yes or no, on or off, true or false, heads or tails, pass or fail, etc. The algorithm works by estimating the likelihood that a specific input belongs to a certain category. It achieves this by applying a logistic (or sigmoid) function to a weighted sum of the input features. The sigmoid function normalizes the output within [0; 1] interval which is why the output is dichotomous, i.e. it has only two possible outcomes.

The **random forest** classifier is a well-known machine learning algorithm employed in data mining for classification purposes. Throughout the training process, numerous decision trees are constructed, with each being trained on a random selection of data and features. Predictions from the trees are further combined to make a final decision. This randomness enhances precision and minimizes overfitting. The classifier is known for being robust, versatile, and effective with extensive datasets.

The **support vector machine** classifier is a supervised machine learning algorithm primarily used for binary classification tasks. It finds the optimal hyperplane that best separates two classes in the feature space, such as predicting whether the applicant will pass or fail. It is effective particularly when dealing with high-dimensional spaces and when the classes can be separated with categories that may not necessarily be linearly separable. This classifier is a preferred option in many applications because to its resilience and capacity to manage outliers and non-linear correlations.

The **confusion matrix** is a table that helps assess how well a classification algorithm is performing. It shows the number of correct and incorrect predictions, organized by each class. For instance, in case of a binary classification problem, the confusion matrix is a 2x2 table with following customary fields:

Table 1. Confusion matrix arrangement for a binary classification problem

	Actually positive	**Actually negative**
Predicted positive	True positives, TP	False positives, FP
Predicted negative	False negatives, FN	True negatives, TN

- True positives, TP: correctly predicted positive cases
- True negatives, TN: correctly predicted negative cases
- False positives, FP: incorrectly predicted positive cases (actually negative)
- False negatives, FN: incorrectly predicted negative cases (actually positive)

From confusion matrix elements, following performance metrics might be evaluated:

- Accuracy: (TP + TN) / (TP + TN + FP + FN)
- Precision: TP / (TP + FP)
- Recall (True Positive Rate, Sensitivity): TP / (TP + FN)
- Specificity (True Negative Rate): TN / (TN + FP)
- F1 Score: 2 * (Precision * Recall) / (Precision + Recall)

The correctly classified instances lie on the matrix main diagonal (colored in green).

For a multi-class classification, the confusion matrix expands to an NxN table, where N is the number of classes, showing the counts of actual vs. predicted class combinations. The confusion matrix provides detailed insight into how well the classifier is performing, especially in imbalanced datasets, where accuracy alone might be misleading.

The **ten-fold cross-validation** is a common method used in data mining to evaluate how well a predictive model works and whether it is applicable or not. It gives an estimate of how the model will perform on new data by dividing the dataset into 10 subsets, called folds. The model is trained on nine of these folds and tested on the remaining one. This process is repeated 10 times, each time with a different fold used for testing. The overall performance is then summarized by averaging metrics like accuracy, precision, recall, and F1-score across all 10 runs.

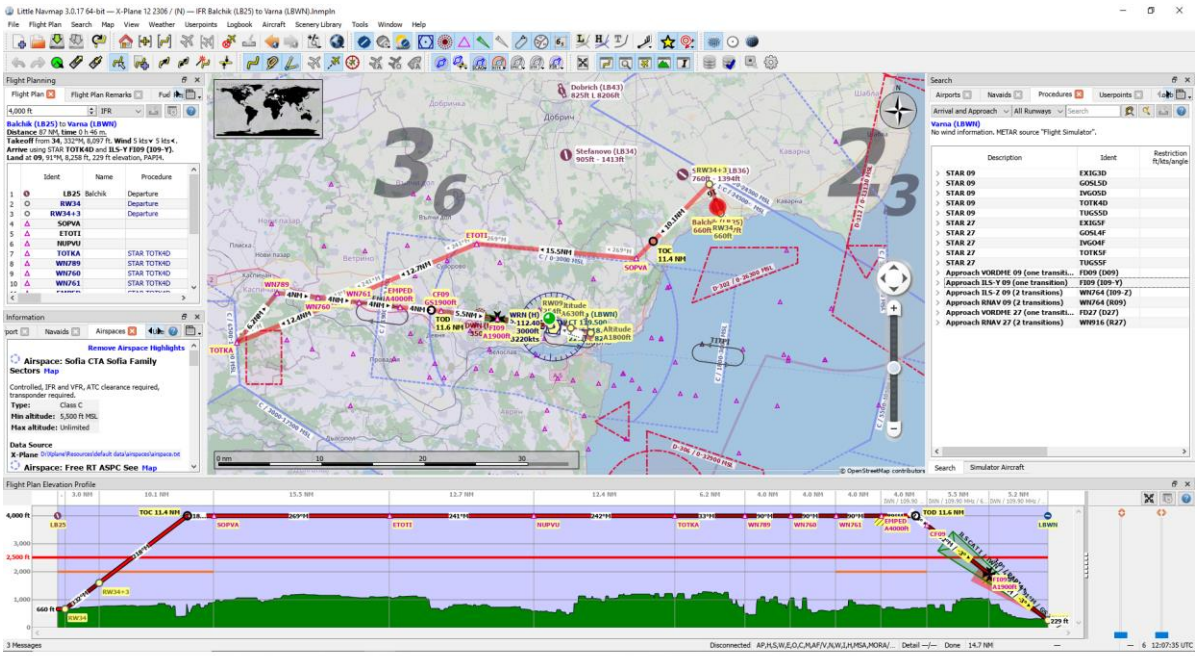

Fig. 2. Flight plan including standard terminal arrival route RWY 09 for LBWN in Little NavMap

The flight task scenario is executed in X-Plane v.12 flight simulator, [3]. A flight plan, Fig. 2 was created in advance in Little NavMap, [4] and uploaded to GNS 530/430 GPS receivers which can be found on aircraft Cessna 172 SkyHawk, [5] dashboard. The examined candidate is told to follow a standard terminal arrival route to Varna airport, LBWN and land using ILS-Y approach at RWY 09. The aircraft flies along the entire flight plan automatically. The candidate is exclusively focused on adjusting throttle and mixture controls to keep EGT (exhaust gas temperature) and engine RPM (revolutions per minute) within acceptable limits. Not long before touchdown, the candidate disengages the autopilot by pressing a dedicated switch on the flight yoke and lands manually. It takes the applicant about eight minutes to complete the task.

During flight task fulfilment, a desktop-based eye tracker stores gaze fixations number and duration, Fig. 3 within zones of interest defined beforehand. Saccadic pupil movement is not taken into account. The eye tracker used is GazePoint GP3 HD, Fig. 3. Data log rate is 150 Hz. The GazePoint Analysis UX raw data recording and processing software computes total Viewed Time in seconds and per cent at each zone of interest. Upon flight plan completion, the results are directed to the Orange data mining tool for evaluating the applicant's performance.

Alongside fixation distribution within the screen, a time plot is created to portray and analyze gaze temporal dynamics across four zones of interest. The Circle Diagram created in this manner closely resembles the static "time plot of the gaze data" suggested by Räihä et al., [6]. In the diagram, fixation ordering pattern (if any) is of primary interest.

Fig. 3. GazePoint GP3 HD desktop-based eye tracker, pulse rate sensor, and self-engagement report tool

The flight simulator display has been pre-divided into four areas of interest as shown in Fig. 4:

- Primary flight instruments including air speed indicator, attitude indicator, barometric altimeter, turn coordinator, heading indicator, variometer, propeller RPM and Hobbs meter.
- Secondary flight instruments including VOR1 / ILS gauge, VOR2 gauge, automatic direction finder (ADF) gauge, audio switching panel, GNS 530/430 GPS receivers, transponder, autopilot. The ILS receiver is set to CAT I IWN frequency. The ADF receiver, while not assigned to a specific zone of interest, has its frequency set to Devnya DWN NDB.
- Tertiary flight instruments including chronometer, fuel gauge, exhaust gas temperature and fuel flow, oil temperature and pressure, vacuum pressure and battery ammeter.
- The environment visible through windshield.

The pilot's head spontaneous motion lies within acceptable limits which is why it can be disregarded.

Fig. 4. Cessna 172 cockpit overlaid with zones of interest and dynamic heatmap in X-Plane 12

During training phase, an experienced pilot fulfills the flight assignment and achieves an outstanding score. The pilot is considered etalon or prototypic. Data for the machine model training is generated by adding white Gaussian noise to the original dataset by function awgn in GNU Octave, [7]. The signal-to-noise ratio varies based on the required pilot's skills. A good pilot is thought to stray less from the original dataset, whereas a poor one diverges significantly. In this way, data is generated for applicants with different level of experience. Whenever an applicant decides to take the exam, the machine model makes a prediction and reports whether the applicant passed or failed the test. Miscalculations are highly unlikely. It depends on how well the classifier is trained.

Results

In Fig. 5, confusion matrices pertained to three used classifiers are shown.

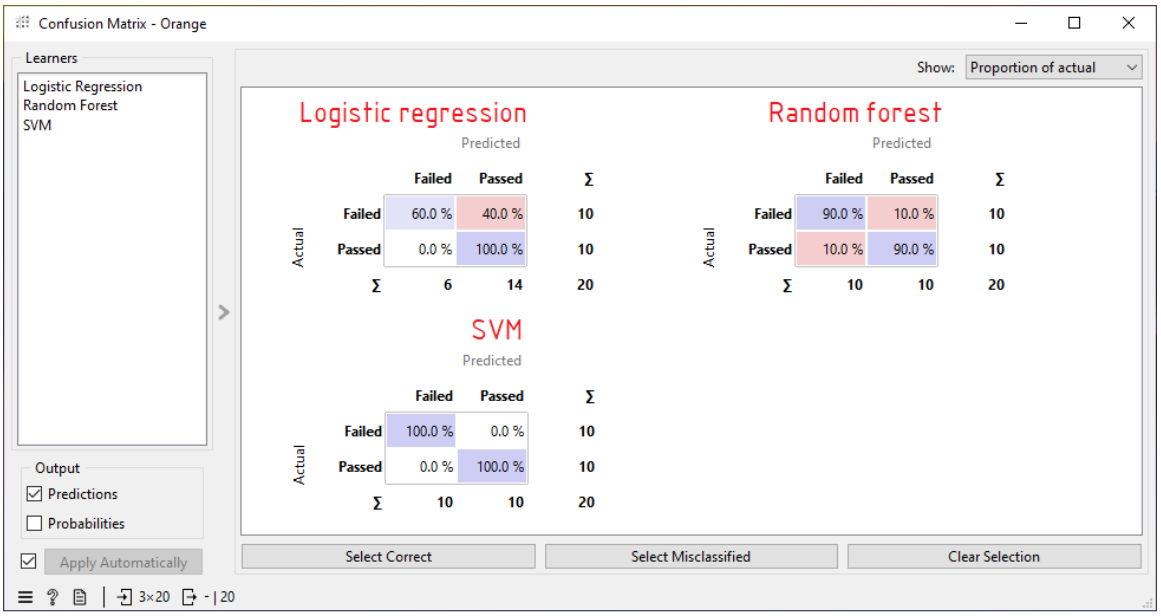

Fig. 5. Confusion matrices related to used classifiers

Upon double clicking on Test and Score widget, following results are revealed.

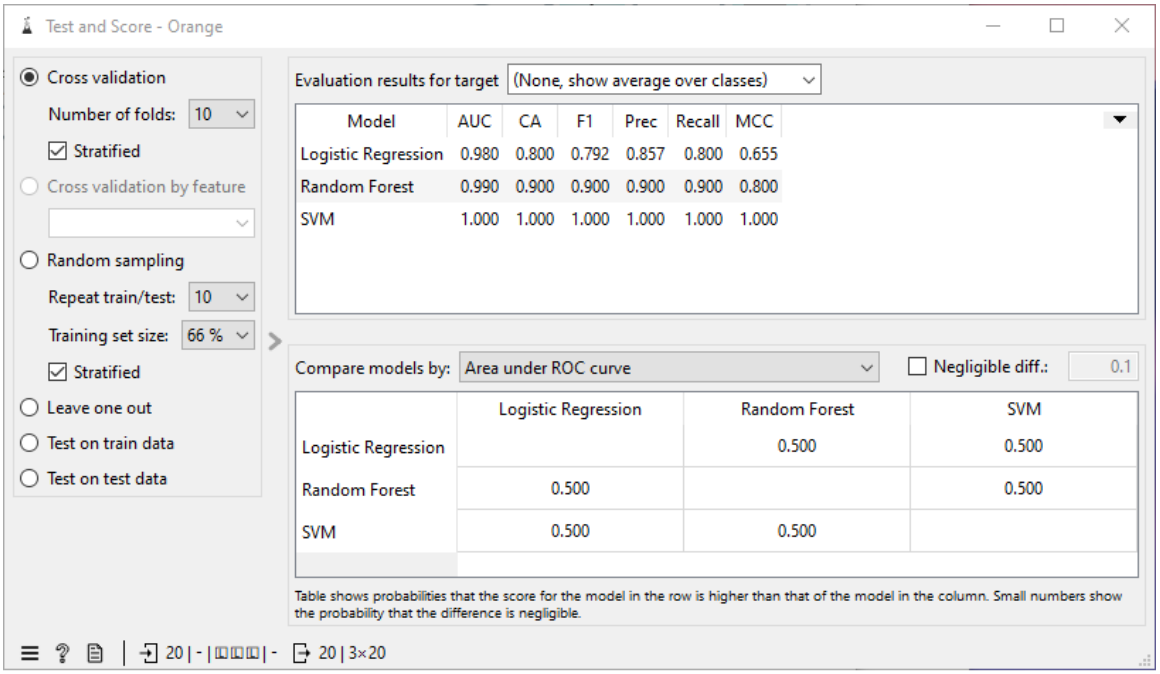

Fig. 6. Test and Score widget contents

In Fig. 7, the applicant's results are displayed.

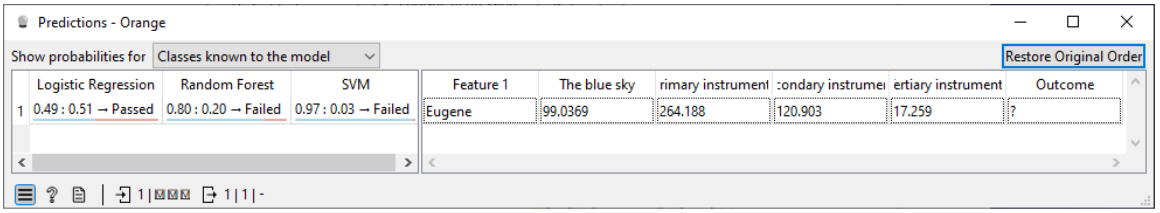

Fig. 7. The applicant's results

In Fig. 8, a circle diagram is shown depicting temporal dynamics of fixations. In the right-hand side, the adopted formalism is briefly explained. The ordinate represents four zones of interest whilst the abscissa indicates time, s.

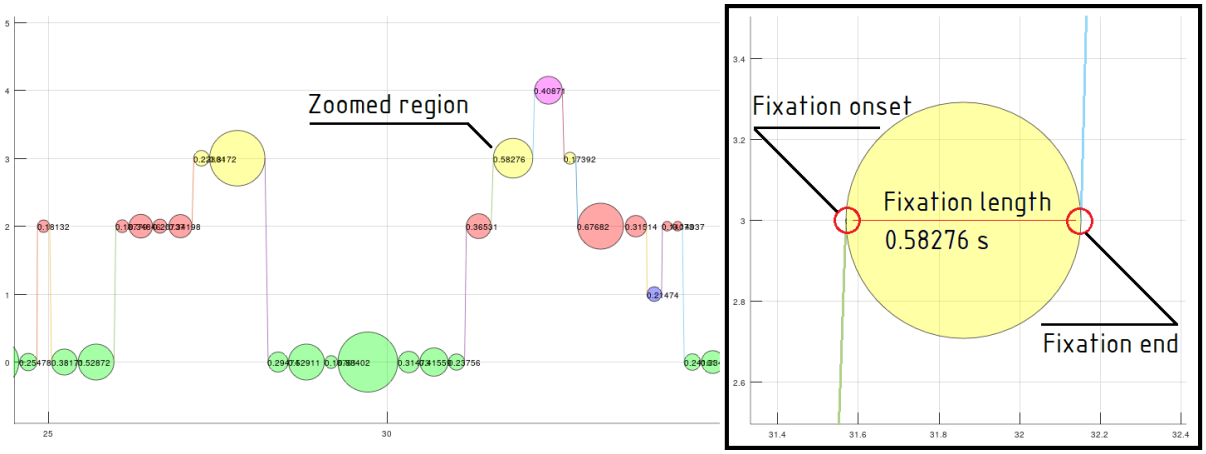

Fig. 8. A fragment of circle diagram

Discussion

According to Fig. 5 (confusion matrices) the Logistic Regression classifier exhibits the poorest performance in the present study compared to the other classifiers. On the other hand, all predictions made by Support Vector Machine are correct. Because of little to no experience, the applicant is expected to fail the test and this is what really occurs according to Random Forest and Support Vector Machine classifiers Fig. 7.

Accuracy metric indicates the number of correct predictions out of all made. **Precision** metric refers to the count of true positive instances among all those identified as positive. **Recall** metric provides insight into whether the classification model finds all instances of the positive class. When precision increases, recall diminishes, and the opposite holds true as well. **F1-score** is the harmonic mean of both. This indicator reaches its highest value when the precision equals the recall. In this regard, the Support vector Machine performs best according to metrics shown in Fig. 6.

The optimal machine learning algorithm for binary classification relies on several factors including the dataset nature and size, interpretability needs, and available computational resources. Choosing the best algorithm for evaluating applicants involves:

- Experimenting with multiple models via cross-validation;
- Considering an interpretability versus accuracy trade-off;
- Tuning hyperparameters to optimize performance.

Overall, a trade-off between accuracy and robustness for binary classification tasks is commonly sought.

In current research, GNU Octave [7] is used to process raw data and draw circle diagram.

References:

1. Bo Fu, P. Gatsby, A. Soriano, K. Chu, and N. Guardado, Towards Intelligent Flight Deck – A Preliminary Study of Applied Eye Tracking in the Predictions of Pilot Success and Failure During Simulated Flight Takeoff, Proceedings of the First International Workshop on Designing and Building Hybrid Human-AI Systems co-located with 17th International Conference on Advanced Visual Interfaces AVI 2024, Arenzano (Genoa), Italy, June 3rd, 2024 https://ceur-ws.org/Vol-3701/
2. https://orangedatamining.com/
3. https://www.x-plane.com/
4. https://albar965.github.io/
5. https://www.x-plane.com/manuals/C172_Pilot_Operating_Manual.pdf
6. Räihä, K. J., Aula, A., Majaranta, P., Rantala, H., & Koivunen, K., 2005. “Static visualization of temporal eye-tracking data. In IFIP Conference on Human-Computer Interaction (pp. 946–949). Springer, Berlin, Heidelberg https://www.researchgate.net/publication/221054730_Static_Visualization_of_Temporal_Eye-Tracking_Data
7. https://octave.org/