Group 6: Anna Lavrentieva and Iris Yu

Data 612 Deep Learning

Due August 15, 2023

Final Project: Neural Networks for Predicting Flood Density in New York State

Abstract / Summary

The risks of flash flooding are increasing in New York state, as climate and land use changes cause more severe storm events and greater likelihoods of flooding during storms. These floods can cause disastrous loss of life and property that impacts communities long after the storm is over, yet forecasting flooding is difficult due to the large number of variables with non-linear relationships involved. This is the motivation for using artificial neural networks to create accurate predictions of flooding in New York using publicly available data. Mapping tools from ArcGIS online and Pro were used to assemble flood risk geologic and hydraulic variables, and flood density was used as a predictor for modeling flood risks. Because of the infrequency of severe flooding events, the data has an inflated amount of zeros for flood density, which was overcome by scaling the data to a smaller range and resampling the lower range of zero risks to create less imbalanced data. Fully connected neural networks of multi layer perceptrons were used to generate a regression of flood density. Deeper networks with more hidden layers and trained on resampled data performed the best, leading us to conclude that resampling is a valid strategy for overcoming issues of studying natural weather phenomena with zero-inflated outcome variables.

Problem Statement: Apply artificial neural networks to predict flood density from publicly available datasets in New York state.

1 Background and Significance

Climate change is shifting patterns in the frequency and intensity of natural disasters. In the United States, extreme precipitation in the Northeast may increase by 52% by 2099 (Picard, 2023), resulting in increased risks of flood and flash flood events. In July 2023, severe flooding devastated several states along the east coast. West Point in New York received over 7 inches of rain over the course of 4 hours, which shut down trains lines and interstates (Betts, 2023). On a personal level, the two authors of this project would have been caught in the storm in West

Point, but changed plans last minute to come back to Maryland ahead of schedule and avoided the floods by lucky circumstances.

To focus the scope of the study, we propose the target state of New York. According to the Risk Factor tool, over the next 30 years, New Yor has 17% of total state properties at over 26% of flood risk (Bates, 2021). New York also has large swaths of farmland, with 6.9 million acres, as well as highly populated urban areas, which are greatly affected by flash floods.

Flash floods have significant and damaging social and economic impacts, from loss of life to property damage that sets back communities for years afterwards. Flood predictors can be utilized to build climate-resilient communities. This includes land use and zoning planning considerations for local officials and natural disaster preparation such as asset prepositioning and response coordination practice. The property insurance industry also requires accurate risk assessments, and homeowners can be properly prepared.

2 Motivation for Deep Learning

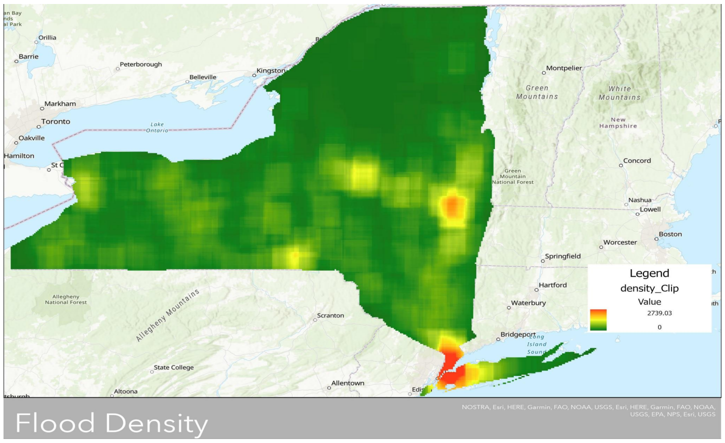
We have previously utilized non-deep learning methods to predict intermittent precipitation in Tacoma, Washington using data collected from weather services. Our previous methodology involved the use of a similar zero-inflated approach to modeling, using an XGBoost classifier to classify instances of precipitation occurring. We then trained a random forest regression model on all instances of precipitation that did occur. However, this methodology only achieved an accuracy of 70% when predicting the following hour's precipitation in millimeters.

We believe that as extreme weather becomes increasingly common, it is critical to be able to accurately forecast these events for the sake of safety and public infrastructure. By using neural networks, we hope to be able to model the complex non-linear relationships to forecast upcoming events more accurately and produce maps of where flooding may take place.

3 Data Sources and Variables

3.1 Data Sourcing and Loading

Public data sets available on ArcGIS online through a UMD academic license were utilized to build the variables of this project. Historic flood data was sourced from NOAA's Storm Events Database, which lists historic flood and flash flood events since the 1950s and associated characteristics (National Centers for Environmental Information). The specific target variable is flood density (Figure 1), which was calculated in ArcGIS as the magnitude of flood occurrences per cell using historic flood locations in the Storm Events database.



Flood Density

NOSTRA, Esri, HERE, Garmin, FAO, NOAA, USGS, Esri, HERE, Garmin, FAO, NOAA, USGS, EPA, NPS, Esri, USGS

Figure 1. Output variable, flood density, that this project seeks to predict from inputs

Input variables came from geomorphologic data that are known to be associated with flood risks, such as elevation, slope, and precipitation. Information about land cover features, which can impact how quickly water is absorbed vs. run off during precipitation events, was sourced from the Multi-Resolution Land Cover Characteristics Consortium's National Land Cover Database (Dewitz, 2019). Hydrologic soil group data, which divides soils into classes based on water infiltration potential, came from the USDA gridded soil survey geographic (GSSURGO) dataset (Soil Survey Staff). All variables had a resolution of 30 meters, except annual precipitation (Fick, 2017), which was resampled at 30 meters from a 5 meter resolution. After removing null values, the final dataset contained 9 million rows. A detailed table of input and target variables and their characteristics can be found in Table 1, and map representations in Figure 2.

Table 1: Variables used in study design

Variable Name	**Description**	**Values**	**Data Source**
x *	Pixel Location	Latitude	ArcGIS Online
y *	Pixel Location	Longitude	ArcGIS Online
elevation	Elevation of point	0 to 255, 0 being the lowest	ArcGIS Online
landcover	Service classes from the National Land Cover	Integers representing	Multi-Resolution Land Cover

	Database	class, see appendix A	Characteristics Consortium
slope_deg	Slope in degrees	0 to 90	ArcGIS Online
imperviousness	Imperviousness of surface as a percent from the National Land Cover Database	Float	Multi-Resolution Land Cover Characteristics Consortium
soilgroup	Soil group type that affect rate at which water is absorbed and the amount that is runoff as a stream	Integer rep. groups A (most well drained) through B, C, D (least), see appendix B	USDA Gridded Soil Survey Geographic (gSSURGO) Database
slope_as	Slope aspect, the direction that the slope on an incline faces	Float (North is ~0/360, South is ~180, Flat is 361)	ArcGIS Online
ann_prcp	Calculated annual mean precipitation in mm based on interpolated station measurements	Float	WorldClim 2.1
flood_den (target)	Magnitude of flood occurrences per cell calculated from ArcGIS using XY point data of historical flood locations	Float	NOAA Storm Events Database

* Not used as input variable for analysis, but kept for mapping purposes.

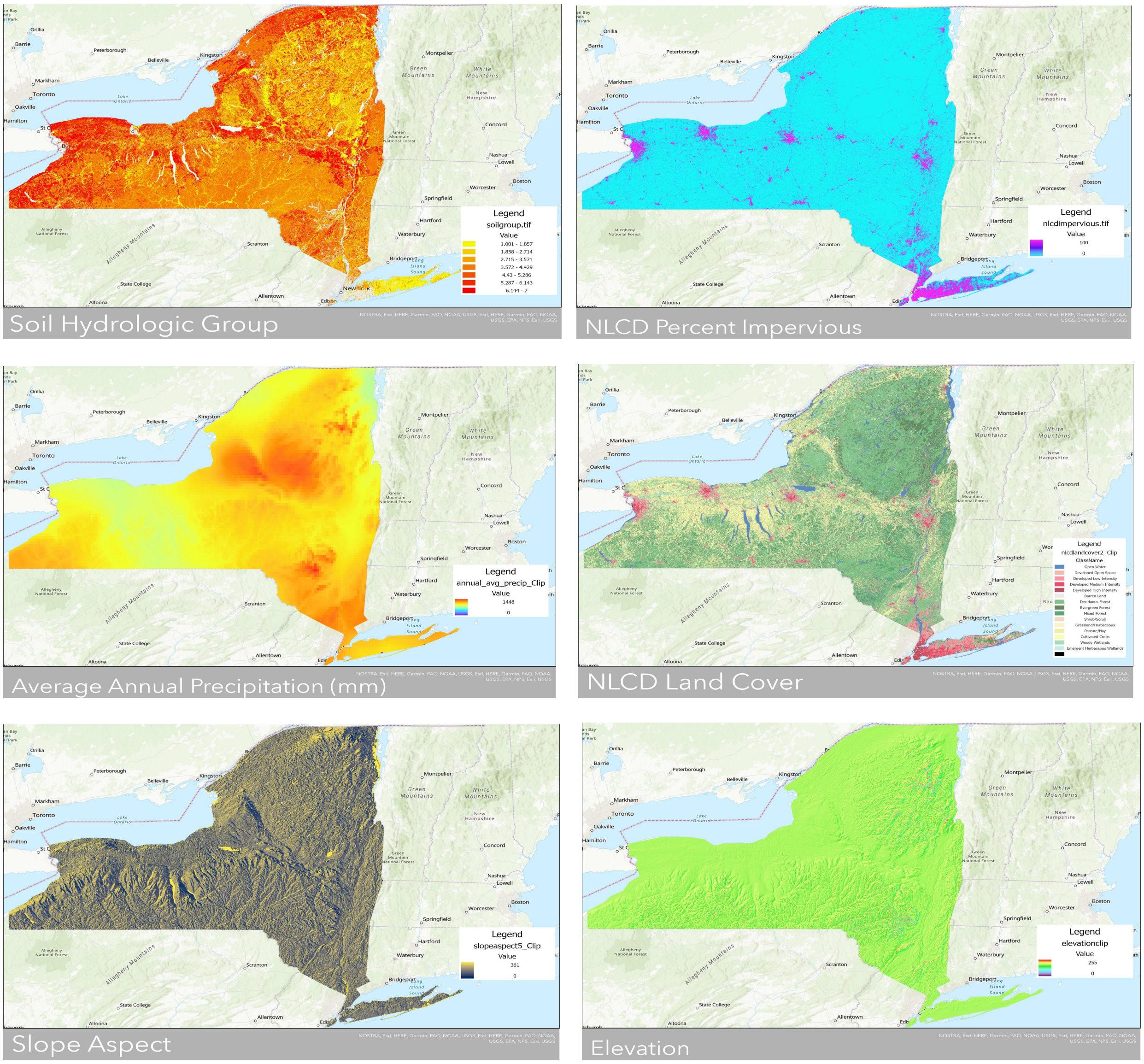


Figure 2. Maps of Input Variables (excluding slope aspect, which is not meaningfully represented in graphical form)

3.2 Data Processing

We can observe from Figures 1 and 3 that most regions in New York have a near zero value of historic flood density so our unprocessed dataset is extremely unbalanced. Additionally, the value per pixel of historic flood density lies on a scale of 0 to about 2500 which makes the outliers (anything greater than 0 really) even more extreme. As a result, we decided to bin the flood density values into 5 discrete classes defined by equal intervals as a means of downscaling the values. After the values were binned, we transformed them back into continuous float values for regression.

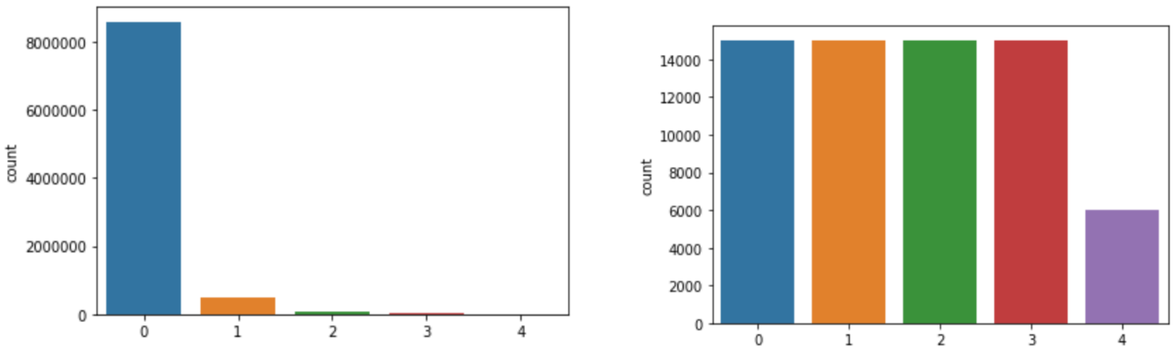


Figure 3. Output variable buckets before (left) and after (right) resampling the lower range

Due to the inflation of zero values in our target flood density and the large size of our dataset, we were able to create a balanced dataset by downsampling so that there are near equal representations of each flood density bin. We trained and evaluated our models on both of these datasets (full and downsampled).

4 Artificial Neural Network Design

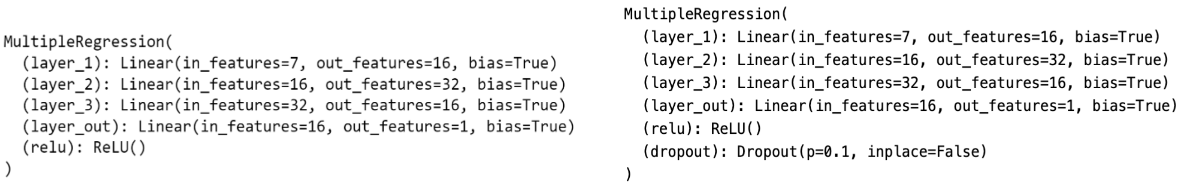
We wrote the code in Python and used PyTorch as our Deep Learning framework to implement our models. We chose PyTorch because it seems to be the most popular Deep Learning framework for conducting research.

We used a Multi-Layer Perceptron (MLP) model to perform a regression on our selected explanatory variables to predict the flood density per area. A Multi-Layer Perceptron is a feed-forward artificial neural network that uses back propagation learning for classification or regression. For this project, we will be using our MLP for regression to predict a continuous variable that has been chunked into smaller buckets. The explanatory variables will be passed into the input layer and then to the hidden layers that aim to find a multi-dimensional expansion of the input layer. The relationship between the output layer and the input layer is created by training the model on a selection of training examples and adjusting the weights of our explanatory variables (Liu, 2023). The training was optimized using a loss function of Mean Squared Error (MSE), also known as the squared L2 norm, which finds the difference between the predicted value and real output, squares it, and averages over the batch. To train the dataset, the data was split into 70-10-20 ratios for training, validation, and testing sets, and several architectures were trained.

4.1 MLP for All Data

The network architecture for the first regression MLPs were fairly straightforward, but quickly caused issues in training. These models were trained on the train set taken from all the data points, rather than the resampled. The structure had an input layer with 7 neutrons that connected to 16, a 16 to 32 neuron hidden layer, and a 32 to 16 neuron hidden layer, and then a final output layer that led to one neuron. Activation functions of ReL Units were between layers. ReLU works as a piecewise function, so that values greater than zero are passed as is, and those at or below zero become zero. This means it is mostly linear and overcomes the vanishing gradient problem, making it a highly used and typical default activation function. The loss function of the MSE was used to train the model and also assess performance in real time. Validation sets were tested at the end of each epoch and the MSE was calculated for the train and validation sets to produce graphs.

Because of initial overfitting concerns and very long run time, we then used the same architecture with dropout regularization added. 10% of the hidden layers' outputs were dropped to prevent possible overfitting to the training data. Additionally, the learning rate and batch size were both adjusted to larger values in hopes of speeding up the training process from the initial 6+ hour runtimes. Both were trained on only 10 epochs due to long run times. Model structures can be seen in Figure 4.



```
MultipleRegression(
  (layer_1): Linear(in_features=7, out_features=16, bias=True)
  (layer_2): Linear(in_features=16, out_features=32, bias=True)
  (layer_3): Linear(in_features=32, out_features=16, bias=True)
  (layer_out): Linear(in_features=16, out_features=1, bias=True)
  (relu): ReLU()
)
```

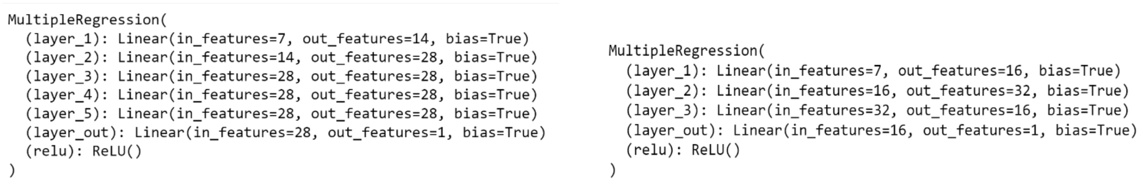
```
MultipleRegression(
  (layer_1): Linear(in_features=7, out_features=16, bias=True)
  (layer_2): Linear(in_features=16, out_features=32, bias=True)
  (layer_3): Linear(in_features=32, out_features=16, bias=True)
  (layer_out): Linear(in_features=16, out_features=1, bias=True)
  (relu): ReLU()
  (dropout): Dropout(p=0.1, inplace=False)
)
```

Figure 4. Architecture of Regression MLPs trained on un-sampled data (dropout on right)

4.2 MLP for Resampled Data

After resampling the lower classes, especially the low-flood risk bucket of 0, to create more even class splits, the resampled data allowed for significantly faster training and validation. This is likely due to the lack of redundancy in the data, which now had less noise but enough meaningful information for the models to learn from. We trained a similar network with the same two hidden layers on the resampled data, without any dropout regularization since we were less worried about overfitting now. The faster run time meant a greater number of epochs was used in training, from 10 before with all data to 42 now with resampled data.

The resampling and significant training time decrease also allowed us to increase the depth of the architecture design. We tested a network with 5 hidden layers, with multiple 28 nodes fully-connected hidden layers, as can be seen in Figure 5.



```
MultipleRegression(
  (layer_1): Linear(in_features=7, out_features=14, bias=True)
  (layer_2): Linear(in_features=14, out_features=28, bias=True)
  (layer_3): Linear(in_features=28, out_features=28, bias=True)
  (layer_4): Linear(in_features=28, out_features=28, bias=True)
  (layer_5): Linear(in_features=28, out_features=28, bias=True)
  (layer_out): Linear(in_features=28, out_features=1, bias=True)
  (relu): ReLU()
)
```

```
MultipleRegression(
  (layer_1): Linear(in_features=7, out_features=16, bias=True)
  (layer_2): Linear(in_features=16, out_features=32, bias=True)
  (layer_3): Linear(in_features=32, out_features=16, bias=True)
  (layer_out): Linear(in_features=16, out_features=1, bias=True)
  (relu): ReLU()
)
```

Figure 5. Architecture of Regression MLPs trained on resampled data (deeper on left)

5 Results

5.1 MLP for All Data

The training results from the unsampled data were very disappointing. The actual MSE values were relatively small, in the 0.08 range. However, the small gradients between each epoch showed that the model was barely learning from each iteration. Additionally, the validation was extremely noisy and did not follow the training curves well. After 10% dropout regularization and larger learning and batch sizes were used, the validation noise slightly decreased (Figure 6b), but the model was still not learning well. In hindsight, we attribute these results to underfitting the data rather than overfitting. Because of the prevalence of 0s in the outcome, the model had high statistical bias towards predicting values closer to zero and failed to learn relevant information about non-zero outcomes. This is reflected in the unstable validation error and lack of learning taking place between epochs.

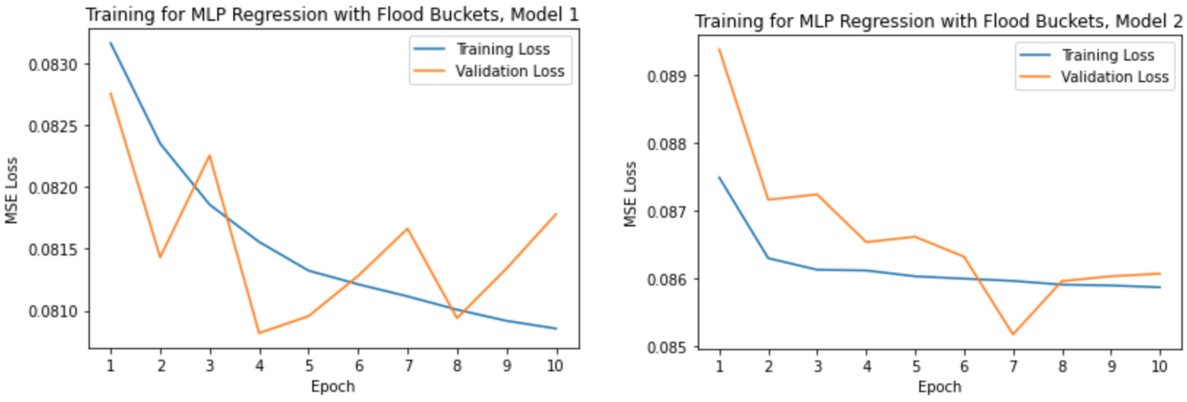


Figure 6. Training results for bucketed unsampled data (10% dropout on the right)

The models were applied to the unseen test data and both MSE and R² were calculated. The first model without dropout regularization had an out-of-sample MSE of 0.0815 and R² of

0.208. The second model with dropout had an out-of-sample MSE of 0.0815 and R² of 0.208. The test MSEs closely reflected the train and validation results, but the combination with the poor R² result shows that the models were not learning the correct patterns or information.

5.2 MLP for Resampled Data

The resampled data yield much better results. Although the values for the MSE were higher, in the range of 0.6-0.8 rather than 0.08, the training and validation curves reflect better learning by the model. Both training and validation loss converge and stabilize after the first 5 epochs, which demonstrates better learning by the model. The validation loss closely follows the training loss, reflecting the model's ability to apply its learning to new data. Figure 7 demonstrates this improvement, with the deeper 5-layer model on the left.

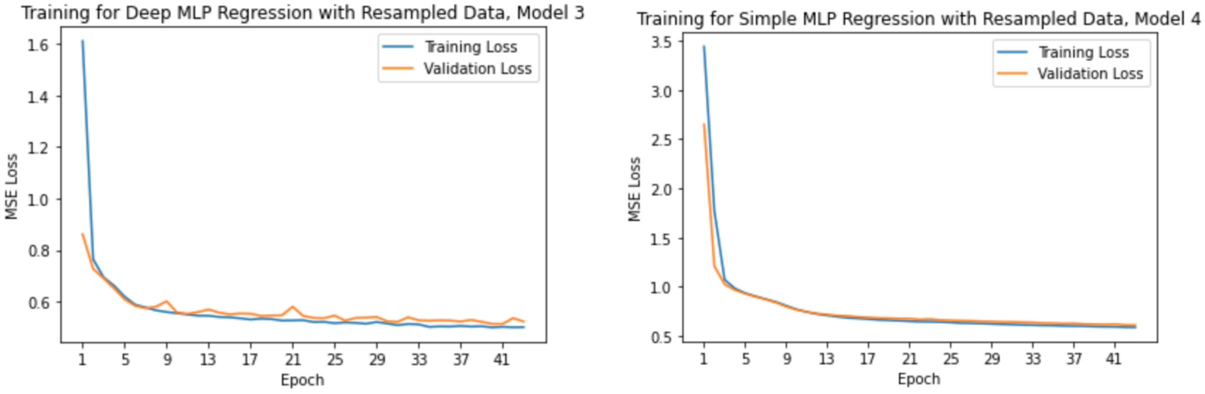


Figure 7. Training results for resampled data (deeper model on the right)

Once again, the models were applied to the unseen test data and both MSE and R² were calculated. The third deep model had an out-of-sample MSE of 0.607 and R² of 0.633. The fourth simpler model had an out-of-sample MSE of 0.522 and R² of 0.685. The MSE again reflected the train and validation metrics, but the change in the R² between predictions and test outcomes showed that these models more accurately learned from the data, once it was resampled to a more even distribution.

5.3 Best Performing Model

To determine the best model, we used a combination of MSE and R² as a metric of success. Based on this, the deeper 5-layer model on resampled data performed the best, with the simpler resampled model as a close second. Although the unsampled models had smaller MSE values, we do not believe that alone accurately reflects performance, as those models consistently underfit the data and did not perform well on test data. A comparison of the network structures and performances can be found in Table 2.

Table 2: Comparison of Test Metrics on All Model Structures

Model Name and Specs	**Architecture of Fully Connected Layers**	**Epochs / Batch Size / Learning Rate**	**Out-of-Sample MSE**	**R²**
1: Unsampled data, low LR and batch size, no dropout	2 hidden layers: 7 nodes to 16, to 32, to 16, to 1 out	10 / 64 / 0.001	0.0815	0.0208
2: Unsampled data, higher LR and batch size, 10% dropout regularization	2 hidden layers: 7 nodes to 16, to 32, to 16, to 1 out	10 / 128 / 0.01	0.0859	0.0166
3: Resampled data, deeper network, more epochs and larger batch size	5 hidden layers: 7 nodes to 14, to 28, to 28, to 28, to 28, to 1 out	42 / 1024 / 0.001	0.607	0.633
4: Resampled data, simpler network, more epochs and larger batch size	2 hidden layers: 7 nodes to 16, to 32, to 16, to 1 out	42 / 1024 / 0.01	0.522	0.685

Last, we used the best-performing deep model to generate a violin plot of predicted values and actual flood density risks, as seen in Figure 8. This visual is a great demonstration of the model's ability to learn, since it was consistently able to accurately predict high flood risk, detonated by resampled values of 2, 3, and 4, and parse that difference from lower risk areas, detonated by 0s and 1s. This shows that this model can be applied to flood density predictions based on geographic and hydraulic input data.

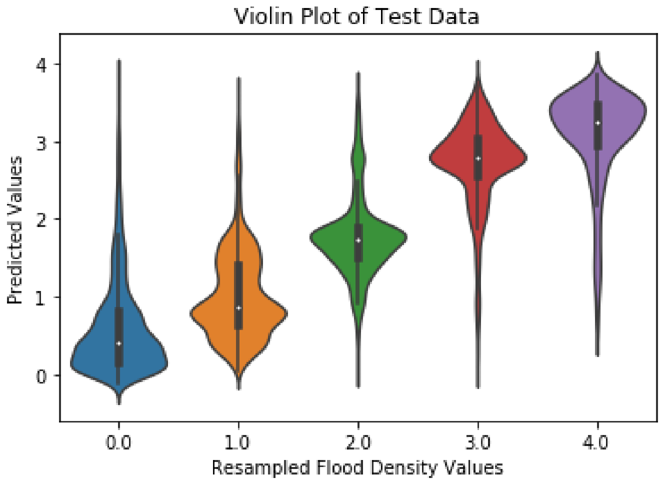


Figure 8. Violin plot for test results of the best performing deeper model on resampled data

6 Discussion and Reflection

We chose this project because we thought it was important to apply our skills as growing data scientists to important real world applications. As a result of that decision, we encountered a large challenge of actually acquiring and structuring the data properly from multiple sources and learning how to use GIS tools. This took a lot more time than we initially anticipated, since the data assembly tasks had a larger learning curve than expected.

After a long data assembly process, we then struggled with being able to actually train regressions models in general. Before we parsed the data into smaller buckets, the training process on unmodified continuous outcome variables took hours and did not converge. Based on this, we scaled the outcome variable to a smaller range of 0 to 4, but as demonstrated with the results above, the model was still extremely underfit. The training process would also take hours with such a large dataset (2 GB in size), so we were encountering computational limits on our devices and times.

Resampling was the key to overcoming many of these challenges, from extremely long training times and slow learning to better performance on validation and test results. This was an important discovery in data assembly and processing, especially when it comes to working with data from natural phenomena like weather outcomes.

Because of this, we learned a lot about working with data, team work, and project design and management. We also learned a lot about the way that neural networks learn and how deep learning models can be improved. We believe that we (Iris and Anna) contributed equally to this project from data preparation to the final presentation of this project. Our individual C scores are 0.95 with a total C of 1.9. We also were very excited when the models worked and are inspired to keep using deep learning methods in future projects.

6.1 Future Directions of Interest

Because of our delayed breakthrough in working with zero-inflated data and developing strategies like resampling, we have many interesting future directions to consider for this project. Data with a large inflation of zeros can be dealt with by resampling like we did or adjusting weights accordingly. Other research has overcome this issue by building a conditional continuous model on top of a binary model (zero or not) first, which meant the combination of models could be used for a semi-continuous case (Diaz, 2019).

Additional future steps for the project would include adjusting the hyperparameters better, such as the number of epochs, learning rates, train: test splits, or regularization, to determine where the best performance lies. Since resampling zero-inflated data vastly reduced the computational time of training the models, using this strategy opens many new paths to applying neural network analysis to severe weather phenomena.

References

Bates, P. D., Quinn, N., Sampson, C., Smith, A., Wing, O., Sosa, J., ... & Krajewski, W. F. (2021). Combined modeling of US fluvial, pluvial, and coastal flood hazard under current and future climates. *Water Resources Research*, *57*(2), e2020WR028673.

Betts, A. (2023, July 12). *What to know about Vermont's devastating floods*. The New York Times. https://www.nytimes.com/2023/07/12/us/vermont-flooding-rain-forecast.html

Dewitz, J., 2019, National Land Cover Database (NLCD) 2016 Products: U.S. Geological Survey data release, https://doi.org/10.5066/P96HHBIE.

Diaz, J., & Joseph, M. B. (2019). Predicting property damage from tornadoes with zero-inflated neural networks. *Weather and Climate Extremes*, *25*, 100216.

Fick, S.E. and R.J. Hijmans (2017). WorldClim 2: new 1km spatial resolution climate surfaces for global land areas. International Journal of Climatology 37 (12): 4302-4315.

Liu, Yuntao (2023). Data 612 Lectures. *Science Academy Deep Learning Course*.

National Centers for Environmental Information. Storm Events Database. 1997-2017. Dataset Identifiers: NCEI DSI 3910_03, gov.noaa.Ncdc:C00510.

Picard, C. J., Winter, J. M., Cockburn, C., Hanrahan, J., Teale, N. G., Clemins, P. J., & Beckage, B. (2023). Twenty-first century increases in total and extreme precipitation across the Northeastern USA. *Climatic Change*, *176*(6), 1-26.

Soil Survey Staff. Gridded Soil Survey Geographic (gSSURGO) Database for *State name*. United States Department of Agriculture, Natural Resources Conservation Service. https://gdg.sc.egov.usda.gov/

Walker K, Herman M (2023). *tidycensus: Load US Census Boundary and Attribute Data as 'tidyverse' and 'sf'-Ready Data Frames*. R package version 1.4.3, https://walker-data.com/tidycensus/.

Wickham H, Averick M, Bryan J, Chang W, McGowan LD, François R, Grolemund G, Hayes A, Henry L, Hester J, Kuhn M, Pedersen TL, Miller E, Bache SM, Müller K, Ooms J, Robinson D, Seidel DP, Spinu V, Takahashi K, Vaughan D, Wilke C, Woo K, Yutani H (2019). "Welcome to the tidyverse." *Journal of Open Source Software*, **4**(43), 1686. doi:10.21105/joss.01686.

Appendix A

National Land Cover Database Service Classes

11. Open Water - areas of open water, generally with less than 25% cover of vegetation or soil.

12. Perennial Ice/Snow - areas characterized by a perennial cover of ice and/or snow, generally greater than 25% of total cover.

21. Developed, Open Space - areas with a mixture of some constructed materials, but mostly vegetation in the form of lawn grasses. Impervious surfaces account for less than 20% of total cover. These areas most commonly include large-lot single-family housing units, parks, golf courses, and vegetation planted in developed settings for recreation, erosion control, or aesthetic purposes.

22. Developed, Low Intensity - areas with a mixture of constructed materials and vegetation. Impervious surfaces account for 20% to 49% percent of total cover. These areas most commonly include single-family housing units.

23. Developed, Medium Intensity - areas with a mixture of constructed materials and vegetation. Impervious surfaces account for 50% to 79% of the total cover. These areas most commonly include single-family housing units.

24. Developed High Intensity - highly developed areas where people reside or work in high numbers. Examples include apartment complexes, row houses and commercial/industrial. Impervious surfaces account for 80% to 100% of the total cover.

31. Barren Land (Rock/Sand/Clay) - areas of bedrock, desert pavement, scarps, talus, slides, volcanic material, glacial debris, sand dunes, strip mines, gravel pits and other accumulations of earthen material. Generally, vegetation accounts for less than 15% of total cover.

41. Deciduous Forest - areas dominated by trees generally greater than 5 meters tall, and greater than 20% of total vegetation cover. More than 75% of the tree species shed foliage simultaneously in response to seasonal change.

42. Evergreen Forest - areas dominated by trees generally greater than 5 meters tall, and greater than 20% of total vegetation cover. More than 75% of the tree species maintain their leaves all year. Canopy is never without green foliage.

43. Mixed Forest - areas dominated by trees generally greater than 5 meters tall, and greater than 20% of total vegetation cover. Neither deciduous nor evergreen species are greater than 75% of total tree cover.

51. Dwarf Scrub - Alaska only areas dominated by shrubs less than 20 centimeters tall with shrub canopy typically greater than 20% of total vegetation. This type is often co-associated with grasses, sedges, herbs, and non-vascular vegetation.

52. Shrub/Scrub - areas dominated by shrubs; less than 5 meters tall with shrub canopy typically greater than 20% of total vegetation. This class includes true shrubs, young trees in an early successional stage or trees stunted from environmental conditions.

71. Grassland/Herbaceous - areas dominated by gramanoid or herbaceous vegetation, generally greater than 80% of total vegetation. These areas are not subject to intensive management such as tilling, but can be utilized for grazing.

72. Sedge/Herbaceous - Alaska only areas dominated by sedges and forbs, generally greater than 80% of total vegetation. This type can occur with significant other grasses or other grass like plants, and includes sedge tundra, and sedge tussock tundra.

73. Lichens - Alaska only areas dominated by fruticose or foliose lichens generally greater than 80% of total vegetation.

74. Moss - Alaska only areas dominated by mosses, generally greater than 80% of total vegetation.

81. Pasture/Hay - areas of grasses, legumes, or grass-legume mixtures planted for livestock grazing or the production of seed or hay crops, typically on a perennial cycle. Pasture/hay vegetation accounts for greater than 20% of total vegetation.

82. Cultivated Crops - areas used for the production of annual crops, such as corn, soybeans, vegetables, tobacco, and cotton, and also perennial woody crops such as orchards and vineyards. Crop vegetation accounts for greater than 20% of total vegetation. This class also includes all land being actively tilled.

90. Woody Wetlands - areas where forest or shrubland vegetation accounts for greater than 20% of vegetative cover and the soil or substrate is periodically saturated with or covered with water.

95. Emergent Herbaceous Wetlands - Areas where perennial herbaceous vegetation accounts for greater than 80% of vegetative cover and the soil or substrate is periodically saturated with or covered with water.

Appendix B

Hydrologic soil group key

Group A (1) - Group A soils consist of deep, well drained sands or gravelly sands with high infiltration and low runoff rates.
Group B (2)- Group B soils consist of deep well drained soils with a moderately fine to moderately coarse texture and a moderate rate of infiltration and runoff.
Group C (3)- Group C consists of soils with a layer that impedes the downward movement of water or fine textured soils and a slow rate of infiltration.
Group D (4)- Group D consists of soils with a very slow infiltration rate and high runoff potential. This group is composed of clays that have a high shrink-swell potential, soils with a high water table, soils that have a clay pan or clay layer at or near the surface, and soils that are shallow over nearly impervious material.
Group A/D (5) - Group A/D soils naturally have a very slow infiltration rate due to a high water table but will have high infiltration and low runoff rates if drained.
Group B/D (6)- Group B/D soils naturally have a very slow infiltration rate due to a high water table but will have a moderate rate of infiltration and runoff if drained.
Group C/D (7)- Group C/D soils naturally have a very slow infiltration rate due to a high water table but will have a slow rate of infiltration if drained.