Automatic Feature Engineering Tool

Daniel Goldstein and Nir Akay

March 12, 2025

1 Abstract

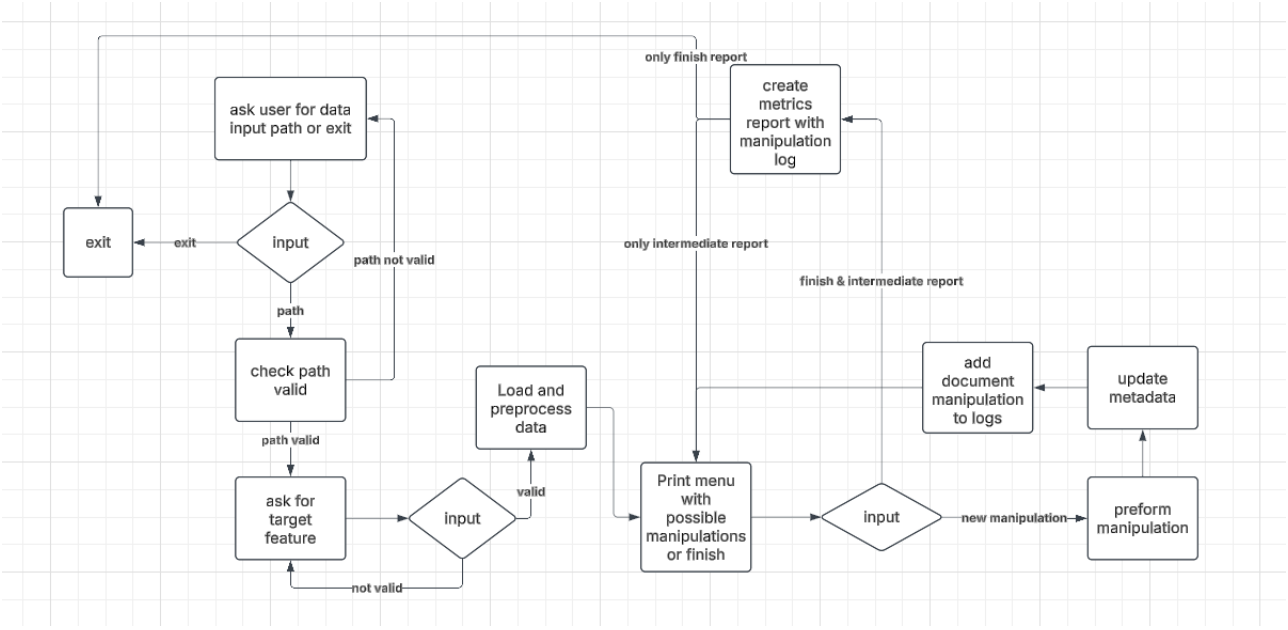
In this project, we will try to improve the r2 parameter and reduce the error parameters for numerical predictions. We will also try to improve thee the accuracy and precision for categorical predictions.
- Mean Absolute Percentage Error (MAPE): $\frac{\sum|y-\text{pred}|}{ny}$
- Mean Absolute Error (MAE): $\frac{\sum|y-\text{pred}|}{n}$
- Root Mean Squared Error (RMSE): $\sqrt{\frac{\sum(y-\text{pred})^2}{n}}$
- Max Error
We will try to achieve this goal by performing feature engineering on the dataset. We will give the user the ability to perform some changes on the columns of the dataset that will help him to achieve a better performance of his model. For Example in one of our improvements we raise the r2 from 0.220 to 0.495 and reduce Max Error by 70% only using our tool.

2 Problem description

We are trying to improve the "Data Cleaning and Preprocessing" element of the DS pipeline. During our first project, we noticed how diverse datasets can be. We imagined that performing the same operations on a lot of datasets can be tedious and we wanted to generate a solution that will help eliminate repeating tasks.

3 Solution Overview

Our solution was made using a notebook, because this is the tool we saw in class and the tool that is mainly used by data scientists. Below is a general diagram of the flow for our solution.



We first of all load the data and try to classify it to one of four categories:
- numerical
- categorical
- date time
- categorical list (in the code referred as categorical commas)
The first three are obvious, so let's show an example for the fourth type. In our first project we used a data set that had a column for genres. A show can have many genres, the genres themselves are of course categorical, but the combinations can be many, and they may not represent the similarities between shows that share some of the categories. I.e. a show that is Action and Fantasy will be closer in it's features to a show that is Action Sci-Fi than a Drama show. So We don't want to use it as a only categorical feature. Tagging this type of features will allow us to split the list into binary features for each category in the list.

The second step is handling nulls. For each type of data we will offer a different type of null handling. Here also we took inspiration from the methods shown in class (like filling with average) and the internet.
For numerical features we will offer:
- Remove rows with nulls, Replace with average, Replace with median and Replace with a custom value

For categorical features we will offer:
- Remove rows with nulls, Replace with most common value, Replace with a custom value
For datetime features we will offer:
- Remove rows with nulls, Replace with the earliest date in the column, Replace with the most common date
For categorical lists features we will offer:
- Remove rows with nulls, Replace with 'Unknown' category, Replace with an empty list
After this step we will choose a target feature (only numeric or categorical) and create a basic prediction for it. This will be our baseline that we will compare our other improvements to. Now we can start performing some feature engineering! After the prediction a menu with all the methods that we offer will appear and we can choose what operation we would like to perform. The list of operations is:
- Outliers reduce, for numerical features
- Log transformation, for numerical features
- Normalize feature, for numerical features
- Boolean features, for categorical list features
- Feature binning, for numerical features
- Hyper parameters tuning
- Days since today, for date time features
After each operation we will create a new prediction, and print some metrics and a graph. The metrics and the graph will be saved to a new folder under the execution folder. This, combined with the log that will be printed in the end, will allow us to understand the whole flow and the effectiveness of each step.
Additionally there is another option in the menu to run some analysis, performing analysis won't create a prediction file.
When we are done, we will type exit instead of choosing an operation, this will create a log that will describe all of our actions, so we could reproduce our actions if needed.

4 Experimental evaluation

case study - anime data set (Link) - target feature is Episodes:
null handling:
Step 1: Replaced null values in feature 'Score' with median (6.39)
Step 2: Removed rows with null values from feature 'Episodes'

Step 3: Replaced null values in feature 'Rank' with median (9930.00)
Step 4: Replaced null values in feature 'Scored By' with custom value (30038)
Step 5: Replaced null values in feature 'Genres' with an empty list

Additional steps:
Step 7: Reduced outliers in feature 'Episodes' using 3 standard deviations
Step 8: Binned feature 'Favorites' into 50 bins
Step 9: Applied log transformation to feature 'Scored By'

analysis:

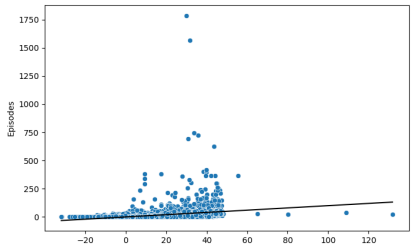


Figure 1: Basic results

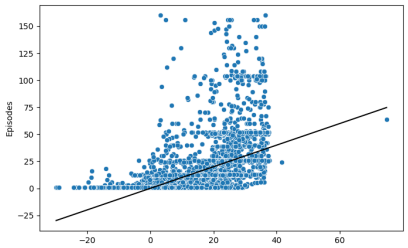
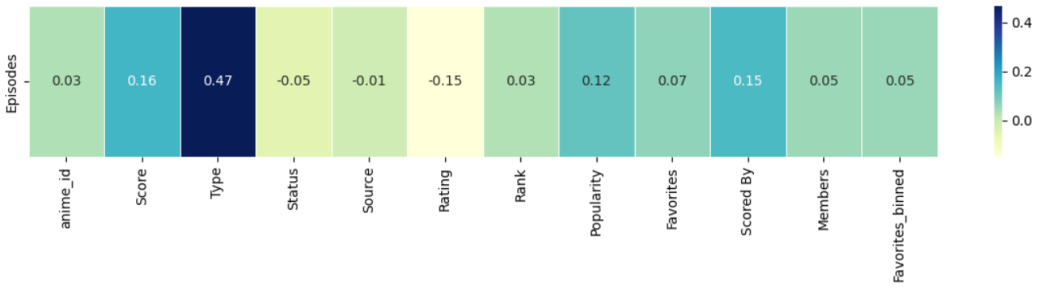


Figure 2: Final results

r2 approved from 0.104 to 0.335
MAPE reduced from 5.755 to 4.347
MAE reduced from 15 to 11
RMSE reduced from 44 to 18
Max Error reduced from 1,757 to 157
As we can see in this pearson diagram:



The numerical features have almost no effect on our target feature and in general the correlations are low. So the most effective operation was to reduce the

outliers.

case study - Space Missions Dataset (Link) - target feature is Mission Success(%):

We applied those steps:
Step 1: Initialized linear model with parameters: 'fit intercept': True
Step 2: Reduced outliers in feature 'Distance from Earth (light-years)' using 2 standard deviations
Step 3: Applied log transformation to feature 'Mission Cost (billion USD)'
Step 4: Applied log transformation to feature 'Payload Weight (tons)'
Step 5: Normalized feature 'Fuel Consumption (tons)' with mean 2543.52 and std 1492.96
Step 6: Binned feature 'Scientific Yield (points)' into 75 bins
Step 7: Added 'Launch Date days since today' based on 'Launch Date'

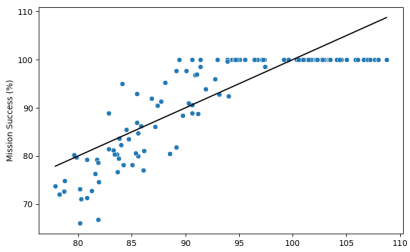


Figure 3: Basic results

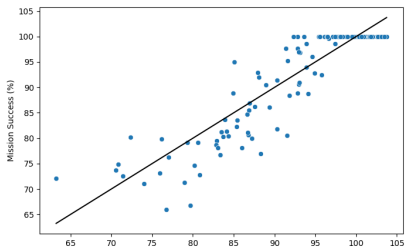


Figure 4: Final results

analysis:
r2 approved from 0.721 to 0.829
MAPE reduced from 0.049 to 0.037
MAE reduced from 4 to 3
RMSE reduced from 5 to 4
Max Error reduced from 15 to 13

most of the improvement we see yields from making log on the 'Mission Cost (billion USD)', 'Payload Weight (tons)' features and that's because they are high correlated (0.85, 0.84) with the target feature. Also we see another small improvement (0.01) when we binned the 'Scientific Yield (points)'.

case study - US Cars Dataset (Link) - target feature is price:

We applied those steps:
Step 1: Initialized linear model with parameters: 'fit intercept': True
Step 2: Reduced outliers in feature 'price' using 2 standard deviations
Step 3: Tuned linear model with parameters: 'fitintercept': False
Step 4: Binned feature 'mileage' into 100 bins
Step 5: Binned feature 'lot' into 100 bins
Step 6: Reduced outliers in feature 'price' using 2 standard deviations

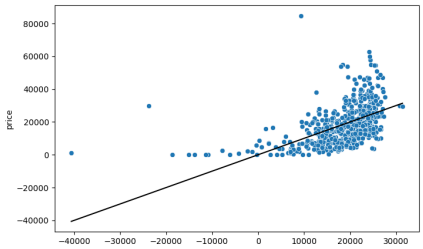


Figure 5: Basic results

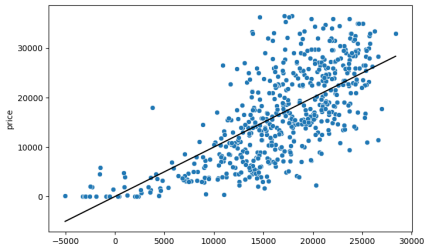


Figure 6: Final results

analysis:
r2 approved from 0.220 to 0.495
MAPE reduced from 145E15 to 96E15
MAE reduced from 7,676 to 5,334
RMSE reduced from 10,374 to 6,679
Max Error reduced from 75,512 to 21,676

One the improvement we see yields from making binning on the 'mileage', 'lot' features and that's because they are relatively highly correlated (-0.40, 0.16) with the target feature. Also we see another great improvement when we remove outliers of the target feature 'price'.

case study - Fires from Space: Australia (Link) - target feature is confidence:

We applied those steps:
Step 1: Initialized linear model with parameters: 'fit_intercept': True

Step 2: Reduced outliers in feature 'brightness' using 3 standard deviations
Step 3: Binned feature 'frp' into 50 bins
Step 4: Applied log transformation to feature 'brightness'
Step 5: Binned feature 'bright_t31' into 100 bins
Step 6: Normalized feature 'frp' with mean 44.46 and std 60.84
Step 7: Added 'acq_date_days_since_today' based on 'acq_date'

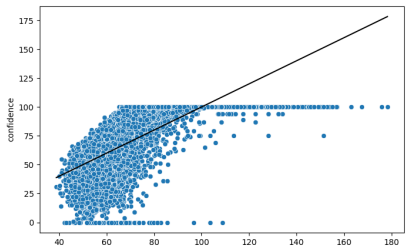


Figure 7: Basic results

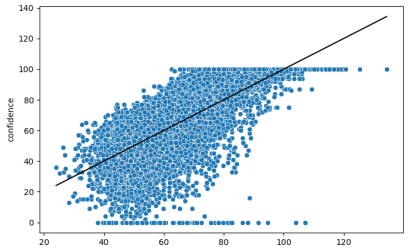


Figure 8: Final results

analysis:
r2 approved from 0.462 to 0.568
MAPE increased from 2,213,502,488,067,951.250 to 2,314,906,822,587,905.000
MAE reduced from 13 to 11
RMSE reduced from 17 to 15
Max Error reduced from 109 to 107

What helped the most was removing outliers and feature binning. because we performed them on features that are highly ocrrelated with our target features

5 Related work

Several existing tools and techniques aim to automate feature engineering, including AutoFE, BigFeat, and getML. These tools streamline feature generation but often lack flexibility and user control. AutoFE and BigFeat primarily focus on fully automated transformations, limiting the ability to manually refine features, while getML is specialized for relational data rather than general tabular datasets. Our approach differs by offering an interactive feature engineering process where users can apply, tweak, and evaluate transformations in real-time,

balancing automation with interpretability. For example "The ultimate goal of AutoFE is to eliminate knowledge and expertise required to build state-of-the-art models for any dataset. In this paper, we focus on automatically generating and selecting a set of useful features." This is a cite from AutoFE that shows they are focusing on building feature engineering without any knowledge but what if the data scientist can achieve better performance by analyzing the data and achieve better performance.

Our solution draws inspiration from these existing tools but aims to improve user control and adaptability. Unlike AutoFE and BigFeat, which act as black-box systems, we provide transparency in feature selection and transformation. By integrating user-guided decisions with automated suggestions, we enhance feature engineering efficiency while maintaining interpretability, filling a gap that current solutions overlook.

6 Conclusion

To sum up our research we can see that by our suggestion of feature engineering we are improving our prediction's model and reduce the Error We learned from this project that not all of our feature engineering works all the time even if it looks like it is going to improve and even in some cases make the model's prediction worse so we need to select the feature engineering wisely.