EAI 6020: AI System Technologies

Week 1:

Regression Model Performance Report on Student Loan

Submitted To:
Prof. Siddharth Rout, Faculty Lecturer

Submitted By:
Abhilash Dikshit
Murtaza Vora
Gunjan Paladiya

Academic Term: Winter 2024
Northeastern University, Vancouver, BC,Canada
Master of Professional Studies in Analytics

March 02, 2024

Part 1:

Introduction:

In the first part of the report, we will discuss about ideal ratio between a model's complexity and capacity for generalization is crucial in the field of machine learning. Important ideas to grasp in this context are the Bias-Variance Trade-off, underfitting, and overfitting. When a model learns the training data too well, it overfits and performs badly on unknown data because it captures noise in addition to the underlying pattern. On the other hand, underfitting occurs when the model is unable to identify the underlying pattern in the data, leading to subpar performance on both new and training sets. The goal of the Bias-Variance Trade-off is to minimize overall error and improve prediction performance on new datasets by striking a careful balance between a model's complexity (variance) and its assumptions about the training data (bias). In-depth

What is Overfitting?

In machine learning, overfitting happens when a model performs poorly on unknown data because it collects noise or random oscillations in the training data instead of the desired output. This frequently happens when a model has too many parameters in relation to the quantity of observations, making it overly complex. A decision tree that grows excessively deep, for instance, may learn to memorize the training set, including its anomalies and outliers, and hence perform poorly when applied to fresh data.

There are several tactics that can be used to counter overfitting. Cross-validation is a useful technique that ensures the model performs consistently across sets by splitting the dataset into subsets and utilizing some for testing and others for training. Regularization is an additional strategy that discourages complex models by adding a penalty to the magnitude of the model coefficients. Common techniques that reduce the model's complexity and assist prevent overfitting are L1 and L2 regularization.

Reducing the number of features in the model or applying methods like decision tree pruning can also aid in its simplification. Reducing overfitting in neural networks can be achieved by adopting dropout, which ignores randomly chosen neurons during training. This reduces the sensitivity of the network to the weights of individual neurons. These techniques seek to achieve a compromise between the model's generalization capabilities to previously unknown data and its learning from the training set.

What is Underfitting?

In machine learning, underfitting happens when a model is too simplistic to fully represent the underlying structure of the data, which is typified by low variance and high bias. This is frequently the consequence of applying overly simple models to complicated problems, which results in subpar performance on both training and unobserved data. Underfitting, for instance, may occur when non-linear data are fitted using a linear regression model.

Increasing the complexity of the model is necessary to overcome underfitting. More features, more sophisticated models, or the use of polynomial characteristics rather than linear ones can all help achieve this. Methods such as cross-validation, which demonstrate the model's performance on omitted data, might be helpful in locating underfitting.

Regularization techniques like L1 and L2 regularization, which apply a penalty to the loss function dependent on the magnitude of the coefficients, can be used to stop overfitting. Neural networks can also be trained using dropout, which randomly ignores neurons and forces the network to learn more resilient characteristics. To reduce overfitting and make sure the model fits fresh data correctly, other useful techniques include pruning, early halting, and validating the model using a hold-out set.

What is Bias-Variance Trade-off?

A key idea in machine learning is the Bias-Variance Trade-off, which expresses the conflict between a model's potential for success on training data (bias) and its potential for generalization to new data (variance). High bias models are oversimplified and may result in underfitting, a situation in which the model fails to consider the pertinent relationships between features and desired outputs. In contrast, high variance models overanalyze training data, including noise, which causes overfitting and subpar performance on fresh data.

Fitting a polynomial regression model to data points serves as an example. The curvature of the data may not be well captured by a linear model (low degree polynomial), resulting in significant bias and low variance. Conversely, a very high degree polynomial may have low bias but large variance if it fits the training data exactly but oscillates erratically between data points.

Several tactics are needed to overcome overfitting: reducing the model's complexity by choosing a more suitable one, penalizing overly complex models with regularization techniques (e.g., Lasso or Ridge regression), and utilizing data partitioning techniques like cross-validation to estimate the model's performance more accurately on unseen data. Regularization strategies discourage complex models that overfit the training data by adding a penalty term to the loss function to constrain the model's coefficients. Cross-validation facilitates improved model complexity and parameter adjustment by evaluating how well the model's predictions transfer to a separate dataset.

Part 2

Student Loan Regression Model Performance Report

I. Introduction:

This report evaluates the performance of a regression model trained to predict student loan amounts originated based on various features. The model was trained on a dataset containing information about student loans and evaluated on a separate validation set. The analysis includes preprocessing steps, model training, evaluation metrics, and recommendations for further improvement.

II. Data Preprocessing:

- The dataset was loaded from an Excel file (`FL_Dashboard_AY2009_2010_Q1.xls`) and preprocessed to prepare it for model training.

- Dollar amount columns were cleaned by removing '$' and ',' characters and converting them to float values.

- Missing values, represented by '-', were replaced with 0.

- Categorical variables ('School', 'State', 'School Type') were encoded using one-hot encoding.

- Numerical features were normalized using standard scaling, excluding identifier columns

('OPE ID', 'Zip Code').

III. Model Training:

- A neural network regression model was implemented using PyTorch.

- The model architecture consists of three fully connected layers with ReLU activation functions.

- The dataset was split into training and validation sets using a 80:20 ratio.

- The Adam optimizer was used for optimization, with a learning rate of 0.001.

- The model was trained for 1000 epochs, and the training and validation losses were recorded.

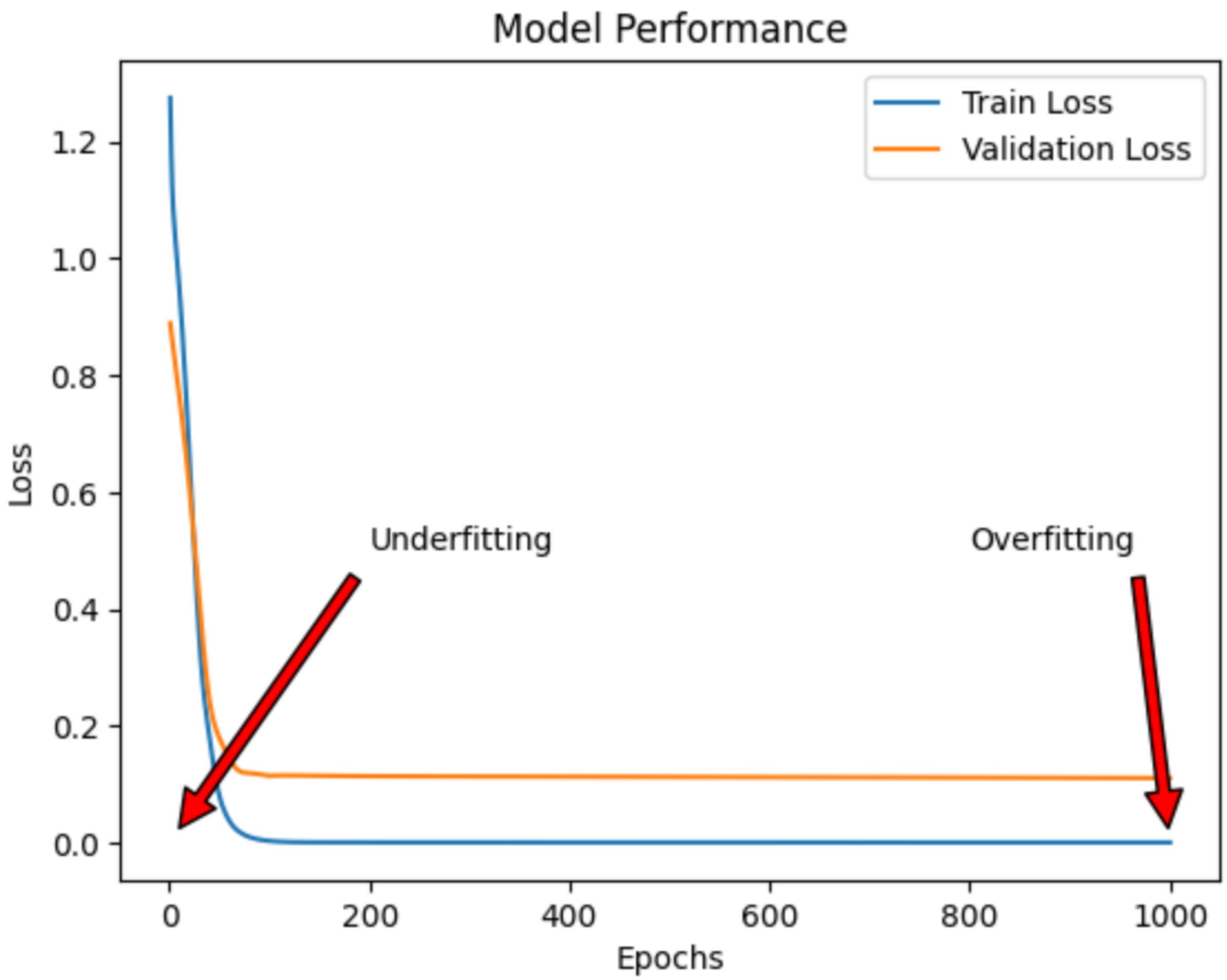
IV. Model Evaluation:

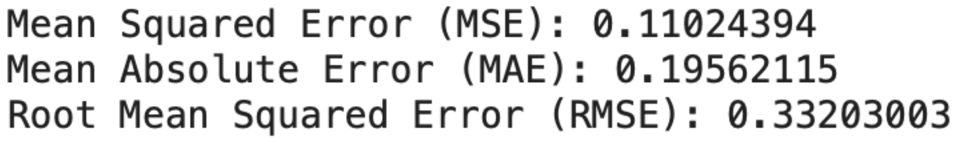
- The model's performance was evaluated using regression metrics: MSE, MAE, RMSE.

- The evaluation on the validation set yielded the following results:

  - Mean Squared Error (MSE): 0.11024394
  - Mean Absolute Error (MAE): 0.19562115
  - Root Mean Squared Error (RMSE): 0.33203003

- These metrics indicate that the model's predictions are relatively close to the actual values on average, demonstrating promising performance.





```
Mean Squared Error (MSE): 0.11024394
Mean Absolute Error (MAE): 0.19562115
Root Mean Squared Error (RMSE): 0.33203003
```

V. Conclusion:

- The regression model performs well in predicting student loan amounts originated based on the provided features.

- The low values of MSE, MAE, and RMSE suggest that the model captures the underlying patterns in the data effectively.

- Continued monitoring of the model's performance and further refinement may enhance its effectiveness in real-world applications.

VI. Recommendations:

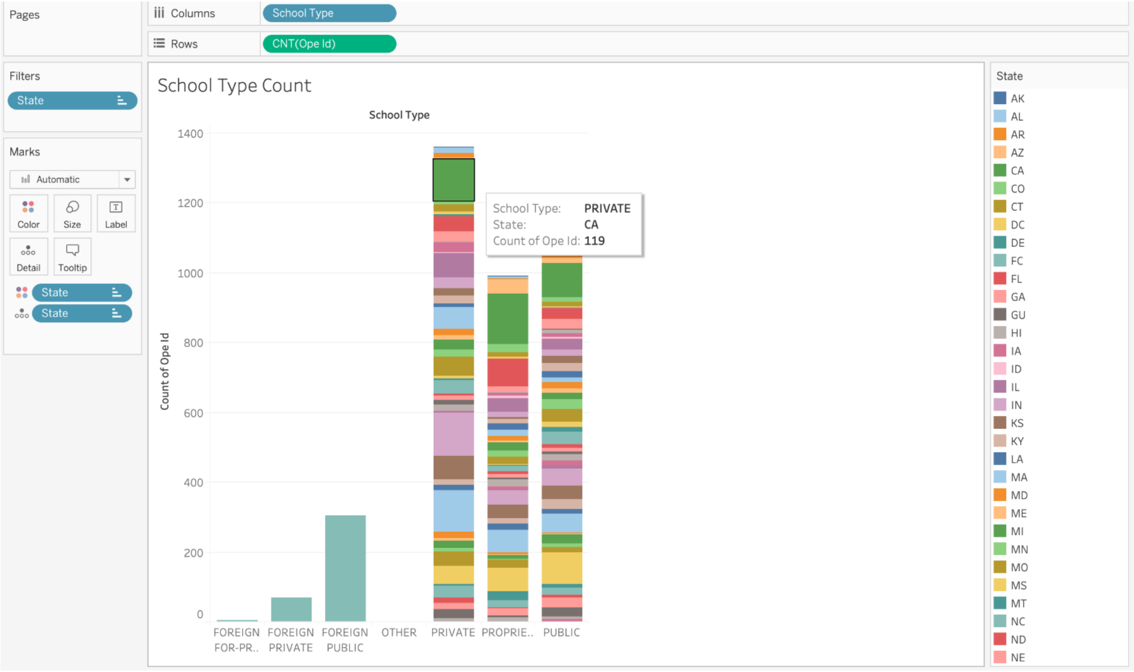
- Consider exploring additional features or refining the model architecture to potentially improve performance further.

- Conduct sensitivity analysis to understand how changes in hyperparameters or preprocessing steps affect model performance.

- Evaluate the model's performance on additional datasets or time periods to assess its generalization ability and robustness.

Overall, the regression model shows promise in predicting student loan amounts originated and serves as a valuable tool for financial institutions and policymakers in understanding and managing student loan trends. Further refinement and evaluation will enhance its utility and effectiveness in decision-making processes.

VII. Tableau Analysis

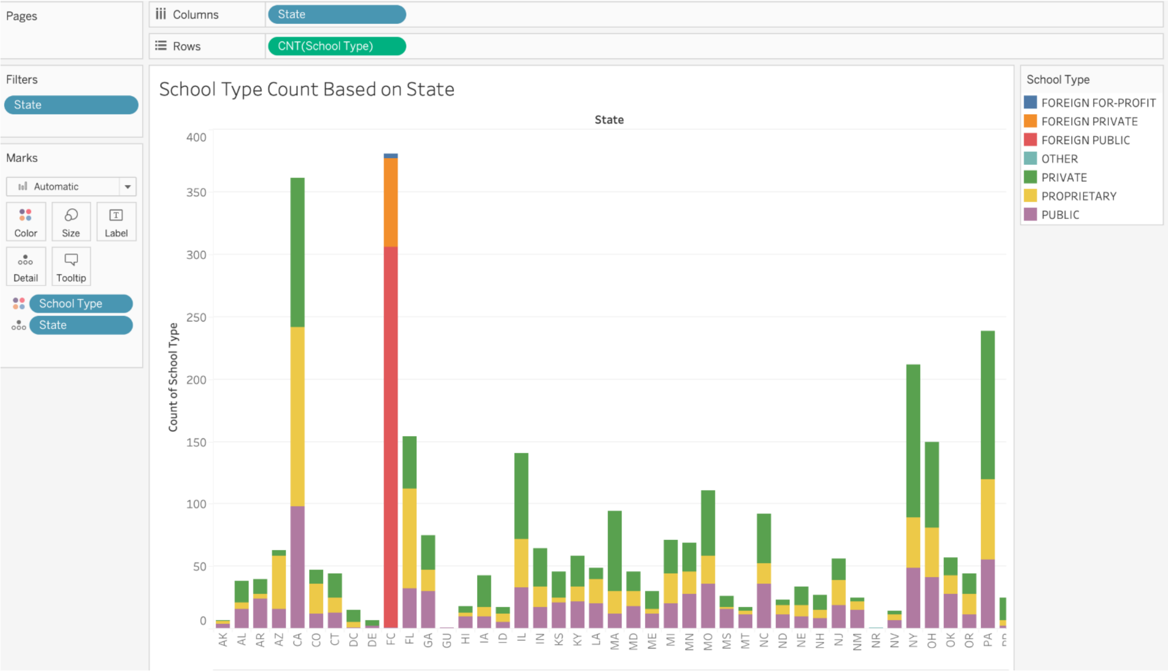


Columns: School Type

Rows: CNT(Ope Id)

Filters: State

School Type: PRIVATE
State: CA
Count of Ope Id: 119



VIII. References

- Brownlee, J. (2019a, August 12). Overfitting and underfitting with machine learning algorithms. MachineLearningMastery.com. https://machinelearningmastery.com/overfitting-and-underfitting-with-machine-learning-algorithms/

- West, D. M., & Allen, J. R. (2018, April 24). How artificial intelligence is transforming the world. Brookings. https://www.brookings.edu/articles/how-artificial-intelligence-is-transforming-the-world/