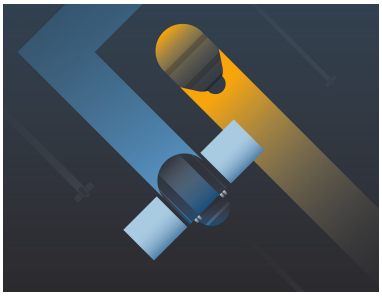
PREDICTION OF COLLISION RISKS FOR SPACE OBJECTS WITH CONJUNCTION DATA MESSAGES USING NEURAL NETWORKS

FREDERIK DALL'OMO, LEO PAUL, LUKAS WALLNER



APPLIED MACHINE LEARNING FOR ENGINEERS
REPORT
-
University of Stuttgart

February 8, 2024

Contents

1 Introduction

This report serves as a comprehensive documentation of the project. More detailed information on the code can be found within the comments provided in the code itself. It meticulously incorporates the principles and concepts covered during the "Machine Learning for Engineers" course at the University of Stuttgart, effectively translating them into practical application.

The introduction covers the background (cf. Section 1.1), problem statement (cf. Section 1.2), objective (cf. Section 1.3) and Scope (cf. Section 1.4) of the project.

1.1 Background

Satellite collisions pose a significant threat to space operations, with the increasing number of objects orbiting Earth. The European Space Agency (ESA) actively engages in collision avoidance activities, supported by the Space Debris Office. To address the challenges of predicting collision risks, ESA's Advanced Concepts Team organized the Collision Avoidance Challenge[1]. This competition seeks to develop models capable of predicting collision risks between satellites and space objects, leveraging real-world Conjunction Data Messages (CDMs) provided by ESA.

1.2 Problem Statement

The challenge stems from the necessity to accurately predict collision risks between satellites and other space objects. With the current volume of alerts and close encounters, space operations demand validated and timely data. The task at hand involves building a model that can predict the final collision risk estimate between a given satellite and a space object based on historical CDMs. This prediction is crucial for informing collision avoidance maneuvers, as the final risk estimation is derived from the last available CDM prior to the close approach.

1.3 Objectives

The primary objectives of this competition are:

1. Develop predictive models that can accurately estimate the final collision risk between satellites and space objects.
2. Utilize the provided dataset of real-world CDMs recorded by ESA's Space Debris Office.

1.4 Scope

The scope of this competition aligns with the "Applied Machine Learning for Engineers" course at the University of Stuttgart, serving as an exercise to apply machine learning techniques to a real-world problem. Within the context of the course, the focus is applying machine learning concepts and methodologies, including data exploration, feature selection, and modeling techniques covered in the lecture.

[1] The competition details and dataset can be found on the official website of the Collision Avoidance Challenge [8]

2 Data

2.1 Data Description

The dataset used in this competition consists of Conjunction Data Messages (CDMs) provided by the European Space Agency (ESA). Each CDM represents a recorded event involving the close approach of satellites and space objects. The dataset contains a total of 103 recorded characteristics/features for each CDM, providing comprehensive information about the events.

2.2 Exploratory Data Analysis (EDA)

Exploratory Data Analysis (EDA) is conducted based on the code provided, focusing on the following aspects[2]:

- **Visualization:** Utilization of visualizations, such as scatter plots, box plots, and correlation matrices, as generated in the code, to illustrate relationships between variables, explore risk distribution (see Figure 1), and identify potential patterns.

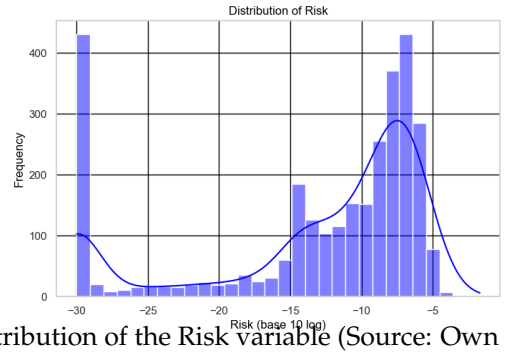


Figure 1: Distribution of the Risk variable (Source: Own representation)

- **Filtering and Cleaning:** Application of proposed filters, including the requirement of at least two CDMs per event and temporal constraints, as implemented in the code[3][9]. A comparison of the unfiltered and filtered data is displayed in Figure 2 and 3.

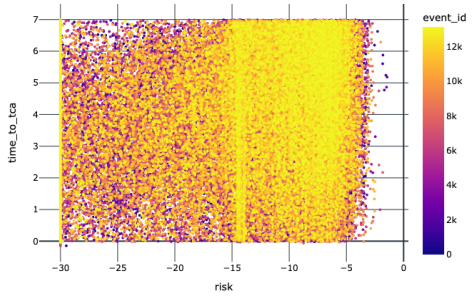


Figure 2: Scatter plot with unfiltered data (Source: Own representation)

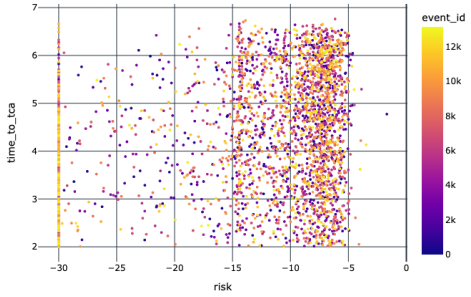


Figure 3: Scatter plot with filtered data (Source: Own representation)

- **Handling Missing Values:** Identification and visualization of missing values within the dataset, as addressed in the code, for further consideration during preprocessing.

[2]https://github.com/freddida/MLProject/blob/main/notebooks/exploratory/visualization.ipynb
[3]https://github.com/freddida/MLProject/blob/main/src/utils/filtering.py

3 Feature Selection

3.1 Initial Feature Set

The initial feature set consists of 103 characteristics, including both numerical and categorical variables that provide comprehensive information about each recorded event. As there are some missing values, declared as 'Not a Number' (NaN), the selection process is executed twice. The first attempt drops[4] the NaN's and the second imputes[5] them using k-Nearest Neighbors [4]. Twelve features are shared between both methods. A model can be trained using both sets for comparison but for simplicity only the features without NaN imputing are used.

3.2 Methods of Feature Selection

For visualization and more information on the methods refer to the jupyter-notebook[6].

3.2.1 Univariate Selection

Univariate selection evaluates the importance of a set of k features together. It utilizes the `SelectKBest` class of scikit-learn [7] class with the `mutual_info_regression` [6] scoring function to select the top features based on their mutual information with the target variable [1].

3.2.2 Feature Importance with ExtraTreesRegressor

Feature importance using `ExtraTreesRegressor` involves training an ensemble of decision trees and assessing the importance of each feature in predicting the target variable. By analyzing the feature importances, we can identify the most informative features that contribute significantly to the model's predictive performance.

3.2.3 Correlation Matrix

The correlation matrix examines the linear relationship between features and the target variable. Features with high correlation coefficients indicate a strong association with the target variable and are considered important for prediction. By analyzing the correlation scores, we can identify key features that influence the risk of collision in space operations.

3.2.4 D-Wave scikit-learn Plugin using a Quantum-Classical Hybrid Solver

The D-Wave scikit-learn Plugin leverages quantum-classical hybrid solvers to select features by solving a quadratic optimization problem. This approach provides an alternative method for feature selection, particularly suited for complex optimization tasks. By incorporating quantum computing capabilities, it offers potential benefits in handling large-scale feature selection problems efficiently.

[4] https://github.com/freddida/MLProject/blob/main/notebooks/engineering/feature_selection_without_nan.ipynb
[5] https://github.com/freddida/MLProject/blob/main/notebooks/engineering/feature_selection_with_nan.ipynb
[6] https://github.com/freddida/MLProject/tree/main/notebooks/engineering

3.3 Final Feature Set

The final feature set comprises the selected features (16 out of 103) by combining the different methods and removing duplicates. For that the elbow method (usually used in clustering) was deployed for each method selecting the top n features.

4 Modeling

4.1 Data Splitting

The data is split into training, evaluation, and testing sets. The training set comprises 64% of the data, while the test set contains 20% and the evaluation set 16% of the entire data set. Stratified splitting is used to ensure proportional class distribution. For that the risk values are converted into binary classes based on the threshold specified in the challenge (high final risk: $r \geq 10^{-6}$, low final risk: $r < 10^{-6}$).

4.2 Over Sampling

As shown in Figure 4 the high/low risk distribution is unbalanced (percent High Risk: 7.33%). To address this class imbalance the Synthetic Minority Over-sampling Technique (SMOTE) is applied [2]. It works by generating synthetic samples for the minority class, thereby balancing the class distribution. SMOTE selects minority class instances and generates synthetic samples by interpolating between these instances and their nearest neighbors. Figure 5 shows the sampled training dataset.

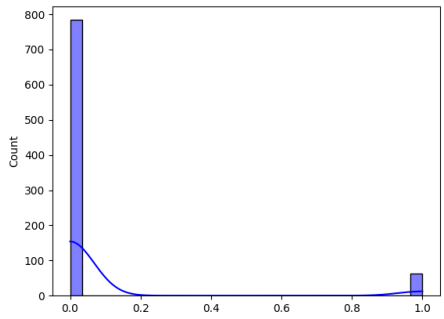


Figure 4: High/low risk distribution of training set, Percent High Risk: 7.33% (Source: Own representation)

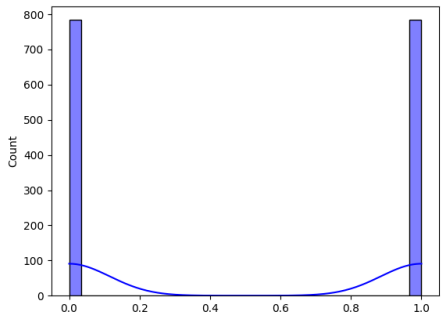


Figure 5: High/low risk distribution of resampled training set, Percent High Risk: 50% (Source: Own representation)

4.3 Scaling

Scaling is applied to prevent features with larger scales from dominating the learning process. For that the MinMaxScaler, which scales features to a range between 0 and 1 is applied to all feature sets [5].

4.4 Model

The chosen model for this project is a feedforward neural network. The neural network architecture consists of an input layer, one hidden layer, and an output layer. The input layer has 124 neurons with ReLU activation, followed by the hidden layer with 64 neurons also using ReLU activation. The output layer consists of a single neuron and a sigmoid activation function.

4.5 Training Process

The model is trained using the Stochastic Gradient descent (SGD) optimizer with a learning rate of 0.01, momentum of 0.9 and binary cross-entropy loss function. The training data is fed to the model in batches, and the model's parameters are updated iteratively to minimize the loss function. The validation data is used to monitor the model's performance and prevent overfitting.

4.6 Prediction Process

The best threshold for converting risk predictions to binary prediction is based on maximizing the F-beta score. A threshold of 0.75 is considered here for evaluation.

5 Results and Conclusion

5.1 Model Performance

For evaluation of the model performance the F-beta score is taken. This is a measure of predictive performance. The F-beta score with the beta parameter set to 2, weighs recall more than precision. Recall is the ratio of true predicted positives to the number of positive cases in the data. Precision is a measure of correctness of prediction. In the given problem, impetus is on predicting all the positive or high collision risk cases than on penalising false positives as is the case with precision.

The different plots in Figure 6 - 11 show the loss and accuracy over 30 epochs of three different training datasets (all 103 features, 16 selected, 6 selected).

Figure 6: Loss with with 16 selected features (Source: Own representation)

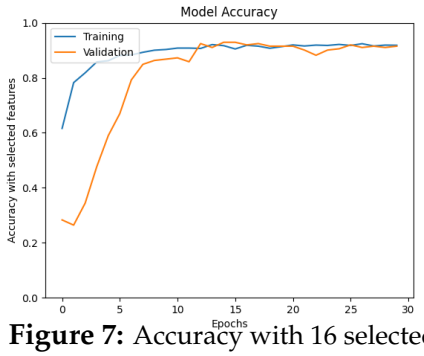


Figure 7: Accuracy with 16 selected features (Source: Own representation)

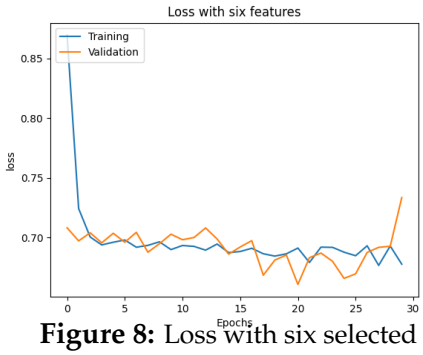


Figure 8: Loss with six selected features (Source: Own representation)

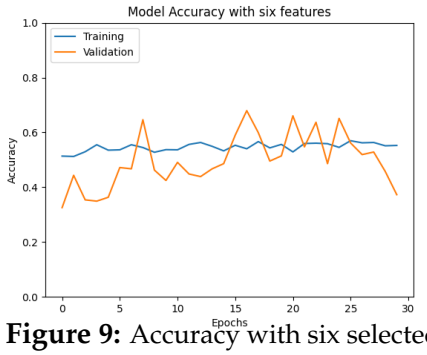


Figure 9: Accuracy with six selected features (Source: Own representation)

The dataset showcases its best performance (cf. Figure 6, 7), when utilizing 16 carefully chosen features (cf. Section 3), yielding an F-beta score of 0.736. However, reducing the feature set to six results in a significant drop in accuracy (cf. Figure 8, 9), with the F-beta score dropping to 0.3. This decline can be attributed to the loss of crucial information caused by feature reduction.

Comparatively, the original dataset containing all 103 features achieves a commendable F-beta score of 0.688, while maintaining a similar accuracy to that of the 16-feature subset (cf. Figure 10, 11). This phenomenon suggests a potential issue of overfitting in the model. It aligns with the concept of the curse of dimensionality in machine learning, which posits that as the number of features increases, the amount of data required for effective modeling rises exponentially [3]. For a dataset with 103 features, approximately 10,000 instances are needed, yet only 800 instances for training are available, potentially contributing to the observed discrepancy.

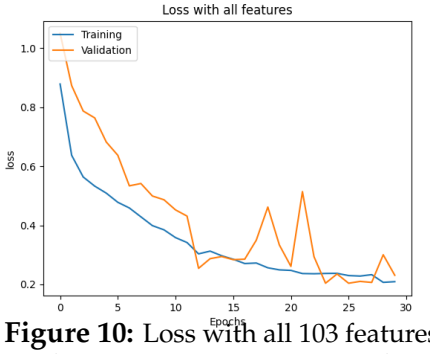


Figure 10: Loss with all 103 features (Source: Own representation)

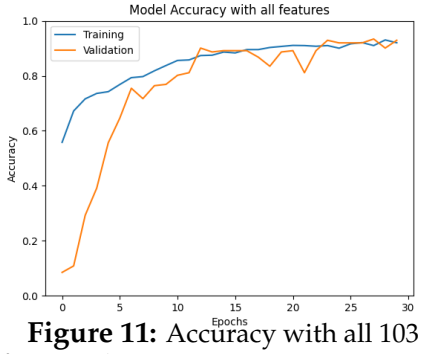


Figure 11: Accuracy with all 103 features (Source: Own representation)

5.2 Conclusion

In summary, the study represents an important initial step towards addressing the complexity of the challenge at hand. While the results show promising advancements, direct comparisons to existing findings are challenging due to the intricacies of the task. Due to time and complexity restriction the scoring was only focused on the F-beta score instead of an evaluation using the Mean Sqaured Error (MSE) divided by the F-beta score. Therefore a comparison to other groups is not possible. Further research and refinement are necessary to establish robust benchmarks for comparison and to deepen our understanding of the problem.

References

[1] D, K. Optimizing performance: Selectkbest for efficient feature selection in machine learning. *Medium* (2023). Accessed: February 16, 2023.

[2] FERNÁNDEZ, A., GARCÍA, S., HERRERA, F., AND CHAWLA, N. V. Smote for learning from imbalanced data: Progress and challenges, marking the 15-year anniversary. *Journal of Artificial Intelligence Research 61* (04 2018), 863–905. Submitted 06/17; published 04/18.

[3] KARANAM, S. Curse of dimensionality — a "curse" to machine learning. *Towards Data Science* (Aug 2021). Published in Towards Data Science, 4 min read, 180 claps.

[4] PEDREGOSA, F., VAROQUAUX, G., GRAMFORT, A., MICHEL, V., THIRION, B., GRISEL, O., BLONDEL, M., PRETTENHOFER, P., WEISS, R., DUBOURG, V., VANDERPLAS, J., PASSOS, A., COURNAPEAU, D., BRUCHER, M., PERROT, M., AND DUCHESNAY, E. Scikit-learn: Machine learning in Python - knnimputer. *Journal of Machine Learning Research 12* (2011), 2825–2830.

[5] PEDREGOSA, F., VAROQUAUX, G., GRAMFORT, A., MICHEL, V., THIRION, B., GRISEL, O., BLONDEL, M., PRETTENHOFER, P., WEISS, R., DUBOURG, V., VANDERPLAS, J., PASSOS, A., COURNAPEAU, D., BRUCHER, M., PERROT, M., AND DUCHESNAY, E. Scikit-learn: Machine learning in Python - minmaxscaler. *Journal of Machine Learning Research 12* (2011), 2825–2830.

[6] PEDREGOSA, F., VAROQUAUX, G., GRAMFORT, A., MICHEL, V., THIRION, B., GRISEL, O., BLONDEL, M., PRETTENHOFER, P., WEISS, R., DUBOURG, V., VANDERPLAS, J., PASSOS, A., COURNAPEAU, D., BRUCHER, M., PERROT, M., AND DUCHESNAY, E. Scikit-learn: Machine learning in Python - mutual info regression. *Journal of Machine Learning Research 12* (2011), 2825–2830.

[7] PEDREGOSA, F., VAROQUAUX, G., GRAMFORT, A., MICHEL, V., THIRION, B., GRISEL, O., BLONDEL, M., PRETTENHOFER, P., WEISS, R., DUBOURG, V., VANDERPLAS, J., PASSOS, A., COURNAPEAU, D., BRUCHER, M., PERROT, M., AND DUCHESNAY, E. Scikit-learn: Machine learning in Python - select kbest. *Journal of Machine Learning Research 12* (2011), 2825–2830.

[8] TEAM, A. C. Collision avoidance challenge, 2021. Accessed: February 8, 2024.

[9] URIOT, T., IZZO, D., SIMÕES, L. F., ET AL. Spacecraft collision avoidance challenge: Design and results of a machine learning competition. *Astrodynamics 6* (June 2022), 121–140.