Journal of Open Aviation Science (2025), Vol.**3**
doi:10.59490/joas.2025.8111

JOAS

EDITORIAL

Reviews and Responses for Open Machine Learning Models for Actual Takeoff Weight Prediction

Authors: Mayara C. R. Murça, Marcos R. O. A. Maximo, Joao P. A. Dantas, João B. T. Szenczuk, Carolina R. Lima, Lucas O. Carvalho, and Gabriel A. Melo

Reviewers: Richard Alligier, Ryota Mori, and Junzi Sun

Editor: Xavier Olive

1. Original paper

The DOI for the original paper is https://doi.org/10.59490/joas.2025.7963

2. Review - round 1

2.1 Reviewer 1

This article presents the methodology used by the authors to obtain a machine learning model predicting the takeoff weight of commercial flights in 2022. This work was done to compete in the Performance Review Commission (PRC) Data Challenge 2024 in which the authors ranked second.

The article describes the data and the method used to obtain good predictions of aircraft takeoff weight. An analysis of the feature importance and an ablation study are done, showing which parts of the model are important or not. Some parts of the method might not be useful which is fine given the limited time competition context. Readers might think that these parts are useful. In order to not mislead the readers, the "Detailed Remarks/Questions" section below details which parts might be tagged as not (that) useful. Some details are missing and there is one theoretical paragraph that might not be relevant (see "Detailed Remarks/Questions" below), but given this easy to add/correct, I recommend an accept with minor revision.

2.1.1 Questions Related to "2.2 Data Preprocessing" and "2.3 Feature Engineering"

- A1 - ADS-B trajectory data are somewhat "dirty", the article does not mention any "cleaning" process. I looked at `get_traj_features4.R` and did not see any but I am not "fluent" in R and maybe I missed something. Is there really no trajectory cleaning ? Maybe the article could state clearly if there is a cleaning process or not, and it should be described in the article if there is one.
- A2 - We do not always have data for the whole trajectory. As a consequence some statistics concerning the efficiency of the vertical profile might not be "accurate". For instance the total time flown in level flight of two flights might differ only because the ADS-B coverage is not the same. Do you have a way to handle these kind of unbalance due to the ADS-B coverage?
- A3 - Concerning the feature taxi_ratio in the duration-based section, it is the taxiing time over the total flight duration, indicating the inefficiency. This ratio is then normalized again by di-

viding it by the median value to facilitate the comparison across flights. However, looking at `feature_engineering.ipynb`, it is divided by the median of all the flights, hence it is simply a linear transformation of the feature that will have no impact on tree-based algorithms/models (which are invariant to strictly monotone transformation of their feature) and neural network models in which each feature is typically normalized/standardized beforehand. If I did not miss something on that, maybe the paragraph about the division by the median could be removed as this division by the median do not change anything and might misled the readers into believing that it does something.

- A4 - Concerning the speed per distance, still in the duration-based section, looking at `feature_engineering.ipynb`, it is actually equal to 1/flight_duration. As such, if I did not miss something, the phrase "This feature provides a normalized speed metric relative to the flight distance, assisting in analyzing speed efficiency over varying distances." in the article should be modified. Furthermore, this transformation is strictly monotone, and hence will have no impact on tree-based algorithms/models, aside from multiplying by 2 the probability to be se considered as a split feature during the tree growing phase.
- A5 - Looking at `get_traj_features4.R`, `feature_engineering.ipynb` and `catboost.ipynb`, for the "first ten points features", some features are not listed in the article despsite being used like the altitude and the sqrt_tas features. I know that the sqrt_tas features are in the end not used as they are eliminated by the catboot selection process, however they could be listed as they were considered in the selection process. As a side note, the "squared" transformation is strictly monotone, and as such tas_1 and sqrd_tas_1 are basically the same feature, and this might have an impact on their respective importance score.
- A6 - From the "Interaction Features" section alone, the rationale is not clear behind the operation done, how multiplying the airspeed by the specific energy "help us understand how an aircraft's speed influences its energy efficiency during different stages of flight". Could you elaborate on this?

2.1.2 Questions Related to "2.4 Predictive Models"

- A7 - As a suggestion, you could change the subsection "Predictive Models" to a section. With this change, you will be able to add two subsections to "Articficial Neural Networks", one for the classical and one for SAINT. An other benefit is having a new section "Gradient Boosted Trees" containing the three gradient based methods. This section could contain a description of the hyperparameter that are shared between these methods. As a side note, the description of these hyperparameters is made inside Extreme Gradient Boosting, the last method described.
- A8 - Even though you did not use optuna to tune the neural networks, maybe you can have hyperparameters tables for the classical and SAINT neural networks. For SAINT, you performed a grid search on the width, maybe you could show this grid of hyperparameters in a table. For the classical neural network, from the text, it is not clear how the hyperparameters were selected. As a side note the batch size is not mentioned in the text despite being an important parameter of the training phase.
- A9 - For the hyperparameter tuning of CatBoost, I am not sure that changing the learning rate is a good practice. The obtained hyperparameters are searched with a "big" learning rate, thus the obtained hyperparameter could be sub-optimal when considered together with a reduced learning rate. Maybe, the article could provide figures on compute times to help justify this choice.
- A10 - For the hyperparameter tuning of CatBoost, the early stopping is used to train the final model, it is not clear that it was also used in the hyperparameter tuning phase.

- A11 - Concerning the ensemble section, I am not sure the theoretical framework presented is consistent with the way the ensemble is built. The theoretical framework presented is typically used in bagging/random forest: the predictive models are just averaged without a "meta-learning" above them that optimize weights using the training set. As the coefficients are actual constants, the variance decomposition is easy to determine.
  In the article, the predictive models are combined using (a part of?) the training set to optimize the weights. Furthermore the bias-variance decomposition is the expectation over (at least) a random training set, as a consequence the weights are not constants (w.r.t. the training set) and the variance decomposition in the article cannot be made here, it is not that straightforward anymore. The ensemble method used in the article is probably more related to "stacking" ([1]), and a more relevant theoretical framework can be found in the "Super Learner" paper ([2]).
- A12 - Concerning the ensemble section, it could detail whether the training set used to learn the models is the same used to compute the optimal weights, or is it split in two (or K) parts to do that?

2.1.3 Questions Related to "3 Discussions"

- A13 - Concerning the Table 6 for the ablation study, the temporal and geographical features are described in their own subsubsections 2.3.2 and 2.3.4 but features such as "aircraft characteristics" and "operational features" are not described. Maybe the authors could add subsubsections for these two groups. Additionaly, some features in these groups are not described in the feature section.

2.1.4 Typos

- A14 - line 303: "hype parameters" to "hyperparameters".
- A15 - line 430: In "The A320 was the most used equipment", the word "equipment" sounds weird, maybe "aircraft type" would be better.

2.1.5 Appendices

Some thoughts I had reading the article that can be left unanswered by the authors as they might be unimportant, irrelevant or wrong:

- A16 - We do not always have data for the whole trajectory. As a consequence the first ten observed points are somewhat random: they are the first points inside the ADS-B coverage. The altitude "frontier" of the ADS-B coverage may vary from one location to an other. As such, the altitude of these first points may vary, regardless of the way the aircraft is operated. To illustrate how this can be problematic, let us consider the feature $VerticalRate_1$ which is the vertical rate computed on the first observed point. If you observe $VerticalRate_1\,(Flight_A) > VerticalRate_1\,(Flight_B)$, maybe $Flight_A$ is lighter than $Flight_B$ or maybe they have a similar weight but $Flight_B$ is higher than $Flight_A$ and consequently have a lower rate of climb. The feature "height relative to the origin airport" can mitigate this issue but the tree based models will have to "chain" two conditions/nodes: one on this altitude and the other on the VerticalRate. Intuitively, this consumes tree nodes (and hence model complexity), and as the tree growing algorithm is a greedy algorithm (it creates one condition/node at a time), it is not certain that the first condition will be selected (because it was the locally optimal choice) as the first condition might be good only if it is followed by the second one.
- A17 - The authors imputed the missing values using the median. I think this is for the best here, but sometimes if the fact they are missing is correlated to the variable to be predicted, it is better to left them as missing values. This way, missing values can be treated separately, and be put on the left or right side of the condition which should help reduce the training loss, see https://github.com/microsoft/LightGBM/issues/2921. In the authors case, the fact that the value

is missing is independent of the mass of the airplane, a priori, and hence it should be better to impute them with median. It was just a question that popped in my head during my reading.

2.2 Reviewer 2

The authors present an interesting framework for predicting aircraft weight using open data. The manuscript is well-structed and the results are promising. However, I have several concerns regarding the model creation process that require further attention.

- M1 - Sec 2.3.1: While the feature selection is important for accurate prediction, some choices are questionable from an aircraft dynamics perspective. For example, the mean temperature might be less informative than the deviation from ISA, which directly reflects atmospheric conditions influencing aircraft performance. Please justify the choice of using mean temperature. In the same way, the maximum altitude could be more relevant than average altitude for weight prediction.
- M2 - Line 140: How was the wind information obtained? Is it included in the open data sets?
- M3 - Line 157: I do not understand how the authors use the level segments. The manuscript mentions three segments (departure, enroute, and arrival). Is "level segment" a fourth category? Please provide a clear definition and explain its role.
- M4 - Line 180: Why were discrete flight durations used instead of continuous values, while most other features are continuous.
- M5 - Line 184: What is the "speed per distance"? Speed is time-dependent while flown distance is a single value of the flight. Please clarify the definition.
- M6 - Line 194: The manuscript calculates both "mean vertical speed" and "elevation gradient". These feature seems redundant.
- M7 - Sec 2.3: Sec. 2.3 lists various variables, but it is unclear which ones are finally used. You should provide a complete list of selected features and their count.
- M8 - Line 258: Please provide references for FastAI's Tabluar learner and SAINT model.
- M9 - Sec 2.4.1: More details about ANN structures are needed, such as the number of nodes, layers, etc. Additionally, please specify the hyperparameters and their chosen values?
- M10 - Figure 9 - This figure suggests that the distribution varies across aircraft types. This infers that the estimation model tries to improve the accuracy of yellow aircraft rather than green aircraft. It would be beneficial to calculate the RMSE separately for each type. Relying solely on a single overall RMSE might mask performance discrepancies.
- M11 - Table 5: Table 5 provides RMSE values for different algorithms. Including a baseline comparison, such as RMSE using typical weights in each aircraft type/range or just FAA_Weight, would provide valuable context for evaluating the model performance. Also, it is important to discuss the performance compared to existing methods in the literature.
- M12 - Table 6: The list of ablated components does not align with the sections in 2.3-2.4, which makes it difficult to understand which features are grouped together. Please clarify the mapping between components and feature categories.

2.3 Reviewer 3

The paper presents one of the winning solutions to the PRC data challenge. The description and explanation of the paper are very thorough, leaving almost no stone unturned. I recommend accepting the paper with minor revisions.

- S1 - It would be nice to mention the data source, Eurocontrol PRU, in the abstract of the paper.

- S2 - Line 13: Abbreviations can be removed or updated (these are examples from the template).
- S3 - I suggest including a table that provides an overview of all features used for training the model in section 2.3.
- S4 - Structurally, I think section 3.1 on exploratory data analysis could be moved earlier, perhaps after section 2.2. However, this is a personal preference.
- S5 - In section 3.4, it would be helpful to provide more background on the "ablation study," such as what it is and how it provides more insights into the models.
- S6 - The colors and labels in Figure 10 are a bit hard to see. Is there a better way to visualize the results? Perhaps by creating a subplot for each aircraft type?
- S7 - Finally, please create a permanent archive with DOI using Zenodo (or a similar service) for the GitHub repository. This way, the code is archived even when GitHub is no longer accessible.

3. Response - round 1

We would like to thank the reviewers for the valuable comments and feedback, which have contributed to enhancing the quality and clarity of our manuscript. Detailed responses to each reviewer's comments are provided below.

3.1 Reviewer 1

This article presents the methodology used by the authors to obtain a machine learning model predicting the takeoff weight of commercial flights in 2022. This work was done to compete in the Performance Review Commission (PRC) Data Challenge 2024 in which the authors ranked second.

The article describes the data and the method used to obtain good predictions of aircraft takeoff weight. An analysis of the feature importance and an ablation study are done, showing which parts of the model are important or not. Some parts of the method might not be useful which is fine given the limited time competition context. Readers might think that these parts are useful. In order to not mislead the readers, the "Detailed Remarks/Questions" section below details which parts might be tagged as not (that) useful. Some details are missing and there is one theoretical paragraph that might not be relevant (see "Detailed Remarks/Questions" below), but given this easy to add/correct, I recommend an accept with minor revision.

3.1.1 Questions Related to "2.2 Data Preprocessing" and "2.3 Feature Engineering"

A1 - ADS-B trajectory data are somewhat "dirty", the article does not mention any "cleaning" process. I looked at `get_traj_features4.R` and did not see any but I am not "fluent" in R and maybe I missed something. Is there really no trajectory cleaning ? Maybe the article could state clearly if there is a cleaning process or not, and it should be described in the article if there is one.

Response

For this study, we did not perform any cleaning process to remove or adjust data observations. However, during the feature engineering process, we tried to mitigate the impact of noisy observations. For example, the feature "maximum altitude" for each flight phase was set as the 99^{th} percentile of altitude values, as we identified unusually high values in some instances. These details have been added in the revised version.

A2 - We do not always have data for the whole trajectory. As a consequence some statistics concerning the efficiency of the vertical profile might not be "accurate". For instance the total time flown

in level flight of two flights might differ only because the ADS-B coverage is not the same. Do you have a way to handle these kind of unbalance due to the ADS-B coverage?

Response

We did not employ any specific pre-processing procedure to address the ADS-B data coverage problem. We should mention, however, that the time series variables, specifically created for the departure phase, were calculated only for the flights with the first observation within 2 nautical miles from the origin, in order to mitigate the impact of the lack of data observed for this phase.

A3 - Concerning the feature taxi_ratio in the duration-based section, it is the taxiing time over the total flight duration, indicating the inefficiency. This ratio is then normalized again by dividing it by the median value to facilitate the comparison across flights. However, looking at `feature_engineering.ipynb`, it is divided by the median of all the flights, hence it is simply a linear transformation of the feature that will have no impact on tree-based algorithms/models (which are invariant to strictly monotone transformation of their feature) and neural network models in which each feature is typically normalized/standardized beforehand. If I did not miss something on that, maybe the paragraph about the division by the median could be removed as this division by the median do not change anything and might misled the readers into believing that it does something.

Response

We appreciate the reviewer's insightful comment. Following the suggestion, we have removed the paragraph about the division by the median, as it does not impact the models and could be misleading to readers.

A4 - Concerning the speed per distance, still in the duration-based section, looking at `feature_engineering.ipynb`, it is actually equal to 1/flight_duration. As such, if I did not miss something, the phrase "This feature provides a normalized speed metric relative to the flight distance, assisting in analyzing speed efficiency over varying distances." in the article should be modified. Furthermore, this transformation is strictly monotone, and hence will have no impact on tree-based algorithms/models, aside from multiplying by 2 the probability to be se considered as a split feature during the tree growing phase.

Response

We thank the reviewer for the observation. In fact, the feature speed per distance corresponds to the inverse of the flight duration. We included this variable because flight duration was stored as integer values, which reduced the precision of the information. Additionally, we wanted to test whether the model would respond better to the inverse of time rather than to the duration itself. That said, we agree that this transformation does not have a significant impact, especially for tree-based models, and the explanation in the text could give the wrong impression. We have, therefore, removed the corresponding comment in the manuscript, as suggested.

A5 - Looking at `get_traj_features4.R`, `feature_engineering.ipynb` and `catboost.ipynb`, for the "first ten points features", some features are not listed in the article despsite being used like the altitude and the sqrt_tas features. I know that the sqrt_tas features are in the end not used as they are eliminated by the catboot selection process, however they could be listed as they were considered in the selection process. As a side note, the "squared" transformation is strictly monotone, and as such tas_1 and sqrd_tas_1 are basically the same feature, and this might have an impact on their respective importance score.

Response

The variable squared airspeed is listed now. We agree with the comment regarding the feature's importance, and one of them could potentially be removed.

A6 - From the "Interaction Features" section alone, the rationale is not clear behind the operation done, how multiplying the airspeed by the specific energy "help us understand how an aircraft's speed influences its energy efficiency during different stages of flight". Could you elaborate on this?

Response

The rationale behind multiplying airspeed by specific energy is to obtain a derived feature that captures how velocity contributes to the total energy state of the aircraft. Specific energy, which combines potential and kinetic energy normalized by weight, provides a measure of an aircraft's maneuverability and efficiency. By incorporating airspeed into this interaction, we aim to analyze how changes in velocity impact the aircraft's energy availability throughout different flight phases. To clarify this point, we have revised the explanation in the manuscript to better illustrate the intuition behind this feature.

3.1.2 Questions Related to "2.4 Predictive Models"

A7 - As a suggestion, you could change the subsection "Predictive Models" to a section. With this change, you will be able to add two subsections to "Articficial Neural Networks", one for the classical and one for SAINT. An other benefit is having a new section "Gradient Boosted Trees" containing the three gradient based methods. This section could contain a description of the hyperparameter that are shared between these methods. As a side note, the description of these hyperparameters is made inside Extreme Gradient Boosting, the last method described.

Response

We are grateful to the reviewer for the suggestion to reorganize the manuscript. Nevertheless, we decided to keep the current organization, following the structure suggested by the journal. As noted by the reviewer, the XGBoost section included descriptions for its hyperparameters, then we decided to also include hyperparameters' descriptions for the other GBDT techniques.

A8 - Even though you did not use optuna to tune the neural networks, maybe you can have hyperparameters tables for the classical and SAINT neural networks. For SAINT, you performed a grid search on the width, maybe you could show this grid of hyperparameters in a table. For the classical neural network, from the text, it is not clear how the hyperparameters were selected. As a side note the batch size is not mentioned in the text despite being an important parameter of the training phase.

Response

We have added detailed hyperparameter information for both the classical neural network (FastAI Tabular learner) and the SAINT model. We have included a new table that summarizes the key hyperparameters for both architectures, including batch size, learning rate, optimizer, network structure, dropout rates, and other relevant settings. For the classical neural network, we have clarified that we used a batch size of 128, Adam optimizer with a learning rate of 0.005, and a network structure of [400, 300, 200] hidden units. For the SAINT model, we have specified the batch size of 2048, embedding size of 32, transformer depth of 6 with 8 attention heads, and dropout rates of 0.1. We have also explained our hyperparameter selection process, noting that we employed a manual tuning approach guided by literature recommendations and empirical observations during preliminary experiments, rather than

automated hyperparameter optimization due to computational constraints. The additional text has been added to the Artificial Neural Networks subsection in the revised manuscript.

A9 - For the hyperparameter tuning of CatBoost, I am not sure that changing the learning rate is a good practice. The obtained hyperparameters are searched with a "big" learning rate, thus the obtained hyperparameter could be sub-optimal when considered together with a reduced learning rate. Maybe, the article could provide figures on compute times to help justify this choice.

Response

Thank you for your insightful comment. Indeed, we expect coupling between the hyperparameters, so a more correct methodology should involve the learning rate as an optimization variable in Optuna. However, considering the limited computational power and the time constraints of the challenge, adopting the described heuristic proved really useful to quickly obtain a good set of hyperparameters for CatBoost. Since the learning rate has such a huge impact on performance, we noticed that the RMSE performance of an Optuna trial was dominated by the learning rate: a model with a lower learning rate converged to a smaller RMSE while at the same time taking much longer to train. We agree that the obtained set will be suboptimal, but we are already dealing with a very complicated optimization problem, so obtaining an actual optimal set cannot be expected, regardless. We consider that the adopted heuristic protocol was very successful in practice since we could reduce the RMSE considerably with respect to the default hyperparameters of CatBoost. To help convince the reader of the importance of the heuristics we adopted, we added the time required to train the final model, as suggested.

A10 - For the hyperparameter tuning of CatBoost, the early stopping is used to train the final model, it is not clear that it was also used in the hyperparameter tuning phase.

Response

Early stopping was used both during the hyperparameter search and final training phases. We added this information to the paper.

A11 - Concerning the ensemble section, I am not sure the theoretical framework presented is consistent with the way the ensemble is built. The theoretical framework presented is typically used in bagging/random forest: the predictive models are just averaged without a "meta-learning" above them that optimize weights using the training set. As the coefficients are actual constants, the variance decomposition is easy to determine.

In the article, the predictive models are combined using (a part of?) the training set to optimize the weights. Furthermore the bias-variance decomposition is the expectation over (at least) a random training set, as a consequence the weights are not constants (w.r.t. the training set) and the variance decomposition in the article cannot be made here, it is not that straightforward anymore. The ensemble method used in the article is probably more related to "stacking" ([1]), and a more relevant theoretical framework can be found in the "Super Learner" paper ([2]).

Response

The framework we initially presented is more relevant to bagging/random forest approaches with simple averaging, while our implementation uses a stacking-like approach with optimized weights. We have rewritten the ensemble section to better align with our methodology. The revised section now clearly identifies our approach as an adaptive ensemble method similar to stacking (Breiman, 1996) and Super Learner (van der Laan, 2007) and removes the variance decomposition discussion that was inap-

propriate for our weighting scheme. We explicitly state that we use a holdout validation set (20% of the original training data) to optimize the ensemble weights and we explain the advantages of weighted ensembles over simple averaging.

A12 - Concerning the ensemble section, it could detail whether the training set used to learn the models is the same used to compute the optimal weights, or is it split in two (or K) parts to do that?

Response

We would like to clarify that the training set used for learning the individual models was not the same as the set used for computing the optimal weights in our ensemble approach. Our implementation follows a two-stage process: First, we trained each individual model (CatBoost, LightGBM, XGBoost, and neural networks) using 80% of the available data (the primary training set). Second, we computed the optimal weights for the ensemble using a separate validation set, which was created by setting aside 20% of the original training data before model training began. This approach is similar to a stacking ensemble method, where the weights are learned on held-out data to avoid potential overfitting that might occur if the same data were used for both tasks. The validation set allows the ensemble to learn how to optimally combine the individual model predictions without being influenced by how well each model performs on its training data. In the final evaluation phase, we retrained all individual models on the entire challenge dataset before applying our pre-computed optimal weights to generate predictions for the submission dataset.

3.1.3 Questions Related to "3 Discussions"

A13 - Concerning the Table 6 for the ablation study, the temporal and geographical features are described in their own subsubsections 2.3.2 and 2.3.4 but features such as "aircraft characteristics" and "operational features" are not described. Maybe the authors could add subsubsections for these two groups. Additionaly, some features in these groups are not described in the feature section.

Response

We have addressed this inconsistency by adding one new subsubsection for Aircraft Characteristics Features. The Aircraft Characteristics Features subsubsection now describes features related to the physical and operational attributes of each aircraft type, including Maximum Takeoff Weight, Operating Empty Weight, maximum payload capacity, fuel capacity, engine specifications, aircraft age category, seating configuration, wing dimensions, and range capability. The Operational Features are related to flight planning, route characteristics, and operational efficiencies, including flight phase durations, rates of climb/descent, cruise altitude variation, speed profiles, level-off segments, flight efficiency metrics, fuel flow proxies, load factor estimates, terminal procedure indicators, and holding patterns. We have included a clarification paragraph in the ablation study section that explicitly maps each feature group mentioned in Table 6 to its corresponding subsubsection in the Feature Engineering section.

3.1.4 Typos

A14 - line 303: "hype parameters" to "hyperparameters".

Response

This has been corrected.

A15 - line 430: In "The A320 was the most used equipment", the word "equipment" sounds weird, maybe "aircraft type" would be better.

Response

We replaced the word "equipment" with "aircraft type".

3.1.5 Appendices

Some thoughts I had reading the article that can be left unanswered by the authors as they might be unimportant, irrelevant or wrong:

A16 - We do not always have data for the whole trajectory. As a consequence the first ten observed points are somewhat random: they are the first points inside the ADS-B coverage. The altitude "frontier" of the ADS-B coverage may vary from one location to an other. As such, the altitude of these first points may vary, regardless of the way the aircraft is operated. To illustrate how this can be problematic, let us consider the feature $VerticalRate_1$ which is the vertical rate computed on the first observed point. If you observe $VerticalRate_1\,(Flight_A) > VerticalRate_1\,(Flight_B)$, maybe $Flight_A$ is lighter than $Flight_B$ or maybe they have a similar weight but $Flight_B$ is higher than $Flight_A$ and consequently have a lower rate of climb. The feature "height relative to the origin airport" can mitigate this issue but the tree based models will have to "chain" two conditions/nodes: one on this altitude and the other on the VerticalRate. Intuitively, this consumes tree nodes (and hence model complexity), and as the tree growing algorithm is a greedy algorithm (it creates one condition/node at a time), it is not certain that the first condition will be selected (because it was the locally optimal choice) as the first condition might be good only if it is followed by the second one.

Response

We believe that the problem of ADS-B coverage impact on trajectory features, as mentioned by the reviewer, is relevant and its mitigation is an important aspect to be further investigated. Nevertheless, we opted to leave it aside for now since it is unclear how to address this issue within the GBDT frameworks used in this work.

A17 - The authors imputed the missing values using the median. I think this is for the best here, but sometimes if the fact they are missing is correlated to the variable to be predicted, it is better to left them as missing values. This way, missing values can be treated separately, and be put on the left or right side of the condition which should help reduce the training loss, see https://github.com/microsoft/LightGBM/issues/2921. In the authors case, the fact that the value is missing is independent of the mass of the airplane, a priori, and hence it should be better to impute them with median. It was just a question that popped in my head during my reading.

Response

We find the insight pointed out by the reviewer interesting. It is something that can also be further investigated in future work.

3.2 Reviewer 2

The authors present an interesting framework for predicting aircraft weight using open data. The manuscript is well-structed and the results are promising. However, I have several concerns regarding the model creation process that require further attention.

M1 - Sec 2.3.1: While the feature selection is important for accurate prediction, some choices are questionable from an aircraft dynamics perspective. For example, the mean temperature might be less informative than the deviation from ISA, which directly reflects atmospheric conditions influencing aircraft performance. Please justify the choice of using mean temperature. In the same way,

the maximum altitude could be more relevant than average altitude for weight prediction.

Response

The maximum altitude for each phase was also used/tested. We acknowledge that the temperature relative to ISA could be tested in future work.

M2 - Line 140: How was the wind information obtained? Is it included in the open data sets?

Response

Yes, it is included in the open datasets.

M3 - Line 157: I do not understand how the authors use the level segments. The manuscript mentions three segments (departure, enroute, and arrival). Is "level segment" a fourth category? Please provide a clear definition and explain its role.

Response

The level segments we refer to are flight segments flown in constant altitude between takeoff until the top-of-climb. "...Additionally, we included information regarding the efficiency of the vertical profile from takeoff until the top-of-climb. We computed the total time (min) and distance flown (NM) in level flight from takeoff until the top-of-climb, as level-off segments tend to generate higher fuel burn. Particularly in the departure phase, these level segments might be more probably associated with structural inefficiencies rather than operational inefficiencies, potentially being accounted for during fuel planning. Moreover, step climbs may correlate with aircraft weight. As fuel is consumed and aircraft mass diminishes, the aerodynamic efficiency at higher altitudes improves, necessitating incremental climbs to optimize fuel consumption. Furthermore, heavier aircraft may exhibit reduced initial climb performance, rendering step climbs a strategic method to achieve optimal flight levels..."

M4 - Line 180: Why were discrete flight durations used instead of continuous values, while most other features are continuous.

Response

We chose to discretize flight duration to help the model recognize patterns in different types of flights, such as short, medium, and long flights. Grouping durations into categories makes it easier to capture operational differences, like fuel usage and air traffic procedures, which might not be as clear when using continuous values. This also helps tree-based models split the data in a way that aligns with real-world flight patterns. While this approach has some trade-offs, we believe it improves interpretability, and we have clarified this reasoning in the manuscript.

M5 - Line 184: What is the "speed per distance"? Speed is time-dependent while flown distance is a single value of the flight. Please clarify the definition.

Response

The feature speed per distance is actually defined as the inverse of the flight duration. Since the flown distance is a fixed value for each flight, this transformation was intended to capture how quickly the flight was completed. We introduced this feature because the original flight duration was stored as integer values, which reduced the precision of the information. Additionally, we wanted to explore whether using the inverse of time might help the model learn certain patterns more effectively than using the duration directly. That said, we agree that the name and description of this feature could be

unclear and potentially confusing. We have decided to remove this reference from the manuscript to avoid any misunderstanding.

M6 - Line 194: The manuscript calculates both "mean vertical speed" and "elevation gradient". These feature seems redundant.

Response

The elevation gradient is the altitude difference between origin and destination airports divided by the actual distance flown. The vertical speed is the altitude variation per time unit.

M7 - Sec 2.3: Sec. 2.3 lists various variables, but it is unclear which ones are finally used. You should provide a complete list of selected features and their count.

Response

We thank the reviewer for the comment. The list of variables used in our analysis is indeed extensive, and including a full list directly in the manuscript would be impractical and potentially overwhelming for the reader. To address this, Section 2.3 was structured into multiple parts, each highlighting a specific group of variables and the corresponding preprocessing steps. This approach helps the reader understand the types of transformations applied during data preparation. Additionally, the notebook `feature_engineering.ipynb` contains detailed comments, explanations, and frequent printouts that clearly indicate which variables are being used at each stage. To clarify this point, we have added a note at the beginning of the section in the manuscript, referring readers to the code for the complete list of features.

M8 - Line 258: Please provide references for FastAI's Tabluar learner and SAINT model.

Response

We have added the following citations to the manuscript: For FastAI's Tabular learner: Jeremy Howard and Sylvain Gugger. "fastai: A Layered API for Deep Learning". In: CoRR abs/2002.04688 (2020). arXiv: 2002.04688. url: https://arxiv.org/abs/2002.04688. For the SAINT model: Gowthami Somepalli, Micah Goldblum, Avi Schwarzschild, C. Bayan Bruss, and Tom Goldstein. "SAINT: Improved Neural Networks for Tabular Data via Row Attention and Contrastive Pre-Training". In: CoRR abs/2106.01342 (2021). arXiv: 2106.01342. url: https://arxiv.org/abs/2106.01342.

M9 - Sec 2.4.1: More details about ANN structures are needed, such as the number of nodes, layers, etc. Additionally, please specify the hyperparameters and their chosen values?

Response

We have added detailed information about the neural network architectures. For the FastAI Tabular learner, we specified a structure with hidden layers [400, 300, 200], while SAINT used a transformer depth of 6 with 8 attention heads. We have included a new table with comprehensive hyperparameter details for both models, including batch sizes (128 for FastAI, 2048 for SAINT), learning rates (0.005 and 0.002, respectively), optimizer choices, dropout rates, and early stopping criteria. We also explained our hyperparameter selection approach, which combined literature recommendations with empirical testing within our computational constraints.

M10 - Figure 9 - This figure suggests that the distribution varies across aircraft types. This infers that the estimation model tries to improve the accuracy of yellow aircraft rather than green aircraft.

It would be beneficial to calculate the RMSE separately for each type. Relying solely on a single overall RMSE might mask performance discrepancies.

Response

We have added Table 7 to present the RMSE values by aircraft type. Indeed, the predictive performance varies significantly, with larger aircraft exhibiting higher RMSE.

M11 - Table 5: Table 5 provides RMSE values for different algorithms. Including a baseline comparison, such as RMSE using typical weights in each aircraft type/range or just FAA_Weight, would provide valuable context for evaluating the model performance. Also, it is important to discuss the performance compared to existing methods in the literature.

Response

We have incorporated these comparisons into Section 3.2, which now includes RMSE and MAPE values for a baseline model that outputs the MTOW along with the results of our proposed algorithms.

M12 - Table 6: The list of ablated components does not align with the sections in 2.3-2.4, which makes it difficult to understand which features are grouped together. Please clarify the mapping between components and feature categories.

Response

We have addressed this concern by adding one new subsubsection for Aircraft Characteristics Features. Furthermore, we have added a clarification paragraph in the ablation study section that provides a clear mapping between each ablated component in Table 6 and its corresponding feature category in each Section 2.3.

3.3 Reviewer 3

The paper presents one of the winning solutions to the PRC data challenge. The description and explanation of the paper are very thorough, leaving almost no stone unturned. I recommend accepting the paper with minor revisions.

S1 - It would be nice to mention the data source, Eurocontrol PRU, in the abstract of the paper.

Response

The data source was included in the abstract.

S2 - Line 13: Abbreviations can be removed or updated (these are examples from the template).

Response

They were removed.

S3 - I suggest including a table that provides an overview of all features used for training the model in section 2.3.

Response

We thank the reviewer for the suggestion. However, due to the large number of features used in the analysis, including a full table listing all of them in the manuscript would be impractical. Instead, we chose to organize Section 2.3 into thematic subsections that describe the process used to construct and transform these features. This structure allows the reader to understand how each group of variables was derived and used. Additionally, for the interested reader, the complete code used for feature engineering is available in the public repository linked at the end of the manuscript, including comments, explanations, and intermediate outputs to facilitate understanding. We have added a note in the text to inform readers about this.

S4 - Structurally, I think section 3.1 on exploratory data analysis could be moved earlier, perhaps after section 2.2. However, this is a personal preference.

Response

We appreciate the reviewer's suggestion. Since Section 2 was already quite lengthy, we decided to keep the exploratory data analysis in Section 3.

S5 - In section 3.4, it would be helpful to provide more background on the "ablation study," such as what it is and how it provides more insights into the models.

Response

In the revised manuscript, we have added a detailed explanation at the beginning of Section 3.4 to introduce this methodology. The added text explains that an ablation study is a systematic experimental procedure used in machine learning to evaluate the contribution of different components in a complex model by removing or "ablating" specific parts and measuring the resulting change in performance. We clarify that this approach helps identify which elements are most crucial to the model's effectiveness, quantifies the relative importance of each component, guides future model refinements, and enhances interpretability. The explanation connects this general methodology directly to our specific analysis of feature groups and model components in ATOW estimation, using RMSE as the evaluation metric. This addition provides readers with the necessary context to better understand the purpose and value of the ablation results presented in this section.

S6 - The colors and labels in Figure 10 are a bit hard to see. Is there a better way to visualize the results? Perhaps by creating a subplot for each aircraft type?

Response

We replaced the figure with a cleaner version that shows the scatterplot of ATOW and distance flown for the most frequent aircraft types within each wake turbulence category (medium turboprop, medium jet and heavy jet).

S7 - Finally, please create a permanent archive with DOI using Zenodo (or a similar service) for the GitHub repository. This way, the code is archived even when GitHub is no longer accessible.

Response

We added our code to Zenodo, as requested by the reviewer. You can find it at: https://doi.org/10.5281/zenodo.15069706.

References

[1] Leo Breiman. “Stacked regressions”. In: *Machine learning* 24 (1996), pp. 49–64.
[2] Mark J Van der Laan, Eric C Polley, and Alan E Hubbard. “Super learner”. In: *Statistical applications in genetics and molecular biology* 6.1 (2007).