Final Project of Supervised Learning

Alessandro Mecchia

December 2024

1 Dataset Analysis and Preparation

1.1 Dataset Exploration

The dataset contains 5908 instances, all filled with non-zero values, ensuring it is large enough for meaningful analysis. The features include categorical variables like *Manufacturer*, *Machine Model*, and *Fuel Category*, alongside continuous variables. Histograms and boxplots indicate that while some numerical features have outliers, they represent valid observations reflecting the variability in machinery performance. Features with high correlations to the target variable suggest good predictive potential.

1.2 Encoding and Scaling

Categorical features were grouped based on their cardinality. Rare categories were consolidated to reduce complexity and avoid overfitting. Scaling was applied to continuous features with near-normal distributions to maintain consistency without removing valid data points.

1.3 Binning

Features such as *Engine Capacity (L)* and *Engine Cylinders* were binned to reduce complexity and capture trends in the data more effectively.

1.4 Dimensionality Reduction

Feature selection was performed comparing the results of the ANOVA-test and Recursive Feature Elimination (RFE) for linear models, while a Random Forests classifier was used for non-linear models. This ensured the most relevant features were retained for each type of model.

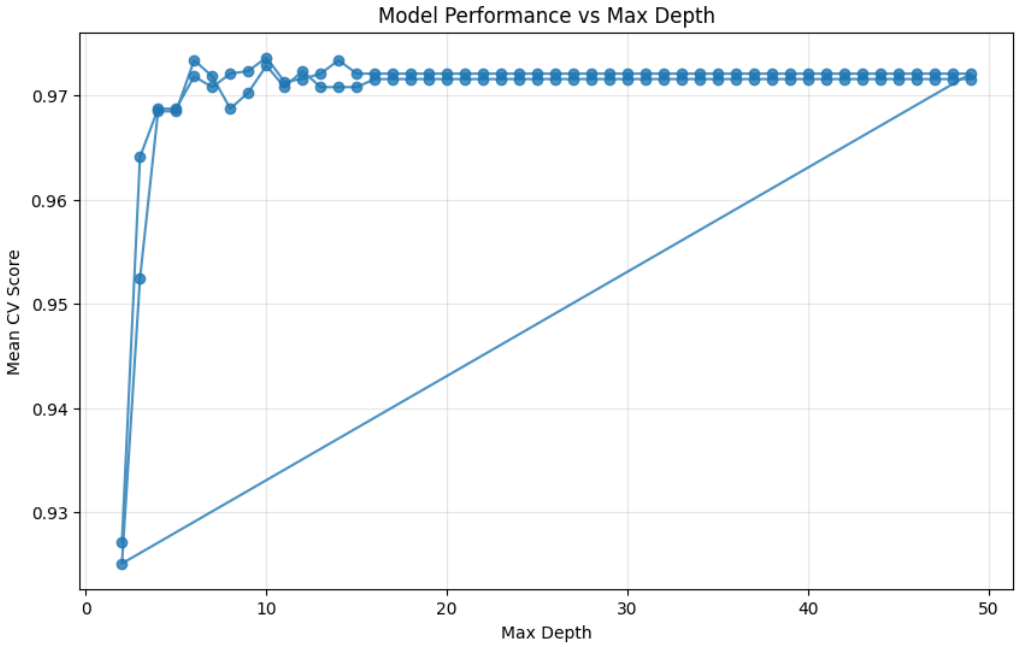
2 Model Implementation

2.1 Softmax Regression

Softmax regression achieved 97% accuracy, with strong performance across all classes except *very low*, which had slightly lower recall (84%). The confusion matrix showed minimal errors, confirming its reliability.

2.2 Decision Tree

The decision tree achieved 98% accuracy after tuning its depth to prevent overfitting. Most classes were accurately predicted, with only minor misclassifications between *high* and *medium*.



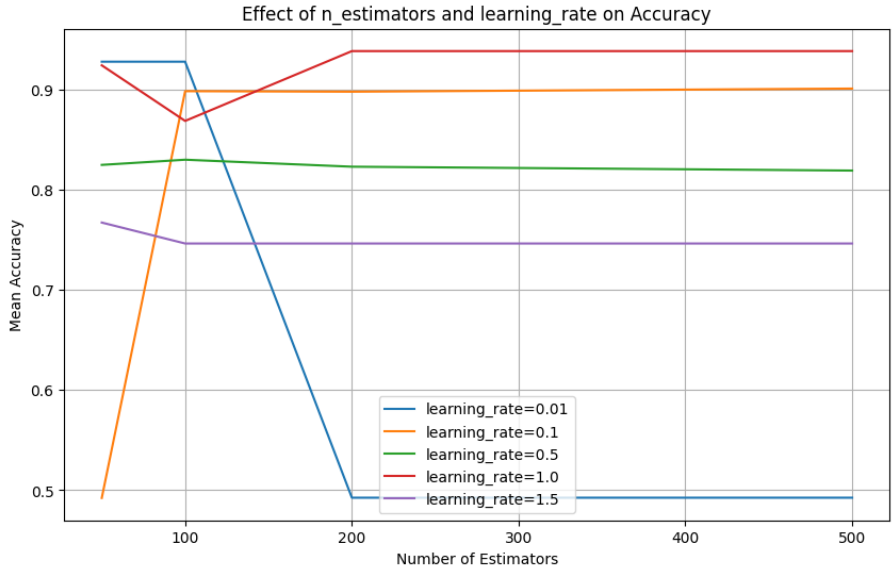
The graph above shows how the model's mean cross-validation score changes with different max depths. Performance quickly rises and settles around 0.97 when max depth is between 5 and 10. Beyond that, increasing max depth doesn't bring much improvement, indicating the model has already reached its optimal performance.

2.3 Random Forest

Random Forest also achieved 98% accuracy, with high feature importance attributed to *Fuel Consumption* and *Average Fuel Efficiency*. The Out-of-Bag (OOB) score validated its performance, and the confusion matrix highlighted strong predictions across all classes.

2.4 AdaBoost

AdaBoost achieved 94.5% accuracy, slightly lower than other models. It performed well for *high* and *low*, but *medium* and *very low* had lower recall and precision, respectively. Tuning the learning rate improved its stability.



This chart above compares how accuracy changes as we vary the number of estimators and learning rate. A moderate learning rate (e.g., 0.1 or 1.0) tends to produce higher, more stable accuracy. Extremely low (0.01) or very high (1.5) learning rates show dips or plateaus in performance. The best learning rate trough the time is as expected 0.1

2.5 Soft Voting Classifier

The Soft Voting classifier combined Logistic Regression, Random Forest, AdaBoost, and Decision Tree, achieving 98% accuracy. **This ensemble approach effectively reduced model-specific weaknesses and improved overall performance**.

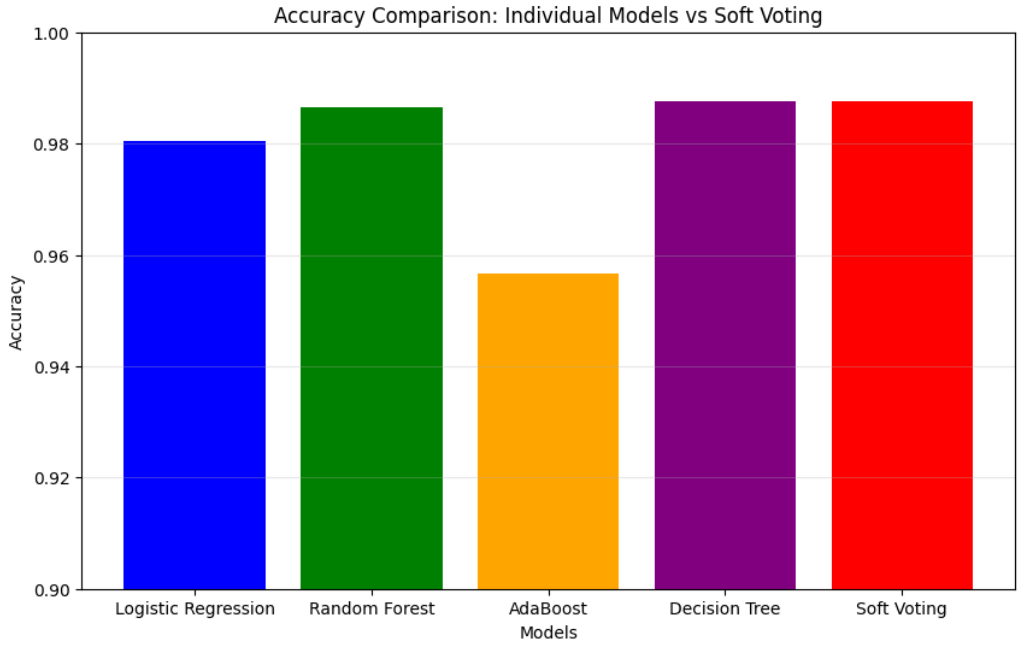


Figure 1: Comparison of all single models and the soft voting combining them

2.6 Stacking

Stacking used a validation set to train a meta-model, achieving 98% accuracy. This approach demonstrated excellent performance across all classes, with minimal confusion in the *low* and *medium* categories.

2.7 Regression Model

The MLP Regressor performed well in regression tasks $R^2 = 1.00, RMSE = 3.04$ but struggled with **classification**, achieving only 78% accuracy. It failed to predict *low* and *very low* due to data imbalance and the challenge of mapping continuous predictions to discrete classes.

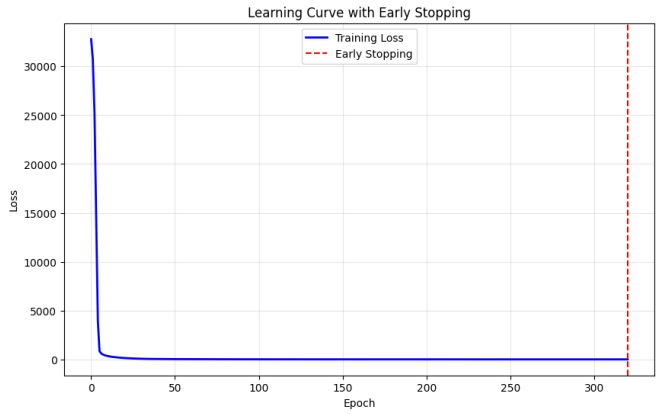


Figure 2: Learning curve with early stopping

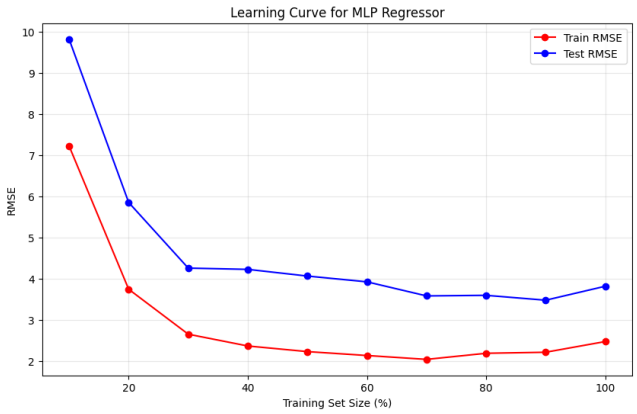


Figure 3: Learning Curve: RMSE vs. Training Set Size

3 Results and Conclusion

Although the MLP Regressor achieves an almost perfect R^2 score in regression (1.00) with a low RMSE (3.04), its **classification performance is unbalanced**. It does extremely well for "high," but fails to recognize "low" and "very low" (with zero precision and recall). **This suggests the model ignores smaller classes, likely due to imbalance or a mismatch between the regression-based training and the classification task**.

The MLP Regressor likely suffers from data imbalance and a mismatch between a regression objective and discrete class labels.
- Data Imbalance: Because "low" and "very low" classes have fewer samples, the model learns to predict the dominant classes ("high" or "medium").
- Regression vs. Classification: A regressor predicts continuous values, which can be hard to map cleanly to class labels, especially for minority classes.

	Classifier	Train Accuracy	Test Accuracy	Precision	Recall	F1 Score
0	Logistic Regression (Softmax)	97.91%	96.89%	96.89%	96.89%	96.87%
1	Decision Tree	99.51%	98.04%	98.04%	98.04%	98.03%
2	Random Forest	99.64%	97.97%	97.98%	97.97%	97.97%
3	AdaBoost	94.81%	94.45%	94.49%	94.45%	94.42%
4	Soft Voting	99.59%	98.10%	98.11%	98.10%	98.10%
5	MLP Regressor	-	78.00%	66.00%	78.00%	71.00%
6	Stacking	-	97.90%	97.90%	97.90%	97.90%

Figure 4: Comparison of all single models and the soft voting combining them

Most models perform very well, with Decision Tree, Random Forest, Soft Voting, and Stacking all reaching around 98% accuracy or higher. Logistic Regression (Softmax) also shows strong results at nearly 97%. AdaBoost lags slightly at 94.5%, and the MLP Regressor stands out as the weakest, likely due to the data imbalance. Overall, ensemble methods (Soft Voting and Stacking) appear to give the best performance.