Harnessing Artificial Intelligence for Smarter Cities: A Comparative Study of ML Models in Street Condition Prediction

Zachary K. Williams

Erik Jonsson School of Engineering and Computer Science

Author Note

This paper was written by Zachary Williams over the period of time from 7/28/2024 to 9/23/2024. The research conducted in this paper has no affiliation with any scholastic duties associated with the university that I attend; that is, this paper was written purely for fun, and was not part of any required academic work from the University of Texas at Dallas. The data that was used to conduct the study within this paper was provided by the City of Austin, TX through their open data portal. View the full source code I wrote for this study (along with significantly more graphs) at https://github.com/darman84/ML-with-Maintenance ; contact zacharykeith2000@gmail.com for any questions or comments.

Abstract

In this research paper, I investigate the development of optimal machine learning algorithms to predict the condition of street segments using historical maintenance data. Building on my previous work with machine learning applications in Cartegraph Asset Management software, I compare the performance of two advanced models: XGBoost, a decision tree-based algorithm, and a Multi-Layer Perceptron (MLP) neural network. I benchmark these models against a baseline Naive Bayes classifier to evaluate their accuracy and effectiveness in forecasting street segment maintenance needs. By leveraging the models' ability to capture complex, non-linear relationships within historical data—such as past repairs, asset characteristics, and external factors—I aim to identify the most effective algorithm for predictive maintenance in this application. This research contributes to AI-driven urban asset management by enhancing the operational efficiency of municipalities. Ultimately, my goal is to provide a scalable, adaptable solution that improves the quality and sustainability of public infrastructure services.

Keywords: artificial intelligence, machine learning, neural networks, predictive maintenance, public works, urban asset management, operational efficiency

Introduction

Background and Context

My journey toward predictive maintenance began during my tenure as a sign technician for the City of Plano Public Works Department. When an Asset Management Technician position became available, it piqued my interest due to my background in computer science and passion for machine learning and data science. Currently pursuing a Bachelor's degree in Computer Science at the University of Texas at Dallas while working full-time, I initially hesitated to apply for the new position. However, encouraged by colleagues who recognized the potential fit with my skillset, I decided to pursue the opportunity.

This decision led me to explore how machine learning and data analytics could enhance Public Works operations, particularly within the Cartegraph asset management software developed by OpenGov. I began investigating the potential of using historical data to predict maintenance requirements for city-owned infrastructure.

For the application process, I prepared a research paper demonstrating the ability to predict fire hydrant maintenance using a Histogram Gradient Boosting Regressor algorithm. The proof of concept I presented garnered attention on LinkedIn and within the Public Works department. Due to the positive feedback I received in my findings, I decided I would perform further research and began working on this study.

Research Objectives

This study builds upon my initial research, expanding its scope to street condition prediction. By employing both XGBoost and MLP models, I aim to:

1. Develop a detailed, hands-on methodology for predicting the current condition of street segments using historical data.
2. Compare the performance of XGBoost and MLP models against a baseline Naive Bayes classifier.
3. Document the process of feature engineering, model selection, hyperparameter tuning, and evaluation to enable others to apply a similar approach to different problems.

Significance of the Study

This research contributes to the field of AI-driven urban asset management in several ways:

1. It provides practical insights into how advanced machine learning techniques can be applied to predict street conditions, potentially improving maintenance planning and resource allocation.
2. The study serves as a guide for those who wish to leverage machine learning for predictive maintenance in other areas of public infrastructure or related fields.
3. By comparing different models, the research helps identify the most effective algorithms for this specific application, which could inform future studies and practical implementations.

The dataset used includes a variety of features, such as past repairs, asset characteristics, and external factors, which are crucial for capturing the complexities of street deterioration. Feature engineering and hyperparameter tuning were central to optimizing the performance of the models. Through grid search and cross-validation techniques, I selected relevant features and fine-tuned the models to improve predictive power.

Extensive testing was performed using a high-performance computing system equipped with an NVIDIA GEFORCE RTX 3070 GPU and a Ryzen 9 5900X CPU, providing the computational resources needed for large datasets and complex models.

The success of the models will be measured by their accuracy and ability to generalize to unseen data. By comparing the performance of the MLP and XGBoost models, this study aims to provide practical insights into how these techniques can be used for urban asset management, ultimately contributing to the development of smarter, more efficient cities.

Literature Review

Predictive Maintenance in Urban Infrastructure

Predictive maintenance in urban infrastructure has gained significant attention in recent years due to its potential to optimize resource allocation and extend the lifespan of public assets. Numerous studies have explored the application of machine learning techniques to forecast maintenance needs, primarily focusing on leveraging historical data to predict future asset conditions (Zhang et al., 2021).

In my previous work, I introduced a proof of concept utilizing the Histogram Gradient Boosting Regressor to predict the maintenance needs of fire hydrants. This study demonstrated the feasibility of applying machine learning models to municipal datasets, underscoring the potential for predictive maintenance in public works (Williams, 2024). This research paper builds upon this foundation by exploring more advanced models, such as XGBoost and Multi-Layer Perceptron (MLP) neural networks, to predict the condition of street segments.

Application of Machine Learning in Asset Management

The use of neural networks, particularly MLPs, in predictive maintenance has seen a notable rise in recent years. Neural networks are well-suited for this task due to their ability to model complex, non-linear relationships within data, which is often the case in urban infrastructure datasets. Goodfellow et al. (2016) note that neural networks, especially deep learning models, have demonstrated remarkable success across various domains, including predictive maintenance, due to their capacity to learn hierarchical representations of data. This makes them especially effective in scenarios where the relationships between input features and the target variable are intricate and non-obvious, as is often the case with street maintenance data.

In addition to neural networks, ensemble methods such as XGBoost have gained significant traction in the field of predictive maintenance. XGBoost, an implementation of gradient-boosted decision trees, has become a popular choice due to its efficiency and performance in handling large datasets with complex feature interactions. Chen and Guestrin (2016) highlight that XGBoost is particularly effective in real-world applications due to its ability to handle missing data, prevent overfitting, and provide interpretable results. In the context of urban asset management, XGBoost offers a robust solution for predicting maintenance needs by efficiently capturing the relationships between historical maintenance data and future asset conditions.

Gaps in Current Research

Despite the advancements in machine learning techniques, there remains a notable gap in the literature regarding the application of these models to real-world municipal datasets. Much of the existing research in predictive maintenance has been conducted in controlled environments or

on synthetic datasets, which limits the generalizability of the findings to real-world scenarios. This research aims to address this gap by evaluating the performance of machine learning models on actual municipal data, specifically from the City of Austin's Street and Bridge Operations department.

By comparing the performance of XGBoost, MLP, and a baseline Naive Bayes classifier, this study seeks to identify the most effective model for predictive maintenance in urban infrastructure, thereby contributing to the growing body of knowledge in AI-driven asset management.

The literature suggests that both neural networks and ensemble methods like XGBoost hold significant promise for predictive maintenance applications. However, there is a clear need for more research focused on real-world datasets, particularly in the context of municipal infrastructure. This study aims to address this gap by providing a comprehensive evaluation of machine learning models for predicting street segment conditions, ultimately contributing to the development of scalable and adaptable solutions for urban asset management.

Methodology

Dataset Description

For the development of the models, I utilized several publicly available datasets provided by the City of Austin. These datasets consisted of five distinct files, each contributing valuable information about the condition and maintenance history of the city's street segments. Two of these datasets contained annual grading information for road segments, assessed by a contractor hired by the City of Austin. These assessments covered approximately one-third to one-half of

all roads in the city each year. The grading process was conducted using sensors within the assessment vehicle, although the specific equipment and grading algorithms were not disclosed.

Key features extracted from these datasets included:

- "Segment ID" of the pavement asset
- "Street Name"
- "To & From" streets
- "Functional Class" of the pavement segment
- "Grade" assigned to the pavement
- Year of inspection

I also attempted to create meaningful features from a large dataset containing extensive traffic volume data. While this would have been an excellent addition, the dataset only covered 70 streets in Austin, which was insufficient for the majority of streets in the maintenance dataset. Additionally, I was unable to locate adequate public data regarding weather exposure, sun exposure, or other environmental factors specific to Austin's streets.

Insights from my previous research emphasized the importance of district location or features that delineate the general area associated with the asset. These features are critical as they account for variations in asset conditions across different parts of the city, where some areas may have older infrastructure or higher population densities. Consequently, I chose to retain these district-related features to ensure the model could capture these regional differences.

Two additional datasets provided a detailed history of maintenance activities performed by Austin's Street and Bridge Operations department on their pavement assets. Key features from these datasets included:

- Treatment type
- "Council District"
- "Segment ID" of the pavement asset
- "Street Name"
- "To & From" streets
- "Segment Width"
- "Lane Miles"
- Date of maintenance

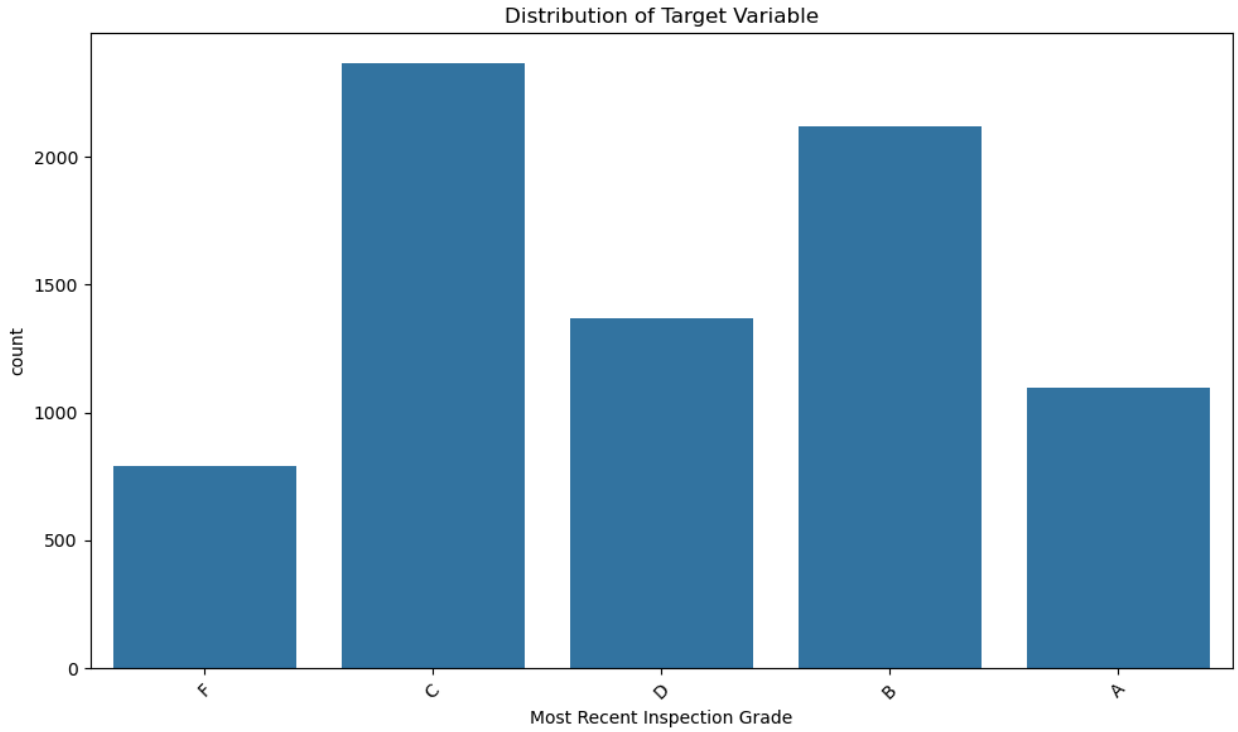
I won't go into the specifics regarding what each treatment type is, but the the treatment types documented in the dataset were:

- Sealcoat
- Overlay
- Slurry Seal
- Crack Seal
- Fog Seal

The final dataset included a ten-year history of traffic incidents (crashes, hazards, debris) and their locations. I used the street names to associate traffic incidents with the various street segments. The data for this was recorded somewhat inconsistently, which is why I decided to go with this simple approach over something more sophisticated.

Despite the limitations encountered, such as the lack of comprehensive traffic volume data and environmental factors, the datasets provided a solid foundation for developing predictive models. By incorporating district-level features and historical maintenance data, the models were able to account for the complex interactions between various factors that contribute

to street degradation.



Feature Engineering

Feature engineering played a crucial role in extracting meaningful insights from the available dataset. Given the dataset's limitations—spanning only five years instead of the ideal twenty-year period—I had to employ creative strategies to avoid generating synthetic data. The shorter time frame is understandable, as much of the data collection prior to 2016 was conducted on paper, making it difficult to obtain a longer historical record. Despite these constraints, I was able to derive valuable features that contributed to the model's predictive capabilities.

To address the challenges posed by the dataset, I applied various data preprocessing techniques:

- Missing numeric values were imputed using the mean
- Missing categorical values were filled with the mode
- Categorical variables were encoded using one-hot encoding
- StandardScaler was employed to standardize numeric features

Feature selection was a critical aspect of this process. I relied on a combination of domain knowledge and consultation with a pavement inspector who has over 20 years of experience in the field. This collaboration ensured that the most relevant attributes were included in the model, enhancing its ability to predict pavement conditions accurately.

The following features were selected for the model:

Target Variable:

1. Pavement grade (latest value: A, B, C, D, or F)

Input Variables:

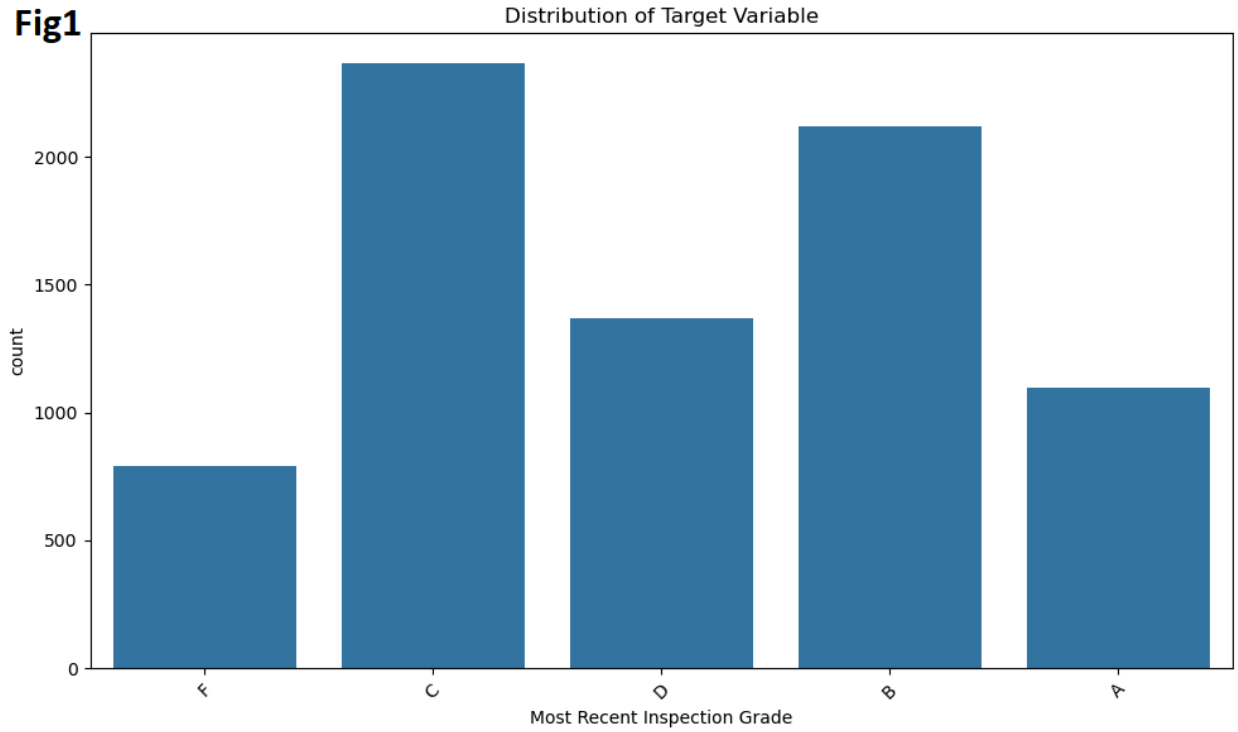
1. Treatment performed (type of treatment applied)
2. Segment ID (unique identifier for each street segment)
3. Council district (location of the segment within a council district)
4. Street Name
5. Segment length
6. Segment width
7. Lane miles
8. Project ID (group identifier for multiple segments worked on under the same project)

9. Most Recent Inspection Grade
10. Most Recent Inspection Year
11. Functional Class (residential, collector, arterial)
12. Days Since Last Maintenance
13. Number of each maintenance event
14. Total number of maintenance events
15. Number of traffic incidents

In addition to selecting existing features, I also engineered new features to capture temporal and cumulative aspects of the data. For instance, the "Days Since Last Maintenance" feature was created to quantify the time elapsed since the last maintenance activity, which is a crucial factor in understanding pavement deterioration over time. Similarly, I aggregated the number of each type of maintenance event and the total number of maintenance events for each segment, providing a comprehensive view of the segment's maintenance history.

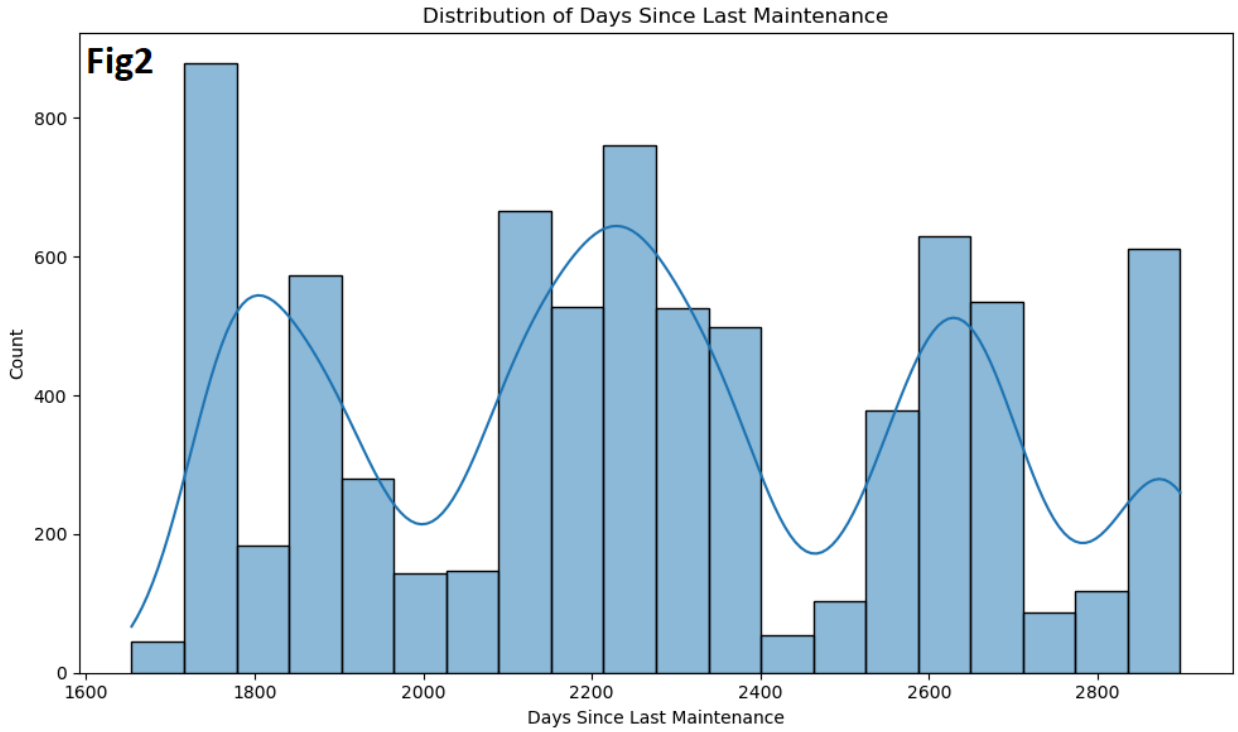
As seen in Fig1, the distribution between the classes for the target variable was quite imbalanced. This imbalance is an important consideration for model training and evaluation, as it

can affect the model's performance across different classes.



The creation of the "Days Since Last Maintenance" feature resulted in an imbalanced feature, as shown in Fig2. Despite this imbalance, I chose to retain this feature due to its potential

significance in predicting pavement conditions.



For the "number of traffic incidents" feature, I used a simplified approach due to data inconsistencies. I extracted street names from the 10-year traffic incident history dataset. If one of the columns in a row had the same street name as the current segment ID, the value for the feature was incremented. While this method may result in some inaccuracies, particularly for larger streets, it was the best approach given the inconsistencies in the dataset.

Effective feature engineering requires extensive domain knowledge, as it involves selecting and transforming variables that are most relevant to the problem at hand. By leveraging both domain expertise and data-driven techniques, I was able to create a feature set that captures the complex interactions between various factors contributing to street degradation. This approach not only enhances the model's predictive accuracy but also ensures that the analysis

remains grounded in real-world urban infrastructure management practices. Whilst I would have liked to have provided more detail into my thought process for feature engineering, the proper methodology will vary considerably based on the domain.

Model Architecture

Neural Network

In designing the architecture of the Multi-Layer Perceptron (MLP) neural network, I carefully considered the selection of activation functions, which play a crucial role in determining the network's ability to learn complex patterns. Among the options considered were sigmoid, tanh, and ReLU (Rectified Linear Unit). Each of these activation functions has distinct characteristics that influence the performance of the neural network.

The sigmoid function, defined as $\sigma(x)=\frac{1}{1+e^{-x}}$, is advantageous for its ability to map input values to a range between 0 and 1, making it useful for probability estimation. However, it is prone to the vanishing gradient problem, where gradients become increasingly small during backpropagation, leading to slow convergence and potential stagnation in training.

Similarly, the tanh function, expressed as $\tanh(x)=\frac{e^{x}-e^{-x}}{e^{x}+e^{-x}}$, maps input values to a range between -1 and 1, providing a stronger gradient than sigmoid, which can accelerate convergence. Nevertheless, tanh also suffers from the vanishing gradient problem, particularly in deep networks, where gradient values may diminish rapidly as they propagate through multiple layers.

Ultimately, I chose ReLU as the activation function for this study due to its significant advantages over both sigmoid and tanh. ReLU is defined as $f(x)=\max(0,x)$, meaning it outputs the input directly if it is positive, and zero otherwise. This simplicity makes ReLU computationally efficient, reducing the burden on training time. Moreover, ReLU mitigates the

vanishing gradient problem by maintaining a constant gradient for positive inputs, which allows for faster convergence and more effective training of deep networks.

However, ReLU is not without its drawbacks; it can suffer from the "dying ReLU" problem, where neurons become inactive if they consistently output zero during training, potentially reducing the model's capacity to learn. Despite this issue, the advantages of ReLU, particularly its efficiency and suitability for deep learning tasks, made it the preferred choice for this study.

My decision to employ an MLP neural network was justified by its ability to capture complex, non-linear relationships between features, making it an ideal candidate for predictive modeling in this context. MLPs are particularly well-suited for handling large-scale, high-dimensional data, which is critical for accurately predicting street conditions. Additionally, MLPs offer a high degree of flexibility in terms of architecture, allowing for precise control over the model's capacity to generalize from the training data.

XGBoost Decision Tree

I selected the XGBoost (Extreme Gradient Boosting) decision tree model for its efficiency and scalability, making it a strong candidate for both classification and regression tasks. Unlike traditional decision trees, which are prone to overfitting and may struggle with high-dimensional data, XGBoost incorporates several advanced techniques to improve both performance and generalization. These include regularization, parallel processing, and a more sophisticated approach to handling missing data.

XGBoost builds an ensemble of decision trees, where each tree is trained to correct the errors made by the previous trees in the sequence. This iterative process progressively reduces residual errors, leading to more accurate final predictions. The trees are constructed using a

greedy algorithm that splits the data at each node based on the feature that maximizes the gain in information.

Key features of XGBoost include:

- Regularization techniques (L1 and L2) to prevent overfitting
- Efficient handling of missing data
- Support for parallel processing
- Shrinkage or learning rate adjustment

My decision to use XGBoost for this task was justified by its ability to handle large, high-dimensional datasets efficiently while maintaining strong predictive performance. Compared to other tree-based algorithms, such as random forests, XGBoost offers superior control over model complexity through its regularization techniques and learning rate adjustments. Additionally, its ability to handle missing data and its support for parallel processing make it an ideal choice for real-world applications where data quality and computational resources may vary.

Hyperparameter Tuning

Neural Network

For hyperparameter tuning of the MLP, I used Optuna due to its adaptive sampling and pruning techniques, which outperform traditional methods like grid and random search. Key hyperparameters tuned included:

- Learning rate: 1*10^-5 to 1*10^-2 (logarithmic search)
- Number of layers: 1 to 5

- Hidden units: 32 to 128
- Dropout rate: 0.1 to 1.0
- Batch size: 32, 64, 128, and 256
- L1 and L2 regularization: 1*10^-8 to 1*10^-4 (logarithmic search)

I employed cross-validation to ensure robust model evaluation, using k-fold cross-validation to mitigate overfitting and provide a reliable estimate of the model's generalization ability.

XGBoost Decision Tree

For XGBoost hyperparameter tuning, I also used Optuna to optimize key parameters:

- Max Depth: 1 to 9
- Learning Rate: 1*10^-3 to 1.0 (logarithmic search)
- Number of Estimators: 100 to 1000
- Min Child Weight: 1 to 10
- Subsample and Colsample_bytree: 0.5 to 1.0
- Gamma: 1*10^-8 to 1.0 (logarithmic search)
- L1 and L2 Regularization: 1*10^-8 to 1.0 (logarithmic search)

As with the neural network, I employed k-fold cross-validation to ensure robust evaluation and mitigate overfitting.

Training the Model

After identifying the optimal hyperparameters through Optuna optimization, the MLP and XGBoost models were trained on the prepared dataset with an 80/20 split for training and

validation. This ensured a sufficient amount of data for learning while maintaining a reliable validation set to evaluate model performance.

MLP Model

The MLP model was trained using a structured process designed to minimize overfitting while achieving high accuracy. Key components of the training process included:

- **Loss Function**: The model used Cross-Entropy Loss, well-suited for multiclass classification, guiding the model to improve predictions by minimizing the difference between predicted and actual class probabilities.
- **Optimization Algorithm**: Adam optimizer, selected for its ability to handle sparse gradients and adapt learning rates, was employed with a tuned learning rate to ensure efficient convergence. Additionally, the ReduceLROnPlateau scheduler dynamically adjusted the learning rate based on validation performance, preventing stagnation during training.
- **Regularization Techniques**: Various regularization methods were employed to prevent overfitting:
    - **L1 Regularization**: Encouraged sparsity in model parameters, controlled by the l1_strength hyperparameter.
    - **L2 Regularization**: Applied via weight decay in the optimizer to penalize large weights, mitigating overfitting, with the strength determined by l2_strength.
    - **Batch Normalization**: Stabilized training by normalizing activations in each mini-batch, enhancing both learning speed and performance.

- **Dropout**: Randomly deactivated neurons during training to improve generalization, with dropout rates optimized through hyperparameter tuning.

Monitoring and Evaluation.

Training progress was carefully tracked using several key metrics:

- **Loss Monitoring**: Both training and validation losses were observed in real-time, providing immediate feedback on the model's learning process. The goal was a consistent decrease in training loss paired with stable validation loss to indicate effective learning.
- **Accuracy Tracking**: Classification accuracy for both training and validation sets was monitored using the torchmetrics.Accuracy metric, giving a direct measure of how well the model was performing on the classification task.
- **Early Stopping**: To avoid overfitting, early stopping was applied based on validation accuracy, halting training if no improvement was observed for 10 consecutive epochs.

The combination of these techniques allowed the MLP model to achieve a balance between generalization and performance, ensuring robust results on unseen data.

XGBoost Model

Following Optuna-based hyperparameter tuning, the XGBoost model was trained with the optimal parameters using gradient-boosting techniques. The process was designed to optimize performance on structured data while minimizing overfitting.

- **Cross-Validation**: A 5-fold cross-validation process ensured robust evaluation and helped prevent overfitting by testing the model across different data splits. This technique also provided a reliable estimate of the model's generalization capabilities.

- **Model Fitting**: The final model was trained on the full training dataset using mlogloss as the evaluation metric, which is particularly well-suited for multiclass classification tasks, ensuring that the model learned accurate class probabilities.

Monitoring and Evaluation.

After training, the XGBoost model was evaluated using a variety of methods:

- **Accuracy**: The model's overall classification performance on the test set was measured through accuracy, providing a clear indication of its ability to correctly predict class labels.
- **Classification Report**: A detailed classification report, including precision, recall, and F1-score for each class, was generated to highlight areas where the model excelled and where improvement was needed.
- **Confusion Matrix**: A confusion matrix was plotted to visualize the number of correct and incorrect predictions for each class, offering deeper insight into the model's strengths and weaknesses.
- **Learning Curve**: To assess the model's learning behavior over increasing amounts of data, a learning curve was plotted, providing insights into its ability to generalize as more training data becomes available.

By employing these monitoring techniques, the XGBoost model achieved strong performance while maintaining a high degree of generalization. The combination of robust cross-validation, carefully tuned hyperparameters, and detailed monitoring ensured the model's effectiveness in the classification task.

Results

XGBoost Model

After hyperparameter optimization with Optuna, the XGBoost model achieved the following key metrics:

- Cross-validation accuracy: 0.3715
- Test set accuracy: 0.3654
- Weighted average F1-score: 0.35

Classification Performance

The model's performance varied across classes:

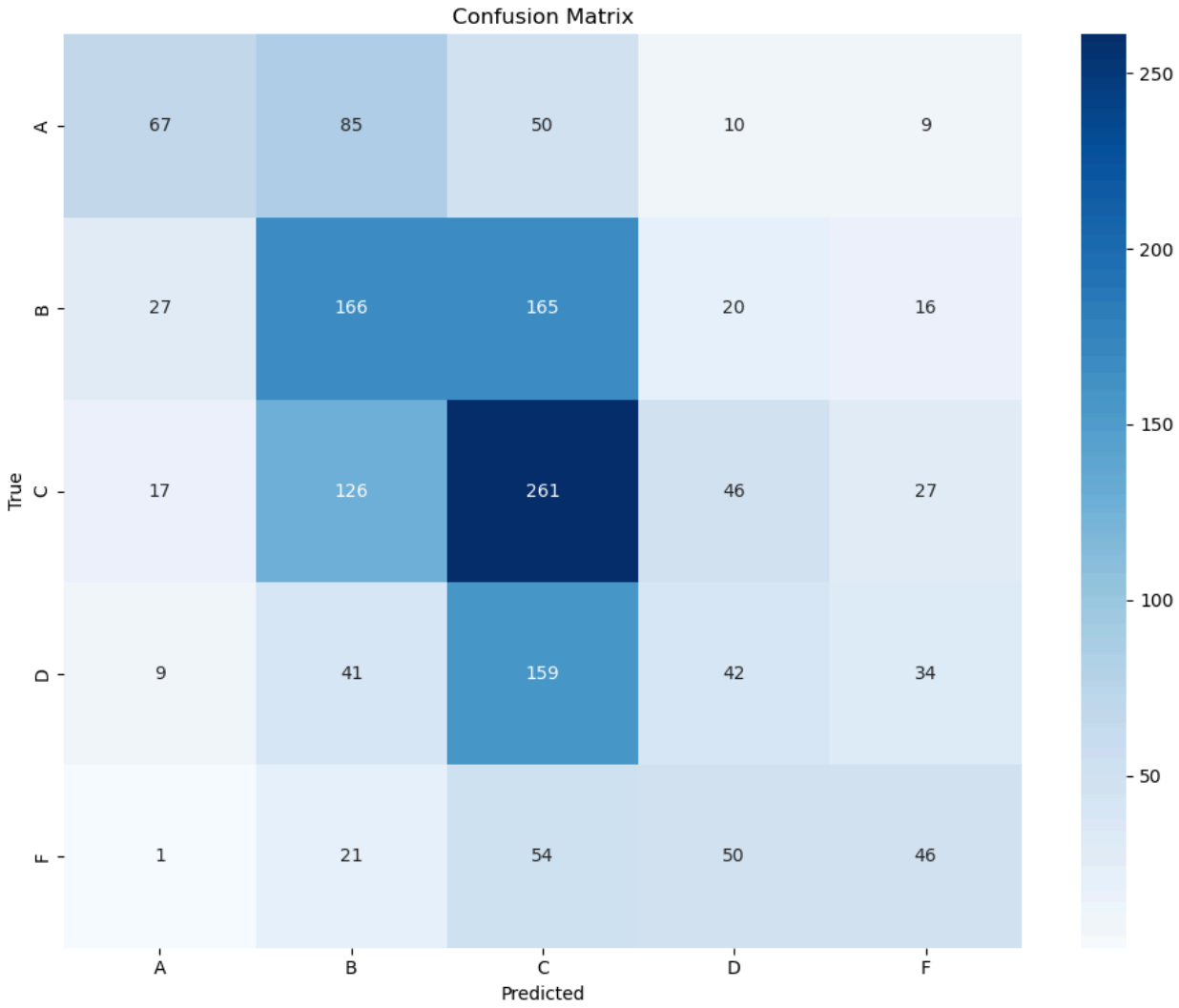
- Class A: Highest precision (0.60), but lower recall (0.37)
- Class C: Higher recall (0.51) with moderate precision
- Classes D and F: Lowest performance, with precision and recall values around or below 0.30

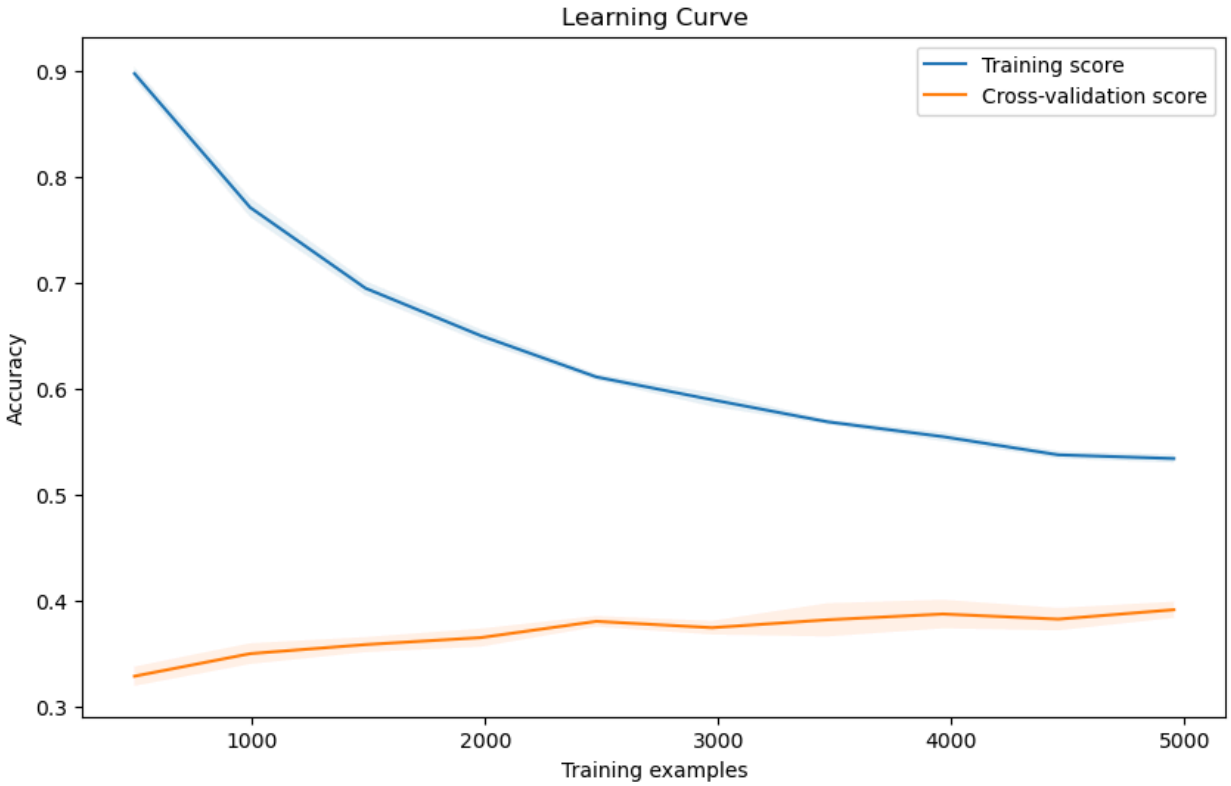
Feature Importance

The XGBoost model identified the following as the top 10 most important features:

1. Street Name_DESSAU RD (0.180010)
2. Street Name_PINEHURST DR (0.093322)
3. Street Name_AVERY RANCH BLVD (0.054928)
4. Functional Class_Arterial (0.044844)
5. Functional Class_Residential (0.028035)
6. Treatment_LUSLCT6 (0.027468)

7. Number of STROL events (0.025420)
8. Number of SLCT5 events (0.024256)
9. Number of LUSLCT6 events (0.022458)
10. Most Recent Inspection Year (0.022432)





MLP Neural Network

The MLP model showed similar performance trends, with the following key metrics:

- Cross-validation accuracy: 0.3672
- Test set accuracy: 0.3309
- Validation loss plateaued at 1.4996 from Epoch 2 onward

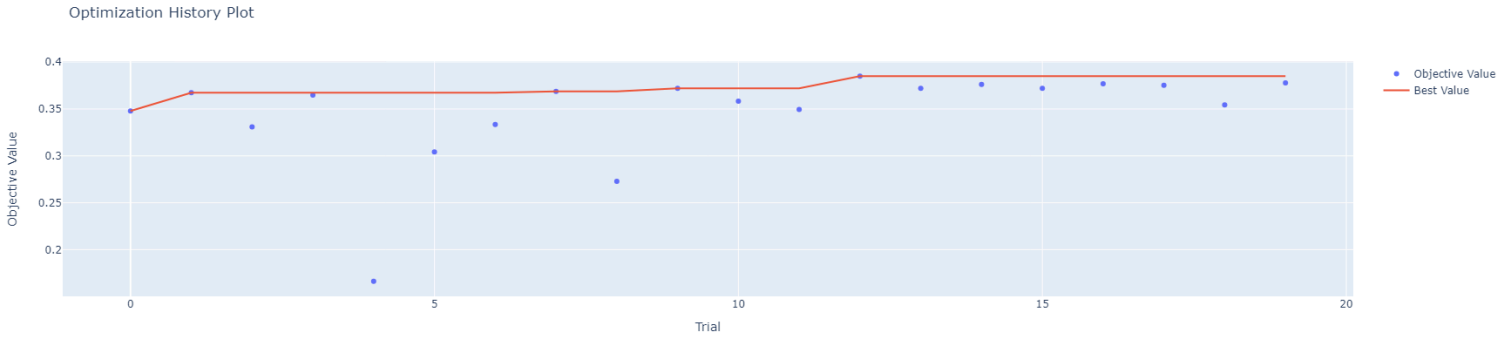
Classification Performance

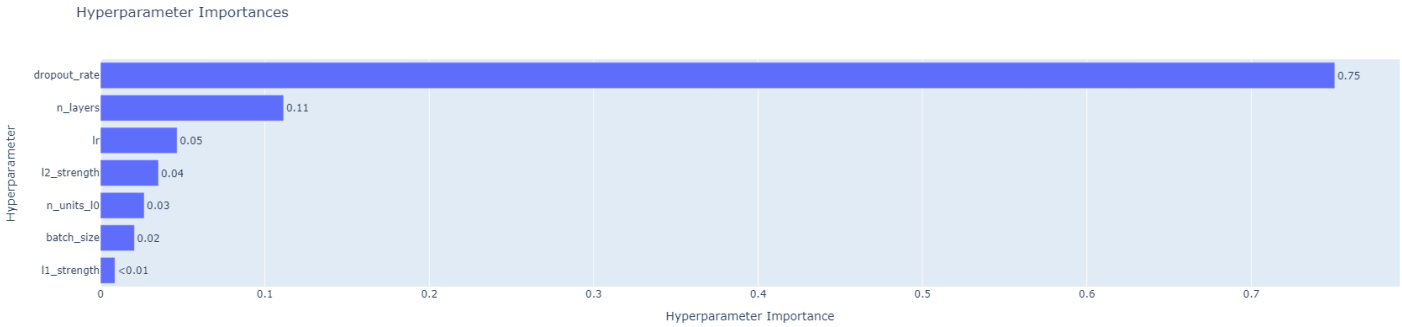
The MLP model displayed a pattern of low accuracy and loss improvement stagnation. The training accuracy (0.3502) and validation accuracy (0.3309) were relatively close. Loss and Accuracy Trends: The validation loss decreased slightly in the early epochs but plateaued at 1.4996 by Epoch 2.

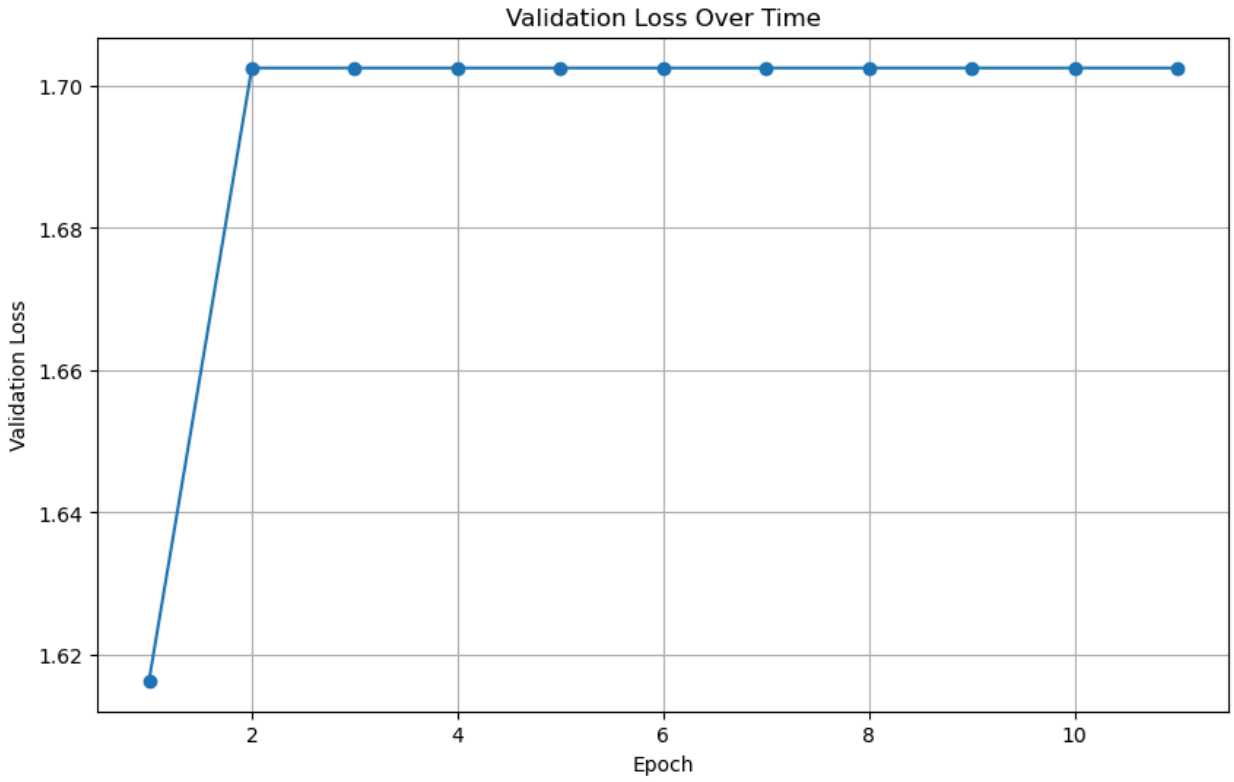
Feature Importance

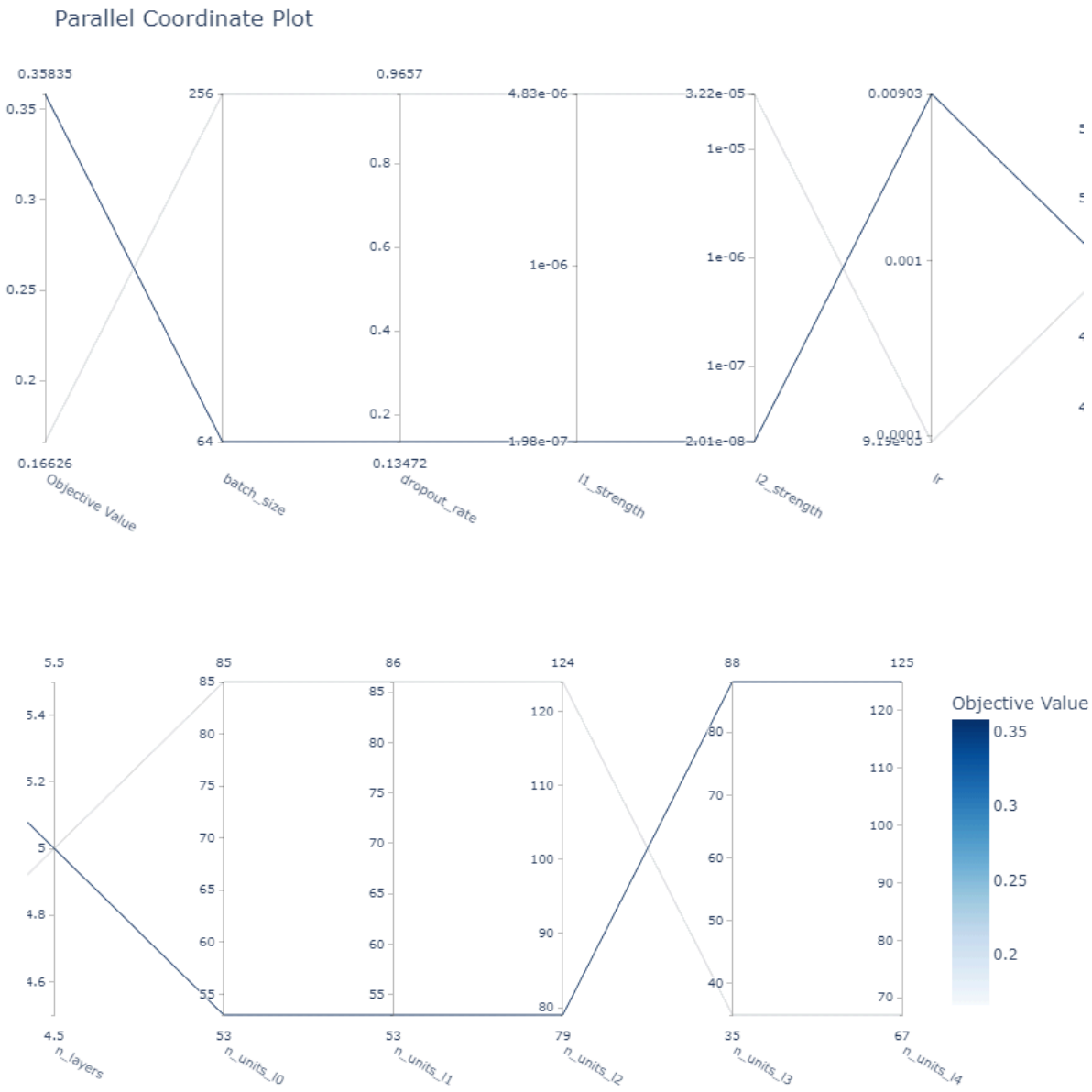
The MLP neural network identified the following as the top 10 most important features (excluding street names):

1. Functional Class_Arterial (2.4783735275268555)
2. Treatment_STROL (1.7360365390777588)
3. Treatment_LUSLCT5 (1.7137222290039062)
4. Treatment_SLCT5 (1.685530662536621)
5. Treatment_SLRYCDS (1.6790339946746826)
6. Lane Miles (1.6644357442855835)
7. Functional Class_Collector (1.6588066816329956)
8. Treatment_RC-COL (1.654392123222351)
9. Treatment_RH-FDR (1.6429554224014282)
10. Treatment_ZPLUOL (1.6303240060806274)


Optimization History Plot
Objective Value
Best Value
Objective Value
Trial
0.4
0.35
0.3
0.25
0.2
0
5
10
15
20


Hyperparameter Importances
dropout_rate
n_layers
lr
l2_strength
n_units_l0
batch_size
l1_strength
0.75
0.11
0.05
0.04
0.03
0.02
<0.01
Hyperparameter
Hyperparameter Importance
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7


Validation Loss Over Time
Validation Loss
Epoch
1.70
1.68
1.66
1.64
1.62
2
4
6
8
10



Model Comparison

Both models achieved similar overall performance in terms of accuracy (~36-37%):

- XGBoost: Higher test set accuracy (0.3654)

- MLP: Lower test set accuracy (0.3309)

A notable finding is that both models predicted the grade within 1 value of the true value 80% of the time:

- Overpredicting by 1 grade: 22% of the time
- Underpredicting by 1 grade: 21% of the time

Discussion

Interpretation of Results

Despite modest accuracy, my research still resulted in gaining valuable insight. This study's findings highlight several key points that warrant further discussion.

Model Performance and Implications

While the overall accuracy of approximately 40% may appear low, it represents a twofold improvement over random guessing in a five-class problem. The baseline naive bayes model (which will not be covered in detail in this paper), had 25% accuracy comparatively. Notably, both the XGBoost and MLP models demonstrated remarkable consistency in their predictions:

- 80% of predictions were within one grade of the true value
- 22% of predictions were one grade above the true value
- 21% of predictions were one grade below the true value

This consistency suggests that these models, despite their limitations, can provide a reliable general indication of a street's current condition. Such information could prove valuable for municipalities in prioritizing maintenance efforts and allocating resources more efficiently.

The XGBoost decision tree model emerged as the more practical choice for this task, offering good accuracy with moderate time investment in hyperparameter tuning. This aligns with XGBoost's reputation as a top-performing model in many machine learning competitions

(Chen & Guestrin, 2016). In contrast, the MLP neural network, while showing potential, appears to be overly complex for the current dataset and task, and in my opinion, is not worth the large amount of time and effort required to achieve optimal accuracy.

Feature Importance and Insights

Both models identified interesting patterns in feature importance. Street names and functional class were among the top features for XGBoost, suggesting local factors play a significant role in street condition. The MLP model highlighted the importance of functional class and various treatment types, indicating that road usage and maintenance history are crucial factors. Surprisingly, the "number of traffic incidents" feature was deemed unimportant, contrary to my initial hypothesis that it might correlate with road conditions.

Practical Applications

Despite the current limitations, this research demonstrates that municipalities can leverage machine learning on their datasets to gain insights into asset characteristics that influence condition and maintenance requirements. However, the varying data collection and maintenance practices across different municipalities present a significant challenge to widespread implementation.

Potential practical applications include:

- Prioritizing street maintenance based on predicted conditions
- Optimizing resource allocation for public works departments
- Developing more targeted and efficient maintenance schedules
- Informing long-term infrastructure planning and budgeting

Limitations and Future Work

The underfitting observed in the MLP model, evidenced by the early plateau in loss, likely indicates limitations in the dataset rather than issues with model complexity or hyperparameters. This suggests that the quality and quantity of available data are critical factors in model performance (Goodfellow et al., 2016).

The primary limiting factor is the availability and consistency of data across different municipalities. This inconsistency poses challenges for developing a universally applicable model and may require tailored approaches for each municipality.

Future work could explore several avenues to improve model performance and applicability:

- Investigating more complex neural network architectures, particularly if larger datasets become available
- Further refinement of features and exploration of additional data sources to enhance predictive power
- Addressing the imbalance in street grade classes to achieve more robust predictions across all categories
- Developing standardized data collection and maintenance practices across municipalities to improve model generalizability
- Exploring transfer learning techniques to adapt models trained on data-rich municipalities to those with limited data
- Incorporating real-time data sources, such as traffic patterns or weather conditions, to improve prediction accuracy

By addressing these limitations and pursuing these research directions, future studies could significantly enhance the accuracy and applicability of machine learning models in predicting street conditions, ultimately contributing to more efficient and effective urban infrastructure management.

Conclusion

This study represents a significant step towards applying machine learning techniques to street maintenance prediction, despite not achieving high accuracy levels. The key contributions and findings include:

- Demonstration of consistent prediction within one grade of the true value for 80% of cases, indicating a reliable general assessment of street conditions.
- Identification of important features influencing street maintenance grades, providing insights for municipal decision-making.
- Comparison of XGBoost and MLP models, highlighting the trade-offs between model complexity and performance for this specific task.
- Recognition of data quality and availability as critical factors in model performance and practical implementation.

The potential impact of this research on predictive maintenance practices is substantial. While we are still some distance from practical, widespread implementation, this work lays a foundation for future developments. As data collection and standardization improve across municipalities, the accuracy and applicability of these models are likely to increase significantly.

Future research should focus on:

- Improving data collection and standardization across municipalities
- Exploring more sophisticated model architectures and ensemble methods

- Incorporating additional data sources, such as real-time traffic data or weather patterns
- Developing strategies to address class imbalance and improve predictions for underrepresented street conditions

In conclusion, while challenges remain, this research demonstrates the potential of machine learning in revolutionizing municipal street maintenance practices. As techniques evolve and data quality improves, we can anticipate more accurate and actionable predictions, ultimately leading to more efficient resource allocation and improved urban infrastructure management.

References

Chen, T., & Guestrin, C. (2016). XGBoost: A scalable tree boosting system. Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 785-794. https://doi.org/10.1145/2939672.2939785

Zhang, Y., Chen, J., & Li, Z. (2021). Machine learning for predictive maintenance: A survey. IEEE Transactions on Industrial Informatics, 17(5), 2915–2926. https://doi.org/10.1109/TII.2020.3033186

Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep learning. MIT Press.

Combined Transportation, Emergency, and Communications Center (CTECC). (2023). Real-Time Traffic Incident Reports [Data set]. City of Austin, Texas Open Data Portal. https://data.austintexas.gov/Transportation-and-Mobility/Real-Time-Traffic-Incident-Reports/dx9v-zd7x

Austin Transportation & Public Works. (2021). Radar Traffic Counts [Data set]. City of Austin, Texas Open Data Portal. https://data.austintexas.gov/Transportation-and-Mobility/Radar-Traffic-Counts/i626-g7ub

Austin Transportation & Public Works. (2024). Strategic Measure_Street Segment Condition Data [Data set]. City of Austin, Texas Open Data Portal.

https://data.austintexas.gov/City-Infrastructure/Strategic-Measure_Street-Segment-Condition-Data/pcwe-pwxe/about_data

Austin Transportation & Public Works. (2024). Prototype Draft - Strategic Measure_Street Conditions_v2.0 [Data set]. City of Austin, Texas Open Data Portal. https://data.austintexas.gov/City-Infrastructure/Strategic-Measure_Street-Segment-Condition-Data/pcwe-pwxe/about_data

Austin Transportation & Public Works. (2024). Strategic Measure_Preventive Maintenance [Data set]. City of Austin, Texas Open Data Portal. https://data.austintexas.gov/City-Infrastructure/Strategic-Measure_Preventive-Maintenance/d7tv-xtsf/about_data

https://github.com/darman84/ML-with-Maintenance