CNST6308-Data Analysis in Construction Management

Transportation Asset Management Competition

Abdul Zakir Sayed: Writing- Reviewing and Editing, Data curation, Supervision, Validation, Software; **Syed Sohaib Ali**: Methodology, Software, Data curation, Validation, Software, Writing- Original draft preparation; **Mohammed Abdul Aziz**: Writing- Reviewing and Editing, Visualization, Investigation, Validation; Shaik **Misbah Bilal**: Validation, Writing- Reviewing and Editing

Department of Engineering and Data Science, University of Houston

Abstract

This report presents the process and results of participating in the "Transportation Asset Management (TAM) competition" organised by the Transportation Research Board (TRB) Standing Committee on Transportation Asset Management (AJE30). The competition focuses on addressing current challenges in transportation asset management through innovative approaches. In this project, we tackle the challenge of predicting culvert ratings using machine learning techniques.

The project begins by loading and preprocessing the CDOT Culvert dataset, which contains information about culverts. Additionally, the MerraClimate_Dat_Subset dataset is loaded and merged with the CDOT Culvert dataset to incorporate climate data for the year 2020. The dataset is cleaned, removing null values and duplicates, and categorical variables are transformed using one-hot encoding.

After preprocessing, the data is split into features and the target variable. To ensure consistent scaling, the features are standardised using the StandardScaler from the scikit-learn library. Three machine learning models, including Random Forest Classifier, CART Classifier, and Gaussian Naive Bayes Classifier, are trained and evaluated using various performance metrics.

The results demonstrate the effectiveness of the Random Forest and CART classifiers in predicting culvert ratings, achieving high accuracy, precision, recall, and F1 score. The Gaussian Naive Bayes classifier shows slightly lower performance in terms of precision and recall. These findings highlight the potential of machine learning models in asset management for transportation infrastructure.

Overall, this project showcases the application of machine learning techniques to address challenges in transportation asset management. The results suggest that leveraging data and advanced modelling approaches can contribute to better decision-making and resource allocation for maintaining transportation assets. Further research and optimization of the models can enhance their predictive capabilities and support more accurate asset management strategies.

1 Introduction

Transportation asset management plays a critical role in ensuring the safety, efficiency, and sustainability of transportation systems. As infrastructure networks continue to expand and age, it becomes increasingly important to adopt innovative approaches that address the challenges associated with managing these assets effectively. In line with this objective, the Transportation Research Board (TRB) Standing Committee on

Transportation Asset Management (AJE30) organised the "Transportation Asset Management (TAM) competition" in 2022-2023.

The TAM competition serves as a platform for researchers and practitioners to showcase their innovative solutions to the current challenges faced in transportation asset management. By encouraging the application of cutting-edge techniques and methodologies, the competition aims to foster advancements in this field and improve asset management practices.

This report presents our participation in the TAM competition, focusing on the specific challenge of predicting culvert ratings. Culverts are vital components of transportation infrastructure, enabling the safe passage of water under roadways. The ability to accurately assess and predict the condition and performance of culverts is essential for proactive maintenance planning and cost-effective asset management.

To address this challenge, we employed machine learning techniques to develop predictive models for culvert ratings. The project involved the utilisation of datasets, including the CDOT Culvert dataset, which contains relevant information about culverts, and the MerraClimate_Dat_Subset dataset, providing climate data for the year 2020. These datasets were processed and merged to incorporate environmental factors into the predictive models.

The first objective of this work is to determine whether culverts require maintenance or not based on the effect of various factors such as temperature, precipitation, pressure etc. For this we utilised different machine learning models and were assessed using performance metrics such as accuracy, precision, recall, and F1 score. Secondly, we determine precipitation change over a period of 100 years that is from 2000-2099.

The results of our study contribute to the broader understanding of how machine learning can be leveraged to improve transportation asset management. By accurately predicting culvert ratings, agencies and decision-makers can prioritise maintenance activities, allocate resources efficiently, and enhance the overall resilience of transportation infrastructure.

The remainder of this report is organised as follows: Section 2 provides a detailed description of the data preprocessing steps, including data cleaning and feature engineering. Section 3 presents the methodology employed in developing and evaluating the machine learning models. Section 4 discusses the results obtained from the models and their implications for transportation asset management. Finally, Section 5 concludes the report by summarising the key findings, highlighting the significance of the project, and suggesting areas for future research and improvement.

2 Methodology

In this project, we aim to analyse the CDOT Culvert dataset and predict the culvert rating with respect to the effect of precipitation using various machine learning models and to determine the change in precipitation over the years. The dataset is loaded using the Pandas library, and necessary libraries for data manipulation, model training, and evaluation are imported.

2.1 Change in precipitation over the years and its effect on culverts

To determine the change in precipitation over the years 2000 - 2099, we divide it into two parts so as to interpret the change in precipitation closely.

Initially we pre-process the precipitation data by checking for any null values and dropping any of the unwanted columns.

Calculating Change in Precipitation for the year 2000-2050:

1. We determined the average precipitation for each year in the time period 2000 - 2050. It is done by adding up the monthly or annual precipitation levels for each year and dividing by the number of months or years.
2. Plot the average precipitation for each year on a graph. This gives us a visual representation of how precipitation levels have changed over time.
3. Then to calculate the change in precipitation we subtracted the average precipitation level of the year 2000 from the average precipitation level of the year 2050.

Calculating Change in Precipitation for the year 2000-2099:

1. We determined the average precipitation for each year in the time period 2000 - 2099. It is done by adding up the monthly or annual precipitation levels for each year and dividing by the number of months or years.
2. Plot the average precipitation for each year on a graph. This gives us a visual representation of how precipitation levels have changed over time.
3. Then to calculate the change in precipitation we subtracted the average precipitation level of the year 2000 from the average precipitation level of the year 2099.

2.2 Predicting culvert performance and maintenance needs based on the culvert ratings

Initially we combined the CDOT Culvert dataset, which contains relevant information about culverts, and the MerraClimate_Dat_Subset dataset, so that we could train our models on the precipitation features that affect the culverts and its performance. To combine both the dataset we make use of the function .merge(), which merges the data on a common key which is 'MERRA_ID'.

2.2.1 Data Preprocessing and Cleaning:

Firstly we pre-processed the data by checking for any null values and dropping unwanted columns in the merged dataset. We also check for any duplicate values by using the function .duplicate(). By cleaning and preprocessing the text, the data is easier to analyse and more accurate results can be obtained. Then we perform Exploratory data analysis (EDA) to derive observations and insights.

2.2.2 Data Preparation and Scaling:

We select the relevant features from the merged dataset. Then to handle categorical variables, we create dummy variables using .get_dummies() . We also drop unnecessary columns from the dataset.

To ensure that all features are on the same scale, we use the StandardScaler from the scikit-learn library to standardise the training data. The function used for scaling the input is StandardScaler(). The same scaling transformation is applied to the test data.

2.2.3 Model Training and Evaluation:

Now, once the data is scaled we fit the scaled data to a machine learning model to determine the performance of the culverts. Here we use six different machine learning models trained and evaluated: Random Forest Classifier, CART (Decision Tree) Classifier, Naive Bayes Classifier, KNN Classifier, Neural Networks and Latent Dirichlet Allocation (LDA) Classifier. For each model, we fit the training data and make predictions on the test set. The performance of each model is evaluated using various metrics, including accuracy, precision, recall, F1 score, specificity, negative predictive value, Matthews correlation coefficient, and geometric mean.

1. Random Forest Classifier

Random Forest Classifier is a supervised machine learning algorithm used for classification tasks. It is an ensemble algorithm that creates multiple decision trees and combines their outputs to make predictions.

The algorithm works by first creating a set of decision trees, each of which is trained on a randomly selected subset of the input data. For each tree, a random subset of the input features is also selected to be used in making each decision.

Random Forest Classifier can be used for both binary and multiclass classification problems, and it can handle large datasets with high dimensionality. It is also capable of handling missing values and outliers in the input data.

We create a Random forest classifier object by using the randomForestClassifier() and pass the hyperparameters n_estimators=100, random_state=42, then we fit the training data to the object. Then we make predictions on the test data using.

Then we determine the Accuracy, Precision, Recall, F1 Score, Specificity, Negative Predictive Value, Matthew Correlation Coefficient, Geometric Mean.

2. Classification And Regression Trees

It is a decision tree-based machine learning algorithm that can be used for both classification and regression tasks. In CART, a tree structure is constructed by recursively partitioning the data into subsets, based on the values of the input features, and fitting simple models to each subset.

At each node of the tree, the algorithm selects the feature that best splits the data into two subsets with the greatest reduction in a chosen cost function. The process continues until a stopping criterion is met, such as reaching a maximum depth or minimum number of samples in a leaf node. Once the tree is built, it can be used to make predictions on new data by following the path from the root node to a leaf node, where a prediction is made based on the model that was fitted to that leaf.

We create a CART classifier object by using the DecisionTreeClassifier(), then we fit the training data to the object. Then we make predictions on the test data using.

Then we determine the Accuracy, Precision, Recall, F1 Score, Specificity, Negative Predictive Value, Matthew Correlation Coefficient, Geometric Mean.

3. Naive Bayes Classifier

Naive Bayes Classifier is a supervised machine learning algorithm used for classification tasks. It is based on Bayes' theorem, which describes the probability of an event occurring based on prior knowledge of conditions that might be related to the event.

In Naive Bayes, the algorithm makes a prediction by calculating the conditional probability of each class given the input features and selecting the class with the highest probability. The "naive" assumption of Naive Bayes is that the input features are conditionally independent given the class label, which means that the presence or absence of one feature does not affect the likelihood of the others. This assumption simplifies the probability calculations and makes the algorithm computationally efficient.

We create a Naive Bayes classifier object by using the GaussianNB(), then we fit the training data to the object. Then we make predictions on the test data using.

Then we determine the Accuracy, Precision, Recall, F1 Score, Specificity, Negative Predictive Value, Matthew Correlation Coefficient, Geometric Mean.

4. K-Nearest Neighbors Classifier

KNNClassifier, or K-Nearest Neighbors Classifier, is a supervised machine learning algorithm used for classification tasks. It is a non-parametric algorithm, which means that it does not make assumptions about the distribution of the input data.

In KNNClassifier, the algorithm makes a prediction by finding the K closest data points (neighbours) to the input data point in the feature space, where K is a user-defined hyperparameter. The class label of the input data point is then determined by the majority class of its K nearest neighbours. The distance metric used to measure the distance between data points can vary, but the most common is Euclidean distance.

We create a KNN classifier object by using the KNeighborsClassifier(), then we fit the training data to the object. Then we make predictions on the test data using.

Then we determine the Accuracy, Precision, Recall, F1 Score, Specificity, Negative Predictive Value, Matthew Correlation Coefficient, Geometric Mean.

5. Neural Netwroks MLPClassifier

The Multilayer Perceptron (MLP) Classifier is a type of neural network that consists of multiple layers of interconnected neurons. It is a feedforward neural network, meaning that information flows from the input layer through the hidden layers to the output layer without cycles or loops.

The MLP Classifier is trained using the backpropagation algorithm, which adjusts the weights and biases of the neurons to minimize the error between the predicted outputs and the actual outputs. The hidden layers of the MLP Classifier can have different activation functions, such as the sigmoid or hyperbolic tangent function, to introduce non-linearities and allow the network to learn complex patterns.

The MLP Classifier is commonly used for classification tasks and has been successful in various domains, including image recognition, text classification, and speech recognition. It can handle both binary and multi-class classification problems. The number of hidden layers and the number of neurons in each layer are hyperparameters that can be tuned to optimize the model's performance.

6. Latent Dirichlet Allocation

Latent Dirichlet Allocation (LDA) is a topic modelling algorithm used for unsupervised learning, which means that it does not require labelled data for training. LDA is used to identify topics in a collection of documents, but it is not a classifier on its own.

However, LDA can be used as a feature extraction technique to create a document-topic matrix, where each document is represented by a probability distribution over the topics identified by LDA. This matrix can then be used as input to a classifier, such as a Naive Bayes or Support Vector Machine (SVM), to predict the class labels of new documents.

LDA can be a useful feature extraction technique for text classification tasks, as it can capture the latent topics in the text data and reduce the dimensionality of the feature space.

We create a LDA classifier object by using the LinearDiscriminantAnalysis(), then we fit the training data to the object. Then we train the random forest classifier on the LDA-transformed data. Lastly, we make predictions on the test data using.

3 Results and Visualization

In this section we will summarize the experimental results.

3.1 Effect of change in precipitation on culvert

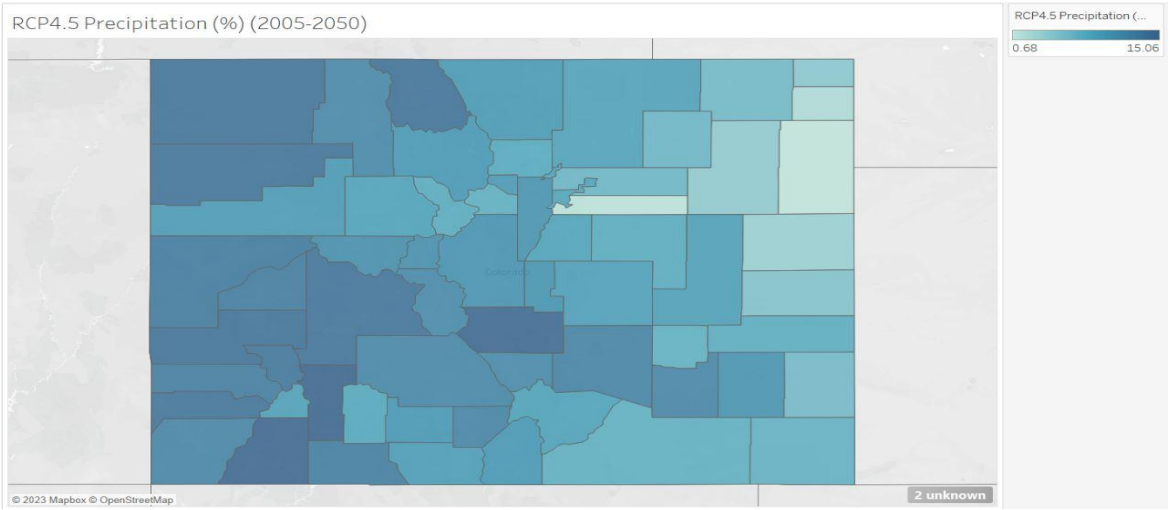
Changes in precipitation can have a significant impact on culverts and their performance. Precipitation refers to the amount of water that falls from the atmosphere to the Earth's surface in the form of rain, snow, sleet, or hail. Here are some effects of changes in precipitation on culverts.

- Increased Water Flow: Higher levels of precipitation can lead to increased water flow in rivers, streams, and drainage systems. This can result in a larger volume of water passing through culverts, potentially exceeding their capacity. If a culvert becomes overwhelmed by the increased flow, it may lead to flooding, erosion, or even structural failure.

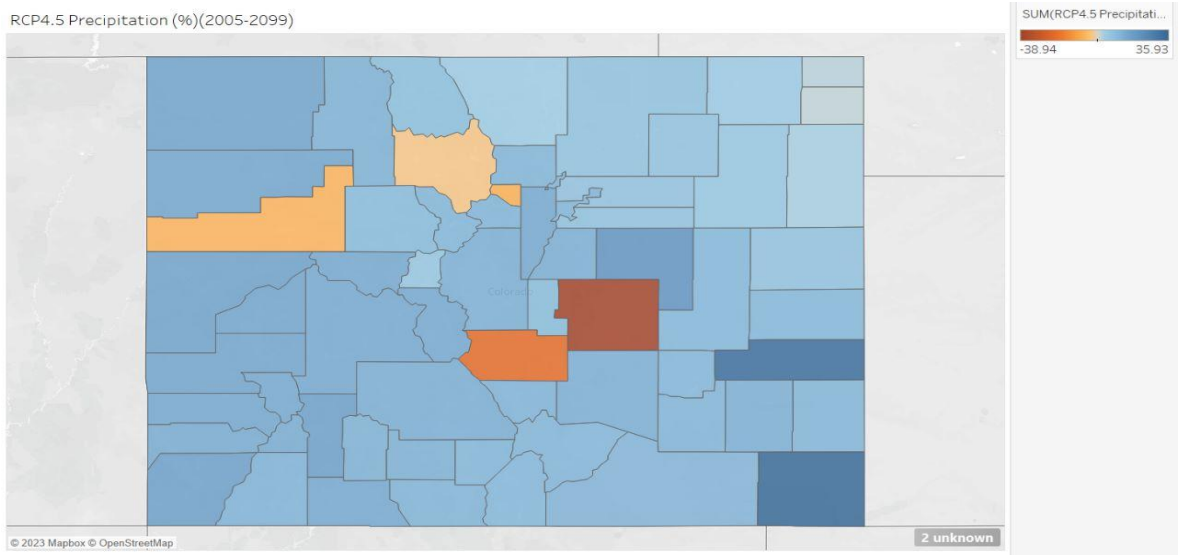
- Sediment Transport: Precipitation can cause erosion and the transport of sediment from the surrounding area into culverts. Sediment can accumulate within the culvert, reducing its capacity and obstructing the flow of water. This can result in reduced hydraulic efficiency and increased maintenance needs for the culvert.

- Scouring and Undermining: High-intensity rainfall or prolonged periods of precipitation can lead to scouring and undermining around culverts. The force of the water can erode the soil around the culvert's foundations or the downstream side, creating voids and instability. This can compromise the structural integrity of the culvert and increase the risk of failure.

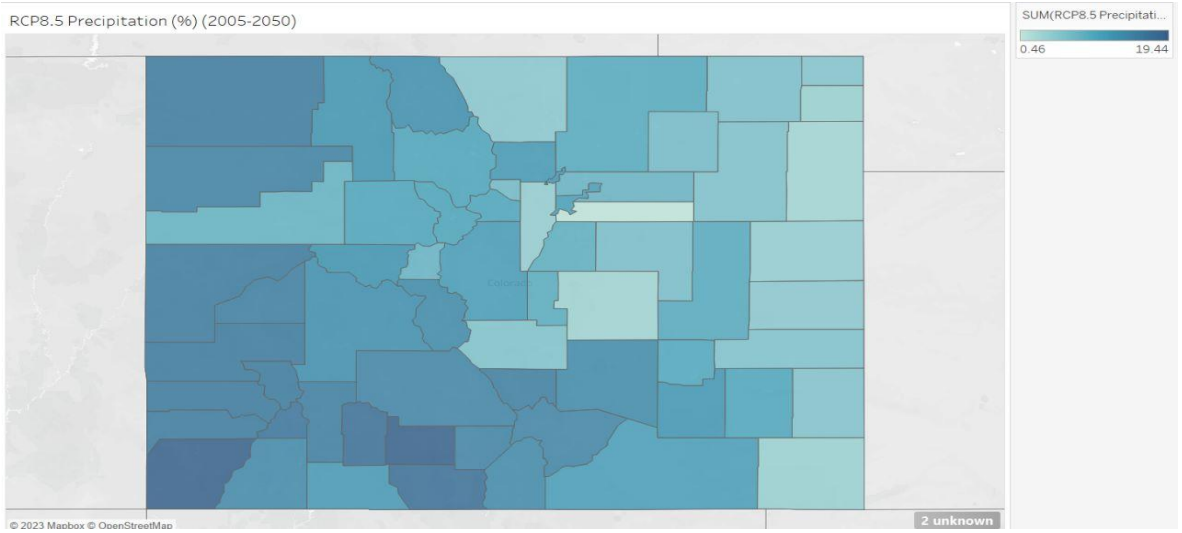
In the project we have analyzed RCP 4.5 and RCP 8.5 scenarios suggested potential changes in precipitation patterns that can impact culverts. Understanding these projections can help inform the design, maintenance, and management of culverts to ensure their resilience to future climate conditions. Local-scale assessments, incorporating specific regional characteristics and considering adaptation strategies, are essential for effectively addressing the impacts of climate change on culvert performance. After the analysis we came up with maps projecting percent change in precipitation in colorado state. The plots are as follows



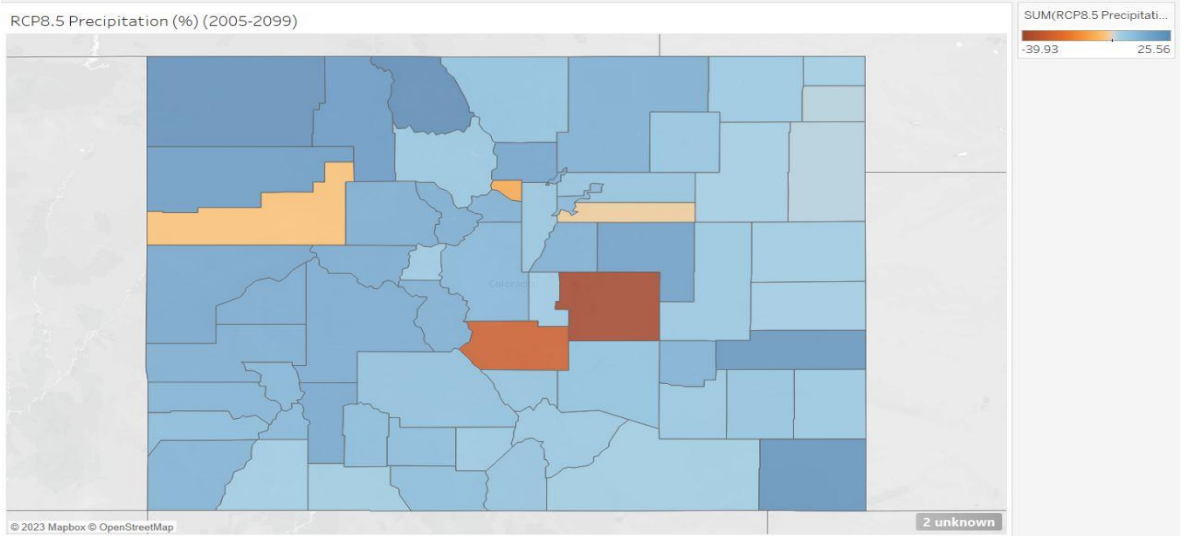
RCP4.5 Precipitation (%)(2005-2099)



RCP8.5 Precipitation (%) (2005-2050)



RCP8.5 Precipitation (%) (2005-2099)



From the hue of the graphs we can see that some county will see increase in rainfall and some may see decline in rainfall, To protect culverts from increased rainfall and mitigate potential impacts, several measures can be implemented. Here are some strategies to consider:

- Regular Maintenance: Ensure regular inspection and maintenance of culverts to keep them clear of debris, sediment, and vegetation. Regular cleaning and removal of blockages will help maintain the culvert's capacity and prevent backups during heavy rainfall events.

- Upsizing and Redesign: Evaluate the existing culverts' capacity and consider upsizing them to accommodate increased water flow. Redesigning culverts to handle larger volumes of water can help prevent overflow and reduce the risk of flooding or structural damage.

- Improved Drainage Systems: Assess the surrounding drainage systems and make improvements if necessary. Consider installing additional catch basins, ditches, or swales to divert and manage excessive runoff away from the culvert. Properly designed drainage systems can help alleviate pressure on culverts during heavy rainfall.

- Erosion Control Measures: Implement erosion control measures around the culvert and in the upstream and downstream areas. This may include using erosion control blankets, geotextiles, or rock armoring to stabilize the soil and prevent erosion that could undermine the culvert structure..

- Climate-Resilient Design: Incorporate climate change considerations into the design of new culverts or culvert replacements. This may involve using climate projections to estimate future rainfall patterns and designing culverts with larger capacity or other innovative features to withstand increased water flow.

- Community Awareness and Preparedness: Educate the community about the risks and impacts of increased rainfall on culverts. Encourage property owners to take measures to reduce their contributions to stormwater runoff. Community preparedness and early warning systems can help mitigate potential damages and respond effectively during extreme rainfall events.

There are several other measure we can take to protect our asset, above are some general measures that a personal can follow.

3.2 Predicting culvert ratings to determine performance and maintenance needs

3.2.1 Random Forest Classifier

These results indicate that the model performs well with high precision, recall, and F1 score across multiple classes. The accuracy of 0.9717 suggests that the model correctly predicts the class label for 97.17% of the instances. The precision, recall, and F1 scores indicate the model's performance for each class, showing high values in most cases. The specificity and negative predictive value are also perfect for this particular classification task. The Matthew Correlation Coefficient measures the overall quality of the predictions, and the geometric mean provides an aggregated measure of performance across all classes.

Overall, the model seems to be performing very well, achieving high accuracy and demonstrating good performance across various evaluation metrics.

```
Confusion Matrix:

 [[ 266    0    0    0    0    0    0    0]
 [   0   47    0    0    0    0    0    0]
 [   0    0  559    0    1    6    1    4]
 [   0    0    1  490    0    4    9    0]
 [   0    0    6    0 2471   73   19    0]
 [   0    0    3    0   45 8099  187    0]
 [   0    0    0    1    3  173 7968   26]
 [   0    0    1    0    0    2   42  939]]

Classification Report:
              precision    recall  f1-score   support

           0       1.00      1.00      1.00       266
           2       1.00      1.00      1.00        47
           3       0.98      0.98      0.98       571
           4       1.00      0.97      0.98       504
           5       0.98      0.96      0.97      2569
           6       0.97      0.97      0.97      8334
           7       0.97      0.98      0.97      8171
           8       0.97      0.95      0.96       984

    accuracy                           0.97     21446
   macro avg       0.98      0.98      0.98     21446
weighted avg       0.97      0.97      0.97     21446

Accuracy: 0.971696353632379
Precision: 0.9717416829603709
Recall: 0.971696353632379
F1 Score: 0.9716967753264244
Specificity: 1.0
Negative Predictive Value: 1.0
Matthew Correlation Coefficient: 0.9586841957087707
Geometric Mean: 0.9781047475515054
```

3.2.2 Classification And Regression Trees

These results indicate that the CART model performs exceptionally well. The accuracy of 0.9793 suggests that the model correctly predicts the class label for 97.93% of the instances. The precision, recall, and F1 scores are high for each class, indicating the model's ability to accurately classify instances. The specificity and negative predictive value are perfect, indicating the model's ability to correctly identify negative instances. The Matthew Correlation Coefficient measures the overall quality of the predictions, and the geometric mean provides an aggregated measure of performance across all classes.

In summary, the CART model demonstrates excellent performance across various evaluation metrics, indicating its effectiveness in classifying instances accurately.

```
Confusion Matrix:

 [[ 266    0    0    0    0    0    0    0]
 [   0   47    0    0    0    0    0    0]
 [   0    0  570    0    1    0    0    0]
 [   0    0    0  503    0    0    1    0]
 [   0    0    7    0 2499   52   11    0]
 [   0    0    0    2   69 8157  106    0]
 [   0    0    0    2    5  134 8007   23]
 [   0    0    0    0    0    0   30  954]]

Classification Report:
              precision    recall  f1-score   support

           0       1.00      1.00      1.00       266
           2       1.00      1.00      1.00        47
           3       0.99      1.00      0.99       571
           4       0.99      1.00      1.00       504
           5       0.97      0.97      0.97      2569
           6       0.98      0.98      0.98      8334
           7       0.98      0.98      0.98      8171
           8       0.98      0.97      0.97       984

    accuracy                           0.98     21446
   macro avg       0.99      0.99      0.99     21446
weighted avg       0.98      0.98      0.98     21446

Accuracy: 0.9793434673132518
Precision: 0.9793428904994347
Recall: 0.9793434673132518
F1 Score: 0.9793411022187021
Specificity: 1.0
Negative Predictive Value: 1.0
Matthew Correlation Coefficient: 0.9698911801392271
Geometric Mean: 0.9845075291680415
```

3.2.3 Naive Ba0.1yes Classifier

These results indicate that the Naive Bayes model's performance is quite poor in this case. The accuracy of 0.1624 suggests that the model correctly predicts the class label for only 16.24% of the instances. The precision, recall, and F1 scores are low for most classes, indicating the model's difficulty in accurately classifying instances. The specificity is relatively high, suggesting the model performs better at identifying negative instances. The negative predictive value is perfect, but this is likely due to the class imbalance in the dataset. The Matthew Correlation Coefficient and geometric mean are also quite low.

In summary, the Naive Bayes model shows poor performance in terms of accuracy, precision, recall, and F1 score. It struggles to effectively classify instances across multiple classes, resulting in low overall performance.

```
Confusion Matrix:

 [[ 256    3    0    0    0    0    4    3]
 [   0   33    0    0   14    0    0    0]
 [   3  125    0    0  430    8    0    5]
 [   6   72    5    3  396    7    5   10]
 [   2  250    2   15 2165   54   60   21]
 [  37  457    2    6 6916  377  380  159]
 [  63  427    3   15 6313  629  561  160]
 [  18   20    1    2  503  277   75   88]]

Classification Report:
               precision    recall  f1-score   support

           0       0.66      0.96      0.79       266
           2       0.02      0.70      0.05        47
           3       0.00      0.00      0.00       571
           4       0.07      0.01      0.01       504
           5       0.13      0.84      0.22      2569
           6       0.28      0.05      0.08      8334
           7       0.52      0.07      0.12      8171
           8       0.20      0.09      0.12       984

    accuracy                           0.16     21446
   macro avg       0.24      0.34      0.17     21446
weighted avg       0.34      0.16      0.12     21446

Accuracy: 0.16240790823463583
Precision: 0.3399261995793943
Recall: 0.16240790823463583
F1 Score: 0.11906360385573019
Specificity: 0.9884169884169884
Negative Predictive Value: 1.0
Matthew Correlation Coefficient: 0.04658586955448389
Geometric Mean: 0.3742721921172076
```

3.2.4 K-Nearest Neighbors Classifier

These results indicate that the k-Nearest Neighbors (KNN) model's performance is moderate. The accuracy of 0.5568 suggests that the model correctly predicts the class label for 55.68% of the instances. The precision, recall, and F1 scores vary across classes, with higher values for some classes (e.g., class 0) and lower values for others (e.g., class 3, 4, 5). The specificity is 1.0, indicating the model performs well at identifying negative instances. The negative predictive value is perfect, but this is likely due to the class imbalance in the dataset. The Matthew Correlation Coefficient and geometric mean are moderate, suggesting the model's overall performance is not strong.

In summary, the k-Nearest Neighbors (KNN) model shows moderate performance in terms of accuracy, precision, recall, and F1 score. It performs reasonably well for some classes, but struggles with others. The model's performance could potentially be improved by tuning the hyperparameters or exploring different algorithms.

```
Confusion Matrix:

 [[ 266    0    0    0    0    0    0    0]
 [   0   19    3    1   12    5    7    0]
 [   0    1  211   11   95  178   74    1]
 [   0    2   36  118   97  175   75    1]
 [   0    3  100   37  934 1015  476    4]
 [   0    3  107   48  730 5293 2123   30]
 [   0    3   67   31  484 2749 4745   92]
 [   0    0    4    8   49  204  364  355]]

Classification Report:
              precision    recall  f1-score   support

           0       1.00      1.00      1.00       266
           2       0.61      0.40      0.49        47
           3       0.40      0.37      0.38       571
           4       0.46      0.23      0.31       504
           5       0.39      0.36      0.38      2569
           6       0.55      0.64      0.59      8334
           7       0.60      0.58      0.59      8171
           8       0.73      0.36      0.48       984

    accuracy                           0.56     21446
   macro avg       0.59      0.49      0.53     21446
weighted avg       0.56      0.56      0.55     21446

Accuracy: 0.5567938077030682
Precision: 0.5593519655753111
Recall: 0.5567938077030682
F1 Score: 0.5528711988493181
Specificity: 1.0
Negative Predictive Value: 1.0
Matthew Correlation Coefficient: 0.34022230235181933
Geometric Mean: 0.6556751301214082
```

3.2.5 Neural Networks MLPClassifier

These results indicate that the Multilayer Perceptron (MLP) Classifier's performance is moderate. The accuracy of 0.5220 suggests that the model correctly predicts the class label for 52.20% of the instances. The precision, recall, and F1 scores vary across classes, with higher values for some classes (e.g., class 0) and lower values for others (e.g., class 3, 4, 5). The specificity is 1.0, indicating the model performs well at identifying negative instances. The negative predictive value is perfect, but this is likely due to the class imbalance in the dataset. The Matthew Correlation Coefficient and geometric mean are moderate, suggesting the model's overall performance is not strong.

In summary, the Multilayer Perceptron (MLP) Classifier shows moderate performance in terms of accuracy, precision, recall, and F1 score. It performs reasonably well for some classes, but struggles with others. The model's performance could potentially be improved by adjusting the network architecture, tuning hyperparameters, or exploring different neural network models.

```
Confusion Matrix:

 [[ 266    0    0    0    0    0    0    0]
 [   0   10    4    0    6   18    9    0]
 [   0    2   73    0   40  353  103    0]
 [   0    0   27   10   61  335   71    0]
 [   0    1   34    0  292 1752  487    3]
 [   0    0   12    0  211 5943 2133   35]
 [   0    3    6    0   84 3678 4247  153]
 [   0    0    1    0    0  202  428  353]]

Classification Report:
              precision    recall  f1-score   support

           0       1.00      1.00      1.00       266
           2       0.62      0.21      0.32        47
           3       0.46      0.13      0.20       571
           4       1.00      0.02      0.04       504
           5       0.42      0.11      0.18      2569
           6       0.48      0.71      0.58      8334
           7       0.57      0.52      0.54      8171
           8       0.65      0.36      0.46       984

    accuracy                           0.52     21446
   macro avg       0.65      0.38      0.41     21446
weighted avg       0.53      0.52      0.49     21446

Accuracy: 0.5219621374615313
Precision: 0.5342648228120708
Recall: 0.5219621374615313
F1 Score: 0.4928516795821772
Specificity: 1.0
Negative Predictive Value: 1.0
Matthew Correlation Coefficient: 0.26274135430350015
Geometric Mean: 0.611475366311946
```

3.2.6 Latent Dirichlet Allocation

In summary, the LDA model's performance varies across different classes. It achieves perfect precision, recall, and F1-score for class 0, indicating accurate identification of all instances in that class. However, for other classes, the model's performance is not as high.

Class 2 shows a relatively high precision but lower recall, suggesting that the model correctly identifies instances of class 2 but may miss some. Class 5 has a moderate precision and recall, indicating that the model identifies a significant number of instances correctly but also has a notable number of false positives.

The weighted average metrics, such as accuracy, precision, recall, and F1-score, provide an overall evaluation of the model's performance. The accuracy of the LDA model is 0.6686, indicating the proportion of correct predictions overall.

It's important to consider the specific context and requirements of the problem when interpreting these results. Different evaluation metrics may hold varying levels of importance, and acceptable performance thresholds can differ based on the application at hand.

```
Confusion Matrix:

 [[ 266    0    0    0    0    0    0    0]
 [   0   26    0    2    2   12    5    0]
 [   0    0  263    6   48  172   80    2]
 [   0    1    6  205   39  167   86    0]
 [   0    0   35    9 1120  928  472    5]
 [   0    0   30    9  272 6118 1883   22]
 [   0    0   20   17  166 2082 5794   92]
 [   0    0    2    3    8  144  280  547]]

Classification Report:
              precision    recall  f1-score   support

           0       1.00      1.00      1.00       266
           2       0.96      0.55      0.70        47
           3       0.74      0.46      0.57       571
           4       0.82      0.41      0.54       504
           5       0.68      0.44      0.53      2569
           6       0.64      0.73      0.68      8334
           7       0.67      0.71      0.69      8171
           8       0.82      0.56      0.66       984

    accuracy                           0.67     21446
   macro avg       0.79      0.61      0.67     21446
weighted avg       0.68      0.67      0.66     21446

Accuracy: 0.6686095309148559
Precision: 0.6757667350453904
Recall: 0.6686095309148559
F1 Score: 0.6637752968716027
Specificity: 1.0
Negative Predictive Value: 1.0
Matthew Correlation Coefficient: 0.5017090347858467
Geometric Mean: 0.7367115525330262
```

4 Conclusion

Changes in precipitation patterns can significantly impact culverts, leading to increased water flow, sediment transport, scouring, and undermining. An analysis of RCP 4.5 and RCP 8.5 scenarios projected potential changes in precipitation in Colorado. Some counties may experience increased rainfall, while others may face a decline. To protect culverts from these impacts, regular maintenance, upsizing and redesign, improved drainage systems, erosion control measures, and climate-resilient design are crucial. Community awareness and preparedness play a vital role in mitigating damages caused by increased rainfall. Implementing these measures can enhance the resilience of culverts and reduce the risks associated with changing precipitation patterns.

In conclusion for the Predicting culvert ratings to determine performance and maintenance needs, we evaluated several classification models. The Random Forest Classifier and Classification And Regression Trees (CART) models demonstrated excellent performance, achieving high accuracy, precision, recall, and F1 scores across multiple classes. These models are well-suited for the task and can accurately classify instances, indicating their potential in determining culvert ratings.

On the other hand, the Naive Bayes Classifier showed poor performance, with low accuracy and low precision, recall, and F1 scores across most classes. It struggled to effectively classify instances, indicating limitations in its ability to predict culvert ratings accurately.

The K-Nearest Neighbors (KNN) Classifier and Multilayer Perceptron (MLP) Classifier showed moderate performance. While they achieved reasonable accuracy and some classes had high precision, recall, and F1 scores, they also faced challenges in accurately classifying instances across all classes. Fine-tuning the hyperparameters or exploring different algorithms may help improve their performance.

Finally, the Latent Dirichlet Allocation (LDA) model's performance varied across different classes. It achieved perfect scores for class 0 but showed lower precision and recall for other classes. The weighted average metrics indicated a moderate overall performance.

Considering the results, the Random Forest Classifier and CART models stand out as the top performers for predicting culvert ratings. Further experimentation and optimization can be conducted to enhance the performance of these models and ensure their suitability for practical implementation in determining culvert performance and maintenance needs.

5 Reference

https://www.tam-portal.com/wp-content/uploads/2021/01/AJE-30-TAM-competition-guidelines.pdf

https://scikit-learn.org/stable/

https://www.tableau.com/trial/tableau-software

https://en.wikipedia.org/wiki/Culvert

https://www.un.org/en/climatechange/what-is-climate-change#:~:text=Climate%20change%20refers%20to%20long,activity%20or%20large%20volcanic%20eruptions.