Unveiling the Complex Dynamics of College Admissions:

Comprehensive Predictive Analysis of University Acceptance Rates Using Advanced Statistical and Machine Learning Methods

Yiran Yao

Literature Review

- Limited amount of studies
- Studies discuss limited predictors

Methodology

1. Data Cleaning
2. EDA
3. Modelling

I. Multi-Variable
II. LASSO
III. Polynomial Regression
IV. Trees
V. Neural Networks (NN)
VI. CNN
VII. Autoencoder
VIII. Resnet50
IX. DenseNet

4. Model Comparison
5. Limitations

Data Sources

College Admissions

College Admissions Admission/Class Demographics by University **– Kaggle**

Racial and Ethnic Representativeness of US Postsecondary Education Institutions

— Urban Catalog

US College Data

Based on US Colleges listing of whether the College is Public or Private – **Kaggle**

“American University Data” IPEDS dataset

– **Kaggle**

Data Cleaning

1. Filtering Data

Renaming Columns

Finding Common Universities in Datasets

2. Aggregating data (Averaging Numeric Values)

Grouped by University:

- [Percent of freshmen receiving student loan aid]
- [Enrolled total]
- [Expend] (Expenditure of students)
- …

3. Merging Data

488 Institutions (Rows)

164 Variables (Columns)

4. Filtering

I. Removing rows and columns when missing values >= 50%

II. **Overfitting ->** Manually selecting variables (60)

5. Filling in Missing Values for 3 Variables

I. Training NN to represent the following:

I. [ACT Composite 75th percentile score]

II. [Endowment Asset per FTE enrollment]

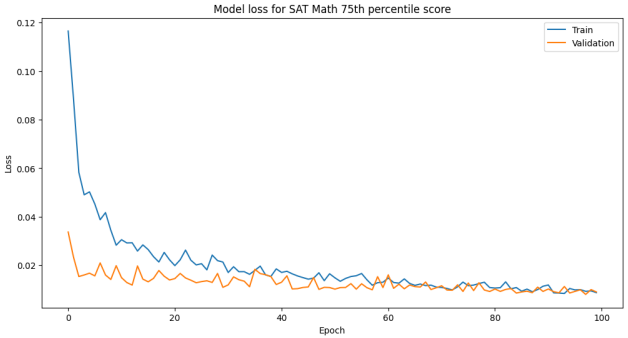
III. [SAT Math 75% percentile score]

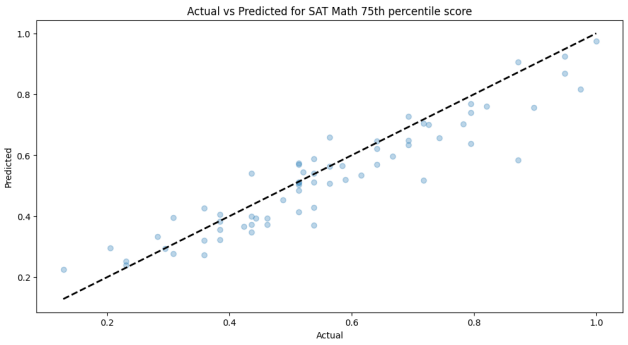
6. Final Dataset

I. [Tuition and Fees] [Percent of freshmen receiving any financial aid] [Grad.Rate]…

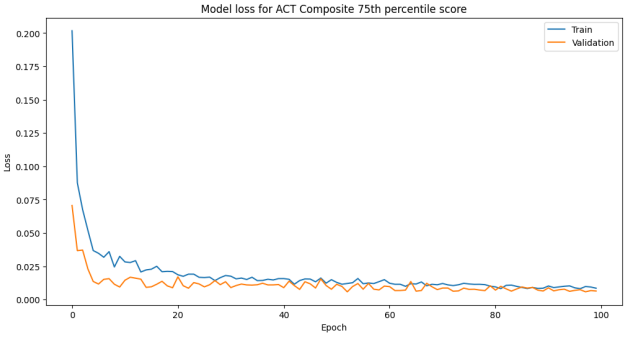
Neural Networks - Filling in Missing Data

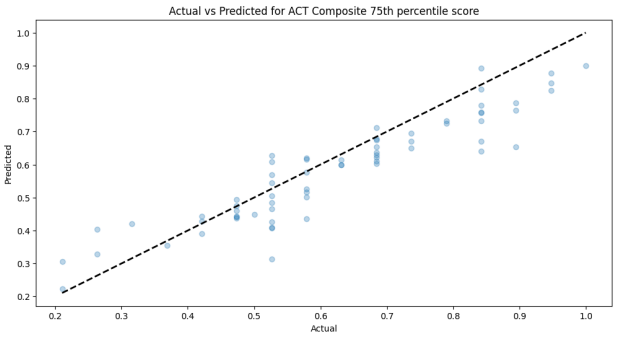
ACT Composite 75th percentile score



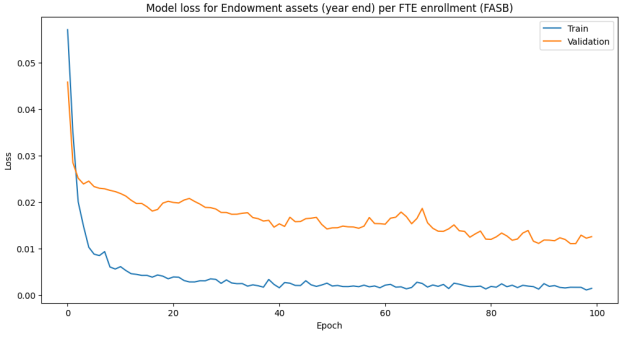


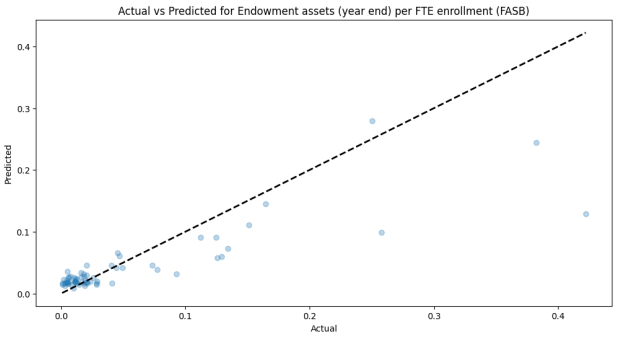
Endowment Asset (year end) per FTE enrollment (FASB)

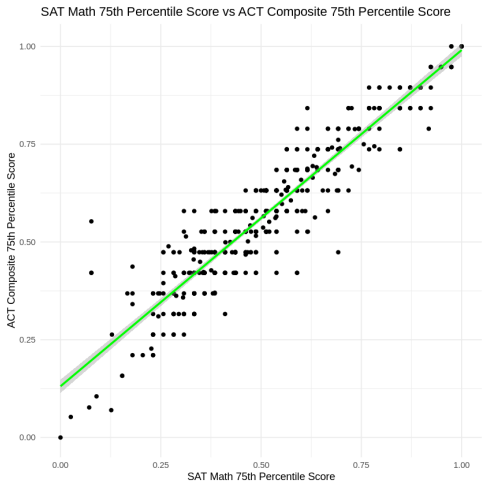


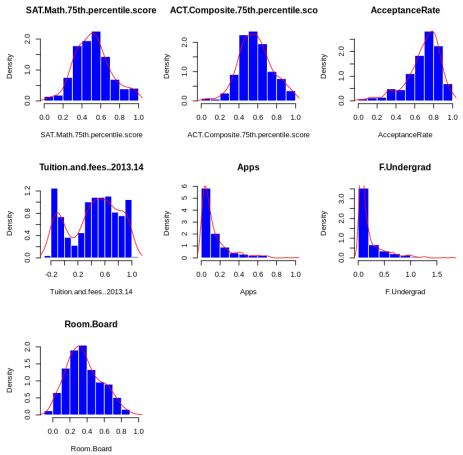


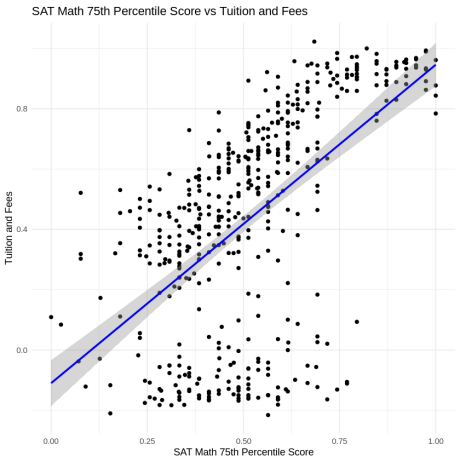
SAT Math 75% percentile score






SAT Math 75th Percentile Score vs ACT Composite 75th Percentile Score
ACT Composite 75th Percentile Score
SAT Math 75th Percentile Score
0.00
0.25
0.50
0.75
1.00

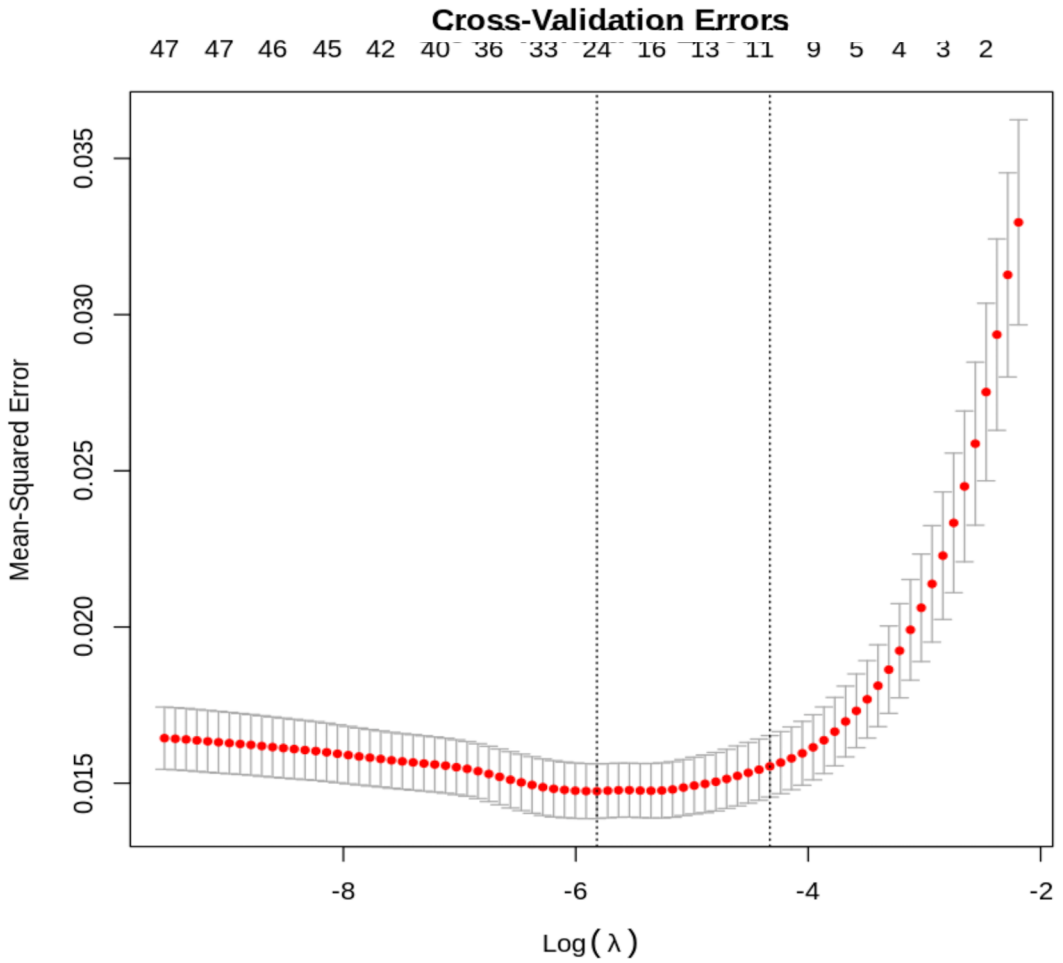

SAT.Math.75th.percentile.score
ACT.Composite.75th.percentile.sco
AcceptanceRate
Tuition.and.fees..2013.14
Apps
F.Undergrad
Room.Board
Density

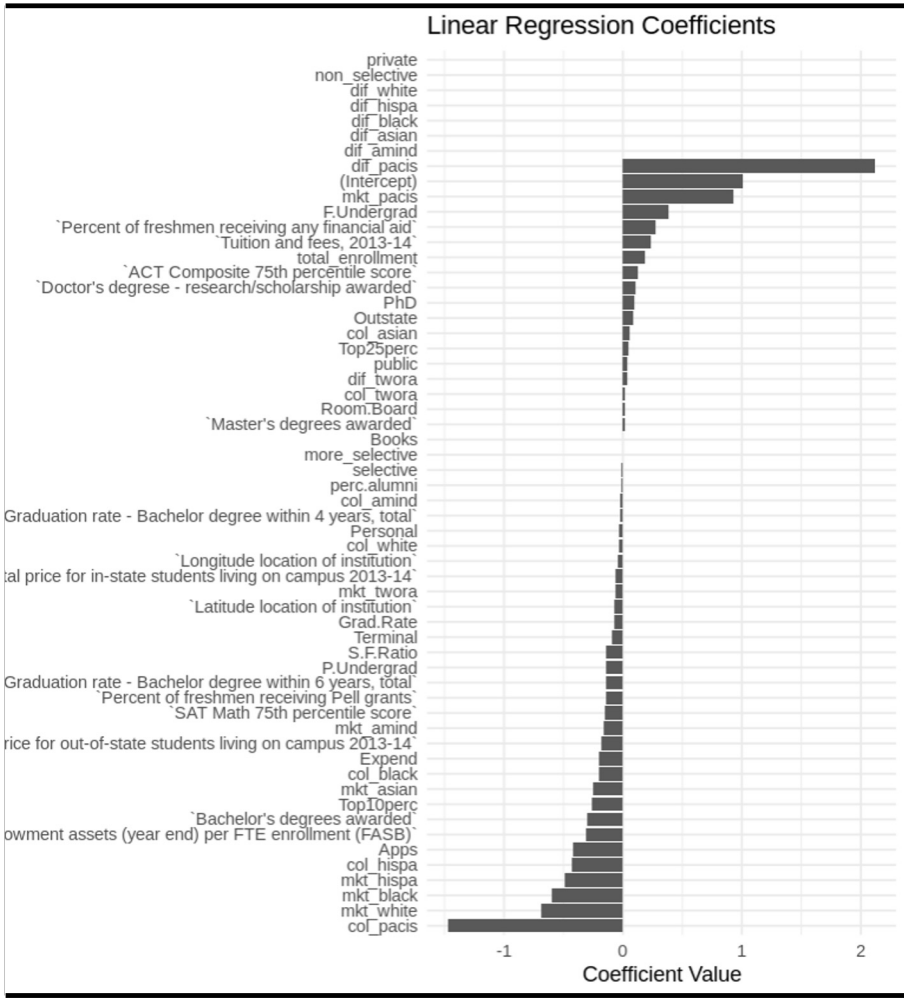

SAT Math 75th Percentile Score vs Tuition and Fees
Tuition and Fees
SAT Math 75th Percentile Score
0.0
0.4
0.8
0.00
0.25
0.50
0.75
1.00


SAT.Math.75th.percentile.score
ACT.Composite.75th.percentile.score
AcceptanceRate
Tuition.and.fees..2013.14
Apps
F.Undergrad
Room.Board
1 0.93 -0.49 0.56 0.49 0.26 0.45
0.93 1 -0.44 0.61 0.41 0.17 0.46
-0.49 -0.44 1 -0.26 -0.46 -0.14 -0.34
0.56 0.61 -0.26 1 -0.02 -0.38 0.63
0.49 0.41 -0.46 -0.02 1 0.79 0.27
0.26 0.17 -0.14 -0.38 0.79 1 -0.04
0.45 0.46 -0.34 0.63 0.27 -0.04 1
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1

Dimension Reduction - LASSO

- Using lasso.min: 24 significant variables



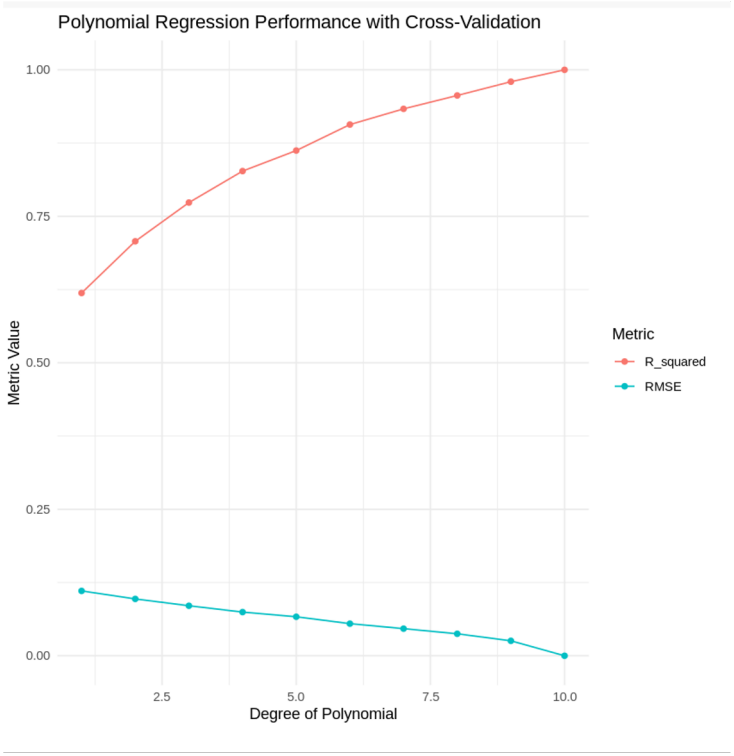


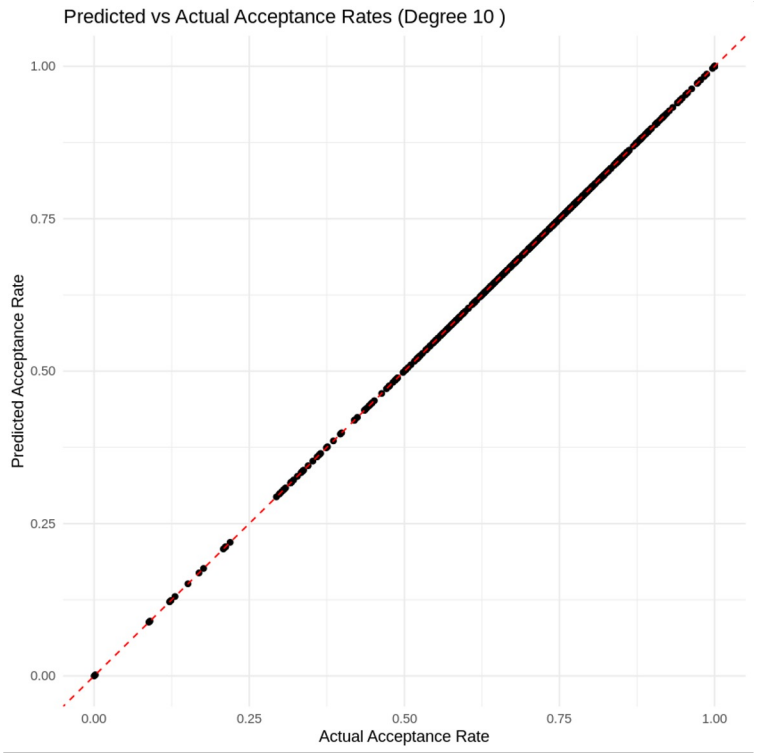
Linear Regression

24 variables that influence admission rates the most:

- [% of Freshmen Receiving Financial Aid]
- [Endowment Assets]
- [Asian in pool- accepted]
- Etc.….

Polynomial Regression





MAE for each degree on the testing data:

Degree 1 : MAE = 0.0940625091793621

Degree 2 : MAE = 0.101738381062515

Degree 3 : MAE = 0.18267447467166

Degree 4 : MAE = 1.62755161993617

Degree 5 : MAE = 17.6003093597632

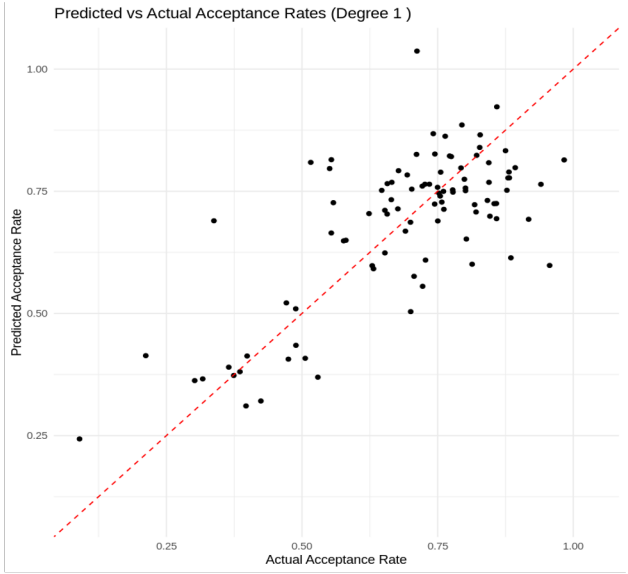
Degree 6 : MAE = 333.983189370729

Degree 7 : MAE = 2093.35970995474

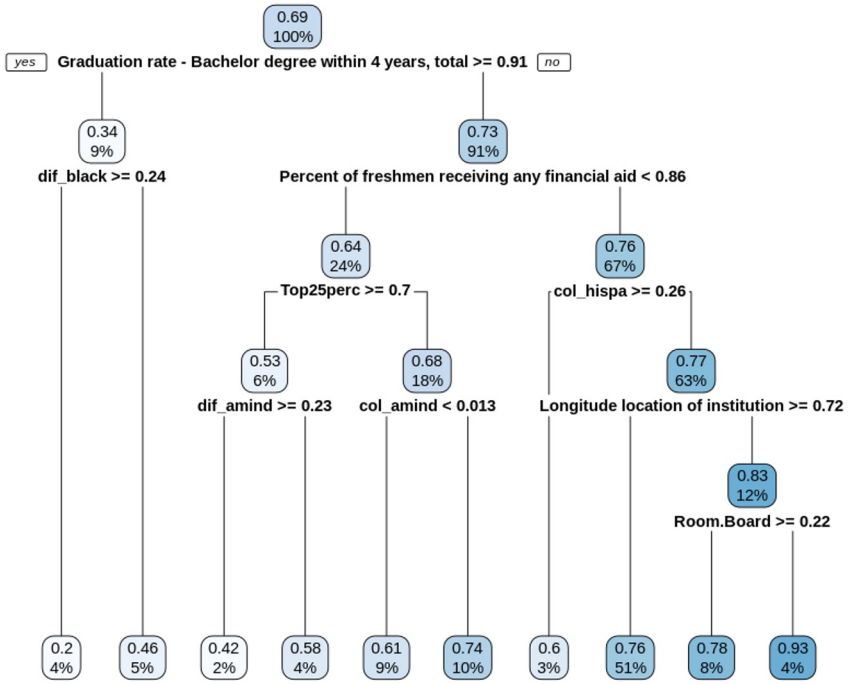
Degree 8 : MAE = 367503.043848007

Degree 9 : MAE = 10667.2486167903

Degree 10 : MAE = 11891.030960912



Overfitting



Random Forest

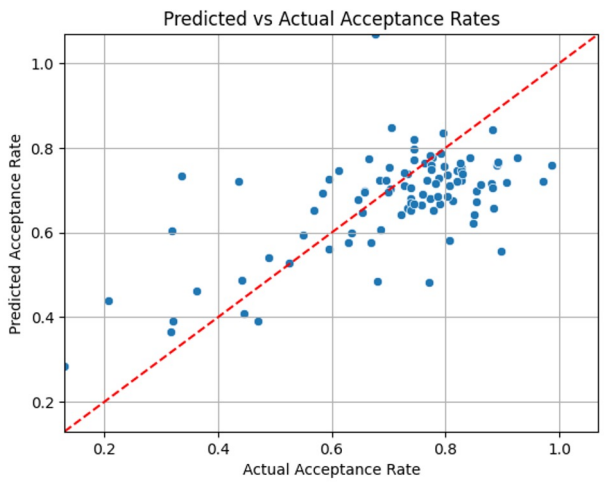
Selected Variables: feature importance ranking

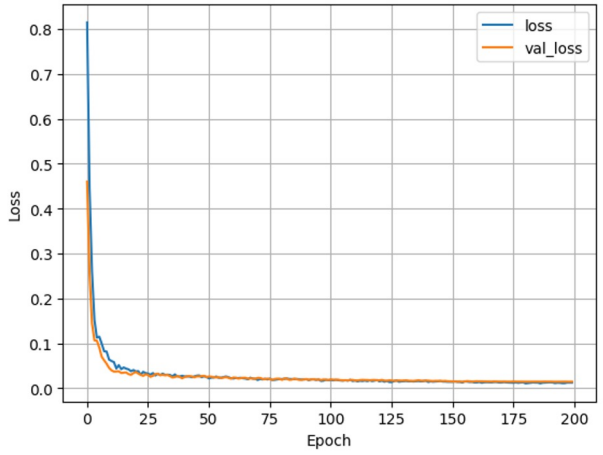
Feedforward Neural Network

Model Architecture:

- Input Layer: 64 neurons, ReLU activation
- Hidden Layers:
  - 64 neurons + Dropout (0.2)
  - 32 neurons + Dropout (0.2)
  - 16 neurons
- Output Layer: 1 neuron

- **Compilation**: Optimizer: Adam
- Loss Function: Mean Squared Error (MSE)
- Metric: Mean Absolute Error (MAE)
- **Training**: Epochs: 200
- Batch Size: 64
- Validation Split: 20%

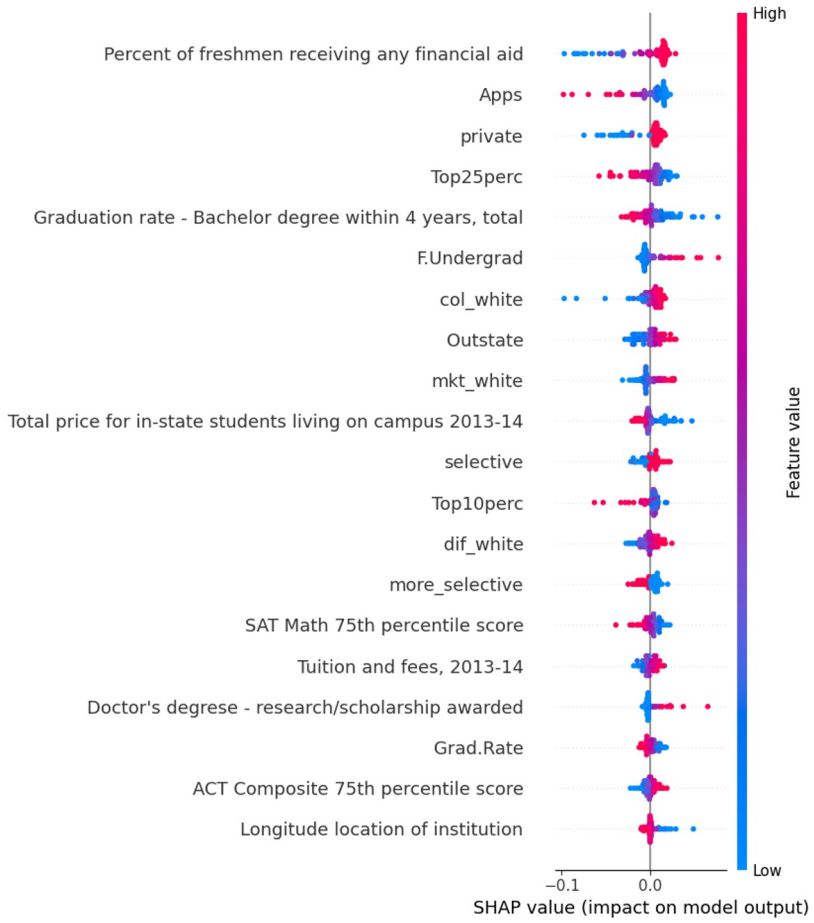


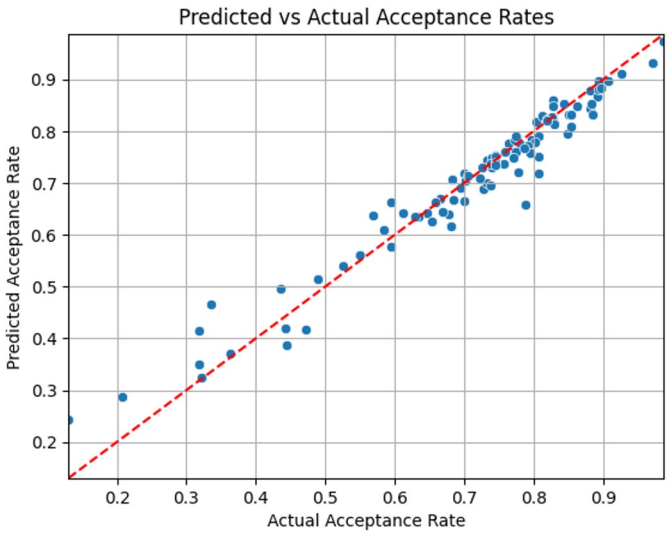


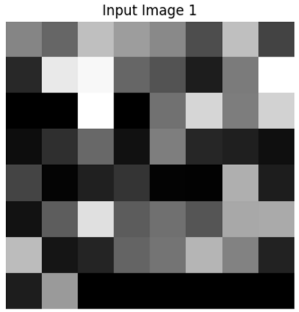
Layer (type)	Output Shape	Param #
dense (Dense)	(None, 64)	3,712
dropout (Dropout)	(None, 64)	0
dense_1 (Dense)	(None, 32)	2,080
dropout_1 (Dropout)	(None, 32)	0
dense_2 (Dense)	(None, 16)	528
dense_3 (Dense)	(None, 1)	17

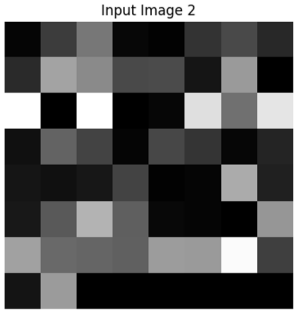
Total params: 6,337 (24.75 KB)

SHAP


Percent of freshmen receiving any financial aid
Apps
private
Top25perc
Graduation rate - Bachelor degree within 4 years, total
F.Undergrad
col_white
Outstate
mkt_white
Total price for in-state students living on campus 2013-14
selective
Top10perc
dif_white
more_selective
SAT Math 75th percentile score
Tuition and fees, 2013-14
Doctor's degree - research/scholarship awarded
Grad.Rate
ACT Composite 75th percentile score
Longitude location of institution
High
Feature value
Low
−0.1
0.0
SHAP value (impact on model output)







Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 6, 6, 32)	320
max_pooling2d (MaxPooling2D)	(None, 3, 3, 32)	0
conv2d_1 (Conv2D)	(None, 1, 1, 64)	18,496
flatten (Flatten)	(None, 64)	0
dense_4 (Dense)	(None, 64)	4,160
dropout_2 (Dropout)	(None, 64)	0
dense_5 (Dense)	(None, 1)	65

- **Data Preprocessing**:
  - Normalized data using MinMaxScaler.
  - Transformed data into a matrix and reshaped it to 2D.
  - Padded data to match the nearest perfect square size

CNN

- **Model Architecture**:
  - Conv2D Layer 1: 32 filters, (3, 3) kernel, ReLU activation.
  - MaxPooling2D Layer: (2, 2) pool size.
  - Conv2D Layer 2: 64 filters, (3, 3) kernel, ReLU activation.
  - Flatten Layer.
  - Dense Layer 1: 64 neurons, ReLU activation.
  - Dropout Layer: 0.5 rate.
  - Dense Output Layer: 1 neuron

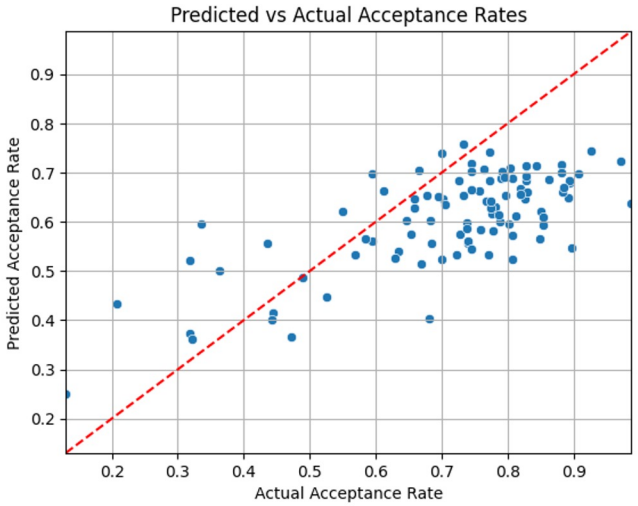
- **Training**:
  - Epochs: 100.
  - Batch Size: 32.
  - Validation Split: 20%.

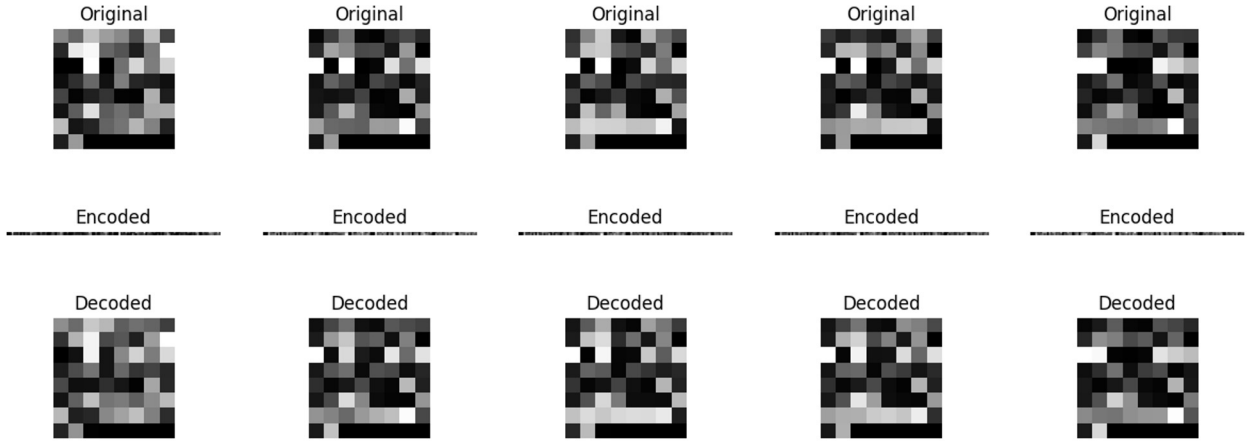
SHAP

Autoencoder

Model Architecture

- **Encoder**: Reduces data dimensions using convolutional layers.
- **Decoder**: Reconstructs the input data from encoded features.
- **Classifier**: Uses encoded features to predict acceptance rates.

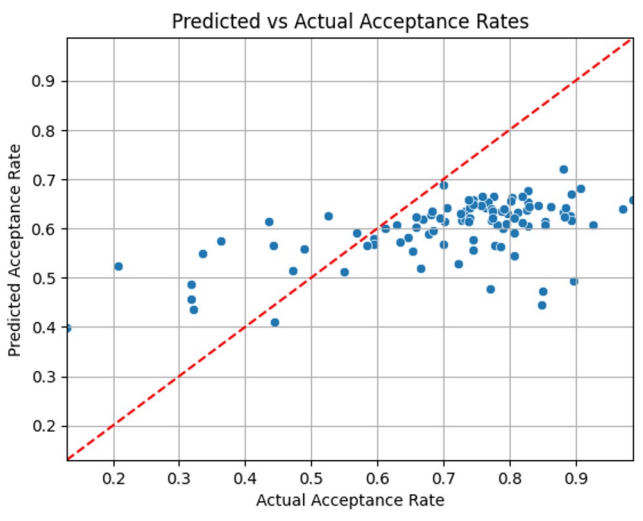


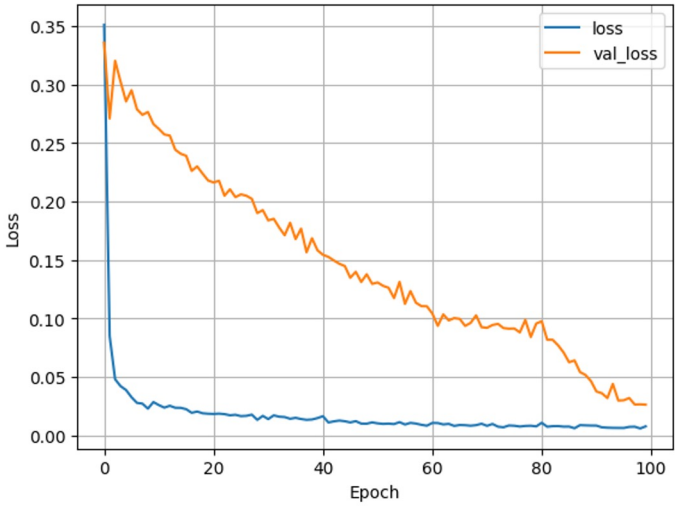


Resnet50

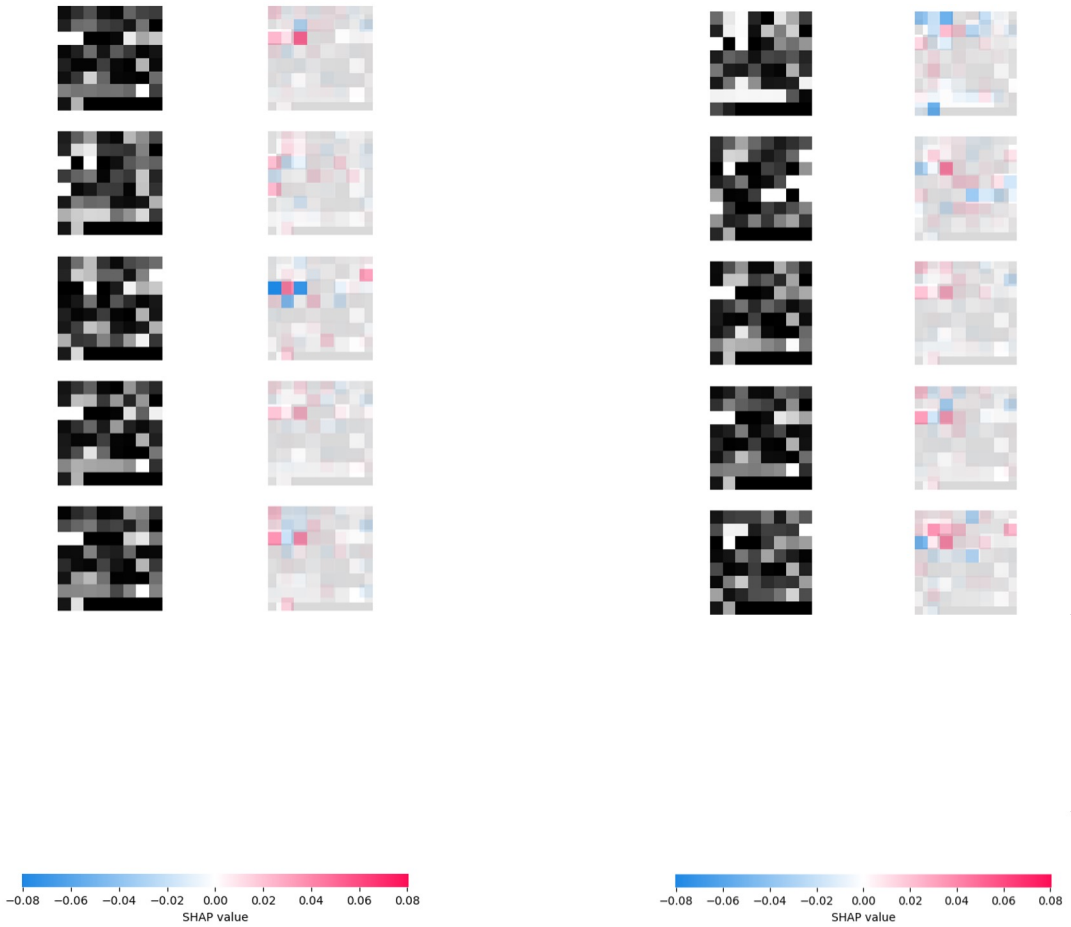
- **Model Architecture**:
  - **Input Layer**: Reshaped 2D data.
  - **ResNet50 Base Model**: Without pre-trained weights, top layers excluded.
  - **Global Average Pooling Layer**.
  - **Dense Layers**:
  - 1024 neurons, ReLU activation, Dropout (0.5).
  - 512 neurons, ReLU activation, Dropout (0.5).
  - 256 neurons, ReLU activation.
  - **Output Layer**: 1 neuron.

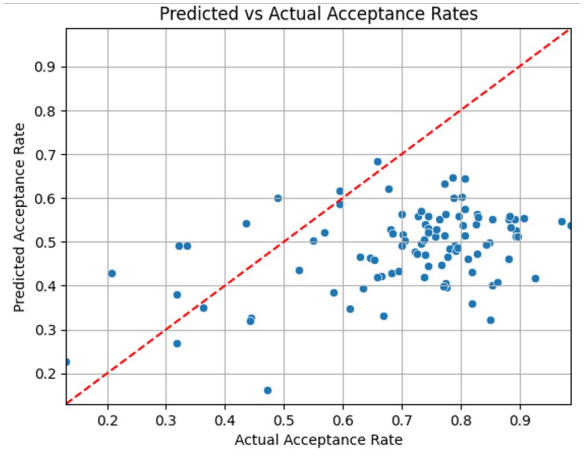
- **Training**:
  - Epochs: 100.
  - Batch Size: 32.
  - Validation Split: 20%.
  - Optimizer: Adam (learning rate 0.0001)

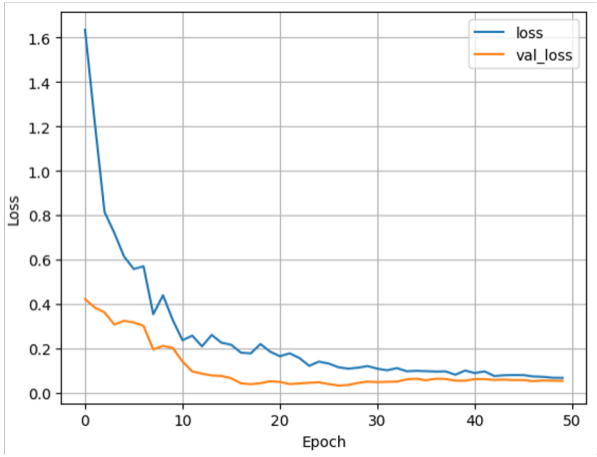




SHAP


−0.08 −0.06 −0.04 −0.02 0.00 0.02 0.04 0.06 0.08
SHAP value
−0.08 −0.06 −0.04 −0.02 0.00 0.02 0.04 0.06 0.08
SHAP value





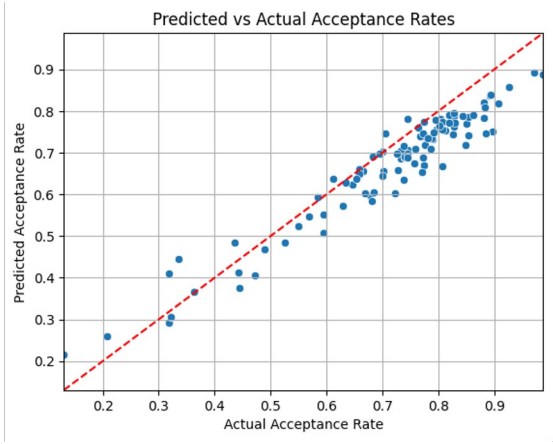
DenseNet

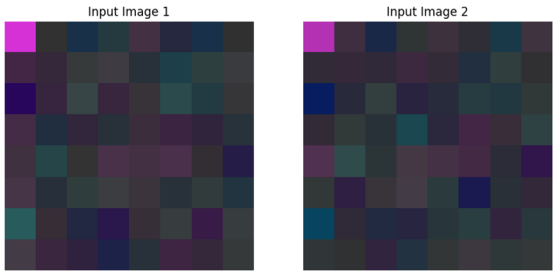
•Model Architecture:

- **Input Layer**: Reshaped 2D data.
- **DenseNet Base Model**: Without pre-trained weights, top layers excluded.
- **Global Average Pooling Layer**.
- **Dense Layers**:
  - 1024 neurons, ReLU activation, Dropout (0.5).
  - 512 neurons, ReLU activation, Dropout (0.5).
  - 256 neurons, ReLU activation.
- **Output Layer**: 1 neuron.

•Training:

Epochs: 50.
Batch Size: 32.
Validation Split: 20%.
Optimizer: Adam (learning rate 0.0001)





$$F(u,v) = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x,y) \cdot e^{-j2\pi\left(\frac{ux}{M}+\frac{vy}{N}\right)}$$

CNN(Modified)

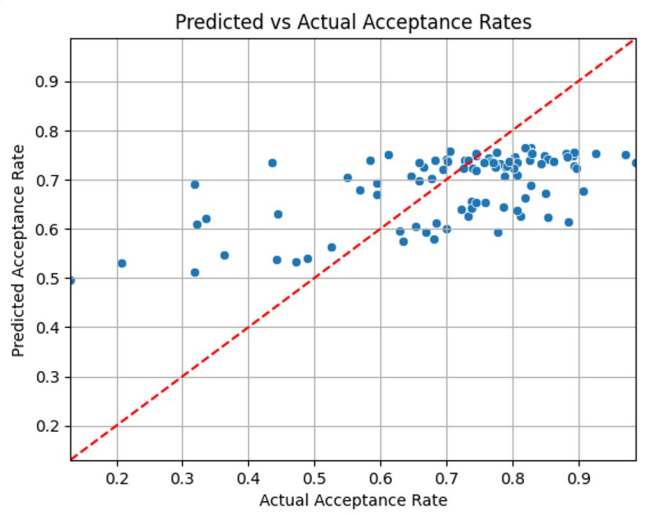
Utilizing Fourier Transform on input data for enhanced feature extraction.

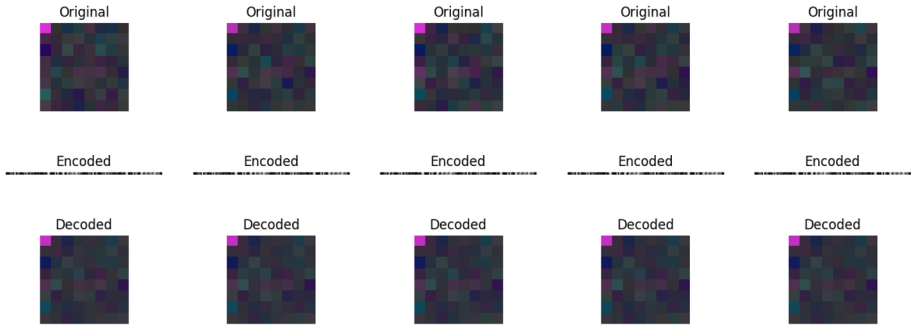
Converting features into RGB format for CNN input

Autoencoder (Modified)

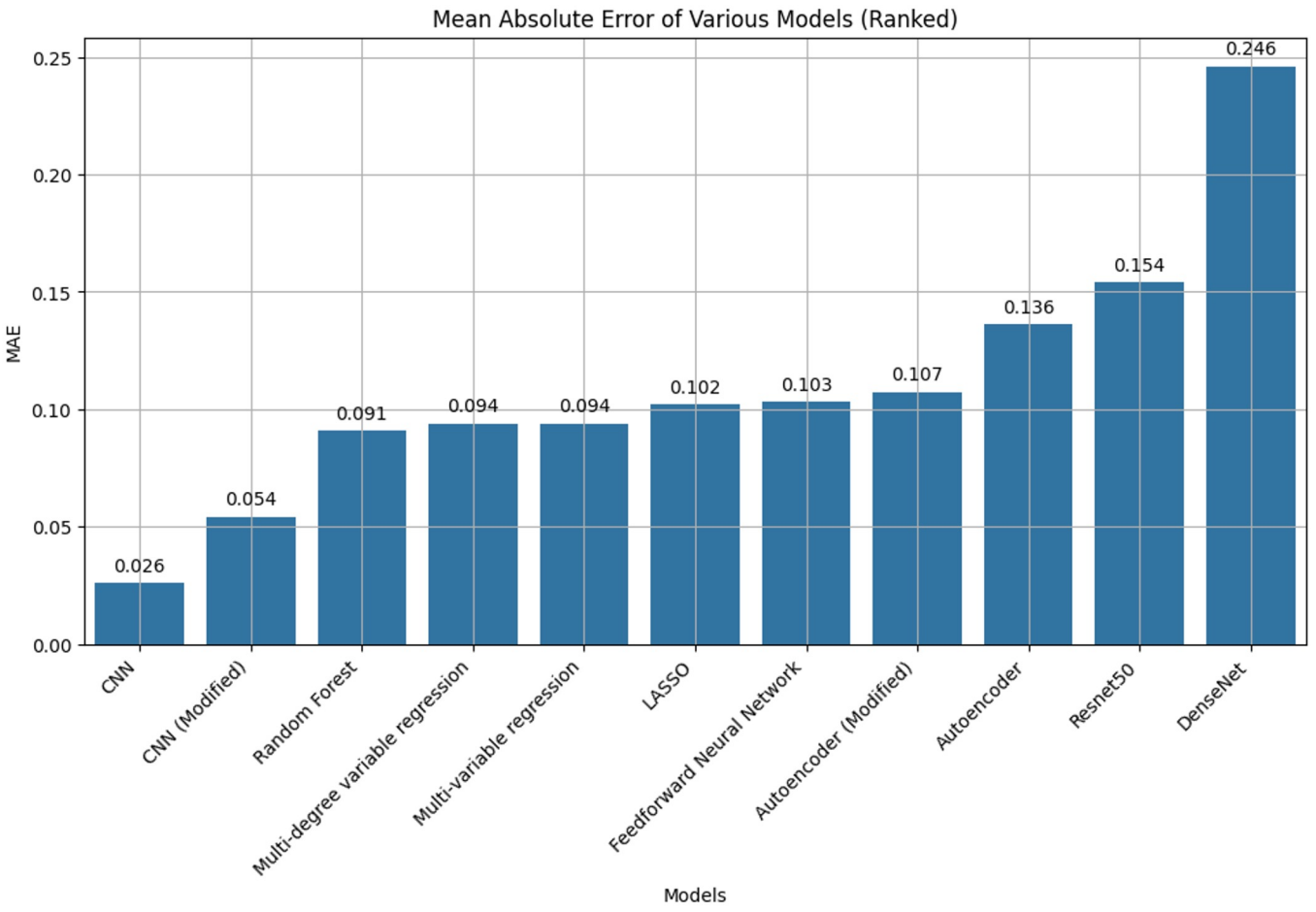
Utilizing Fourier Transform on input data for enhanced feature extraction.

Converting features into RGB format for CNN input





Model Comparison



Limitations

- University-Level Analysis: Student-level data is confidential
- Less Variable Explanation: Complications of neural networks

Citations

Lundberg, S. M., & Lee, S.-I. (2017). A Unified Approach to Interpreting Model Predictions. In Advances in Neural Information Processing Systems (NeurIPS) (pp. 4765-4774). Retrieved from https://arxiv.org/abs/1705.07874

Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep Learning. MIT Press. Retrieved from https://www.deeplearningbook.org/

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep Learning. Nature, 521(7553), 436-444. doi:10.1038/nature14539

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep Residual Learning for Image Recognition. In 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (pp. 770-778). Retrieved from https://arxiv.org/abs/1512.03385

Huang, G., Liu, Z., Van Der Maaten, L., & Weinberger, K. Q. (2017). Densely Connected Convolutional Networks. In 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (pp. 2261-2269). Retrieved from https://arxiv.org/abs/1608.06993

Thank you