MIT MANAGEMENT SLOAN SCHOOL

LANGUAGE MODELING USING TREE-BASED METHODS

15.095 ML under a Modern Optimization Lens – Final Project

Team Members:
Seth Chatterton, Jason Jia

PROBLEM AND MOTIVATION

- Neural networks are highly popular as models of choice for autoregressive language modeling (predicting the next token/character/word).
- However, they are computationally expensive to train.
- On the other hand, tree-based models such as XGBoost, RF, CART and OCT can be easier to train and might be able to generate similar predictions
  - CART and OCT can even be interpretable
- This is also in light of the proof in class that neural networks are equivalent to trees under certain assumptions.
- We would thus like to experiment with tree-based methods and compare their performance with established neural network-based methods for autoregressive language modeling.

DATA

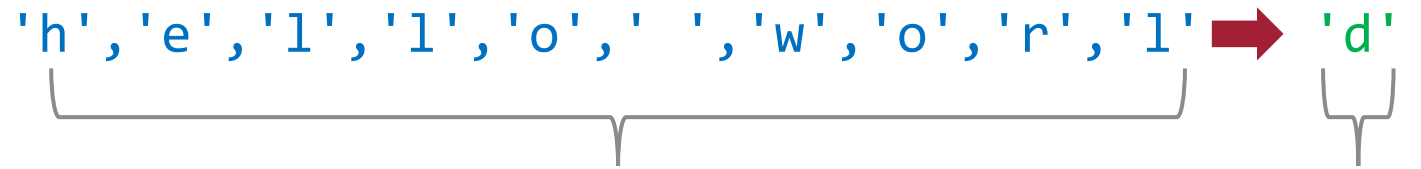
- **Cleaned Alpaca Dataset**, a slightly modified version of the dataset used to fine-tune the Alpaca large language model.
- The data consists of instructions for the model, inputs for those instructions, and outputs that the model should try to recreate.


```
{
    "instruction": "Give three tips for staying healthy.",
    "input": "",
    "output": "1. Eat a balanced and nutritious diet: Make sure your meals are inclusive of a variety of fruits and vegetables,
},
{
    "instruction": "What are the three primary colors?",
    "input": "",
    "output": "The three primary colors are red, blue, and yellow. These colors are called primary because they cannot be create
},
```

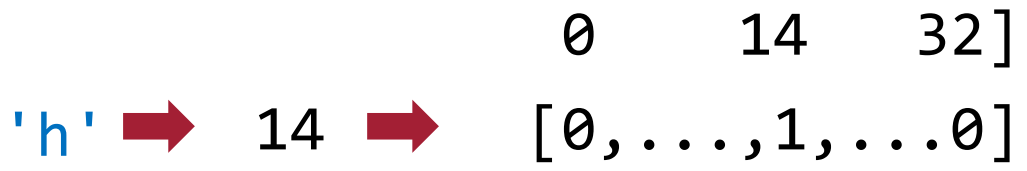
METHODS

- Preprocessing
- Tree-Based Methods
  - CART, Random Forest, XGBoost
- Neural Network Methods (Benchmark)
  - Feedforward Neural Networks
- Evaluation
  - Training and Test Accuracy
  - Training Time
  - Interpretability

Tokenization: individual characters



One-hot encoded vectors:

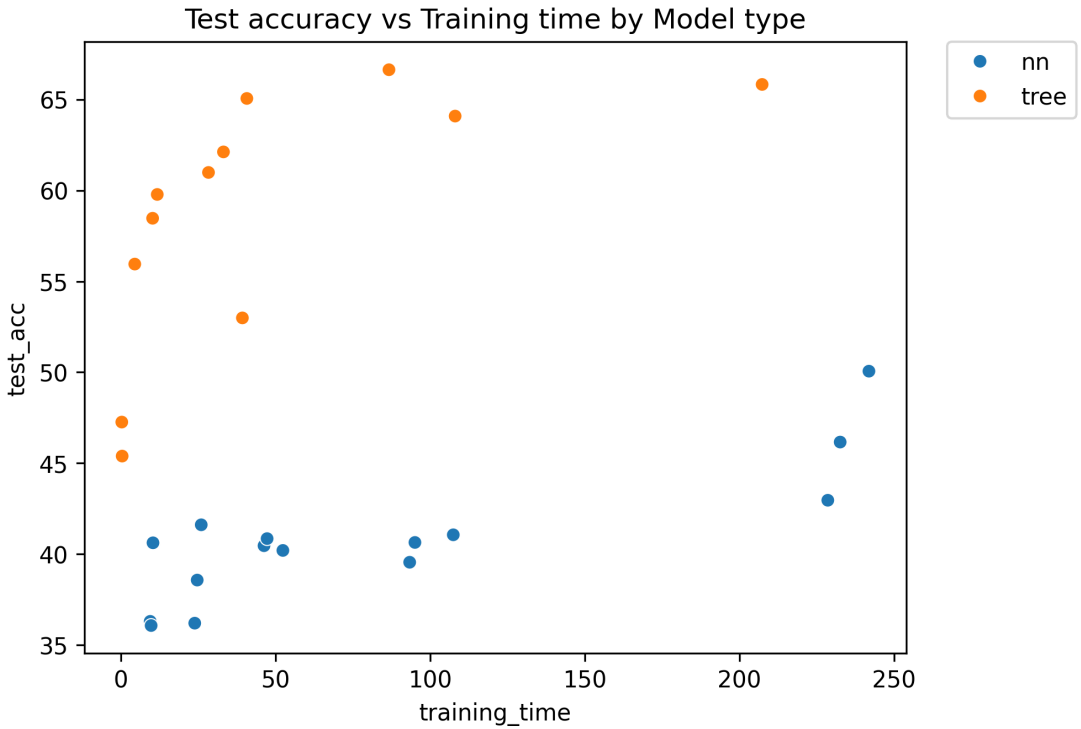


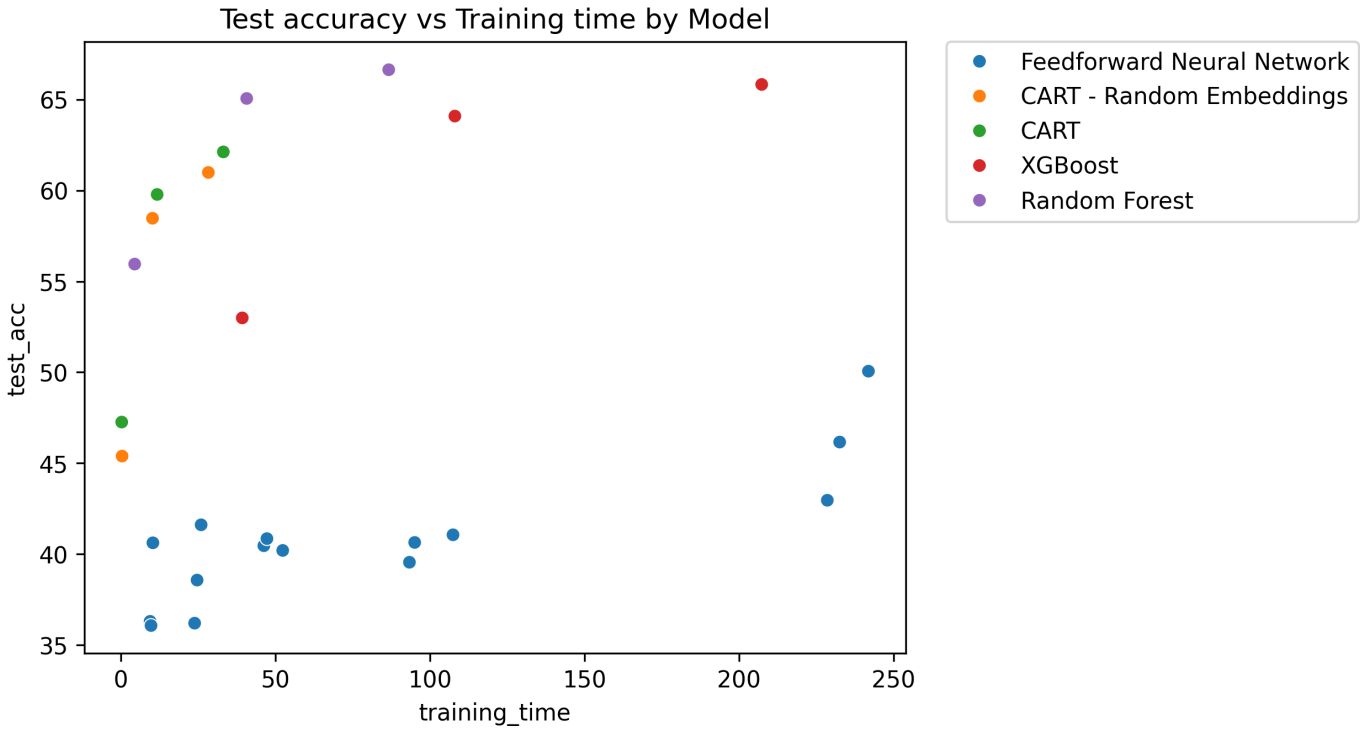
Concatenate to make 32*10 = 320 input features

MIT MANAGEMENT SLOAN SCHOOL

KEY FINDINGS

Tree-based methods can sometimes **outperform** neural networks in next character prediction for the dataset sizes we trained on.

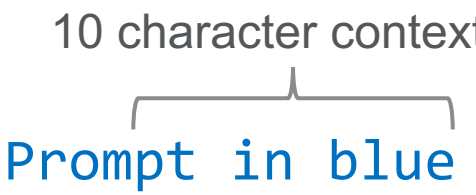




Method	Embedding size	Train Accuracy (%)	Test Accuracy (%)	Training Time (s)
FFN	32	49.98	50.06	801.5
CART – RE	8	83.56	60.99	**28.35**
CART	-	80.39	62.12	33.19
XGBoost	-	90.22	65.83	207.4
Random Forest	-	**90.36**	**66.64**	86.7

MIT MANAGEMENT SLOAN SCHOOL

EXAMPLE GENERATIONS



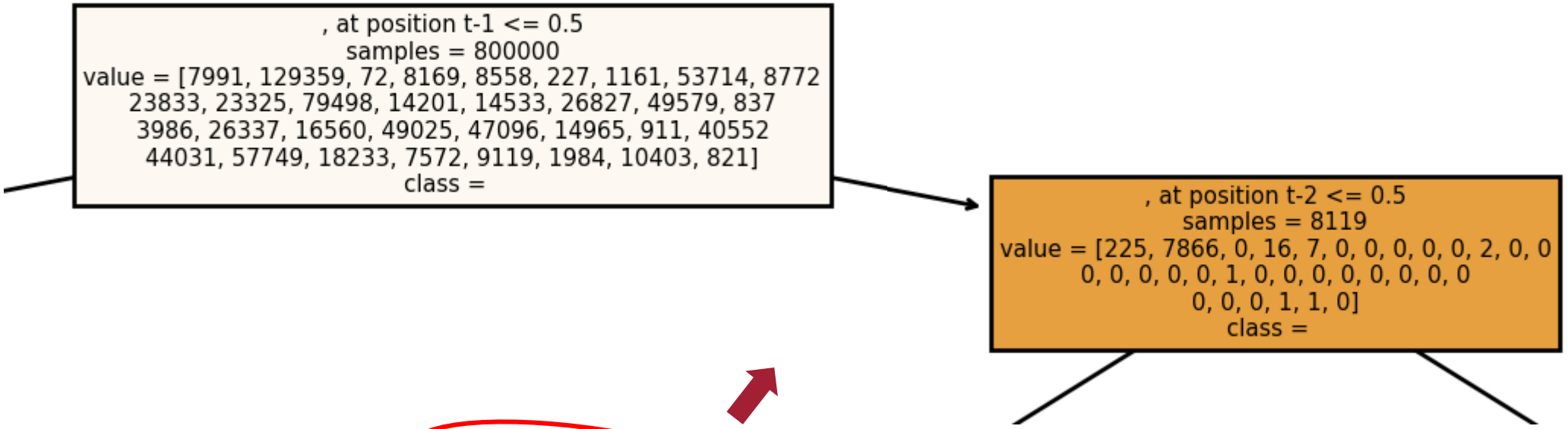
Top performing CART model (62% character accuracy):

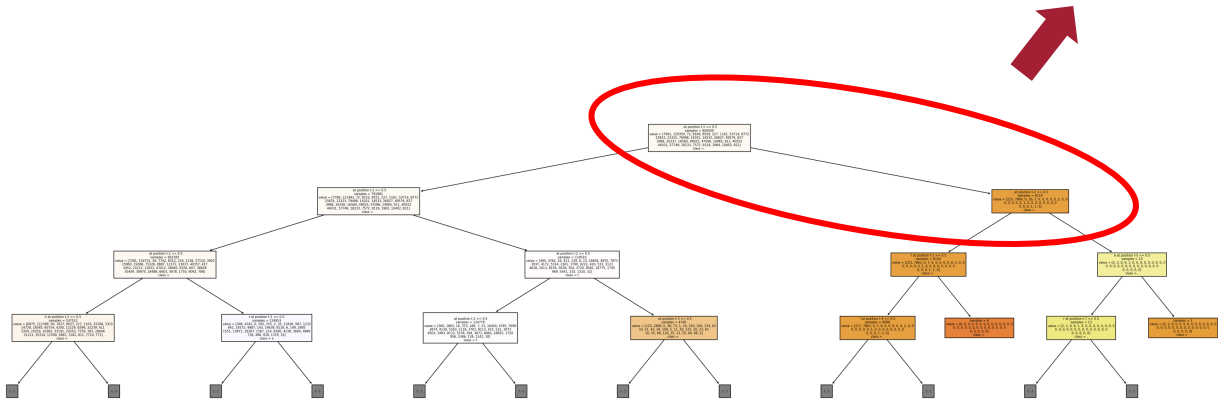
'The best place to go skiing is an oil on canvas paint constantly replenish your nots\n readers, causing wildfire more specific information\n\nile summarization, where the wonders of each character in the creation for only of learning is particularly vacation'...

Top performing Random Forest Model (67% character accuracy):

'The best place to go skiing is an exceptional church, and vast amounts of water pollution can have on clean roadmap novel context processing power. dairy, and were not seconds, saves time on the music of life a healthy lifestyle and write increases are, use the contracted by the network architectural and season with salt and pepper' ...

INTERPRETABILITY

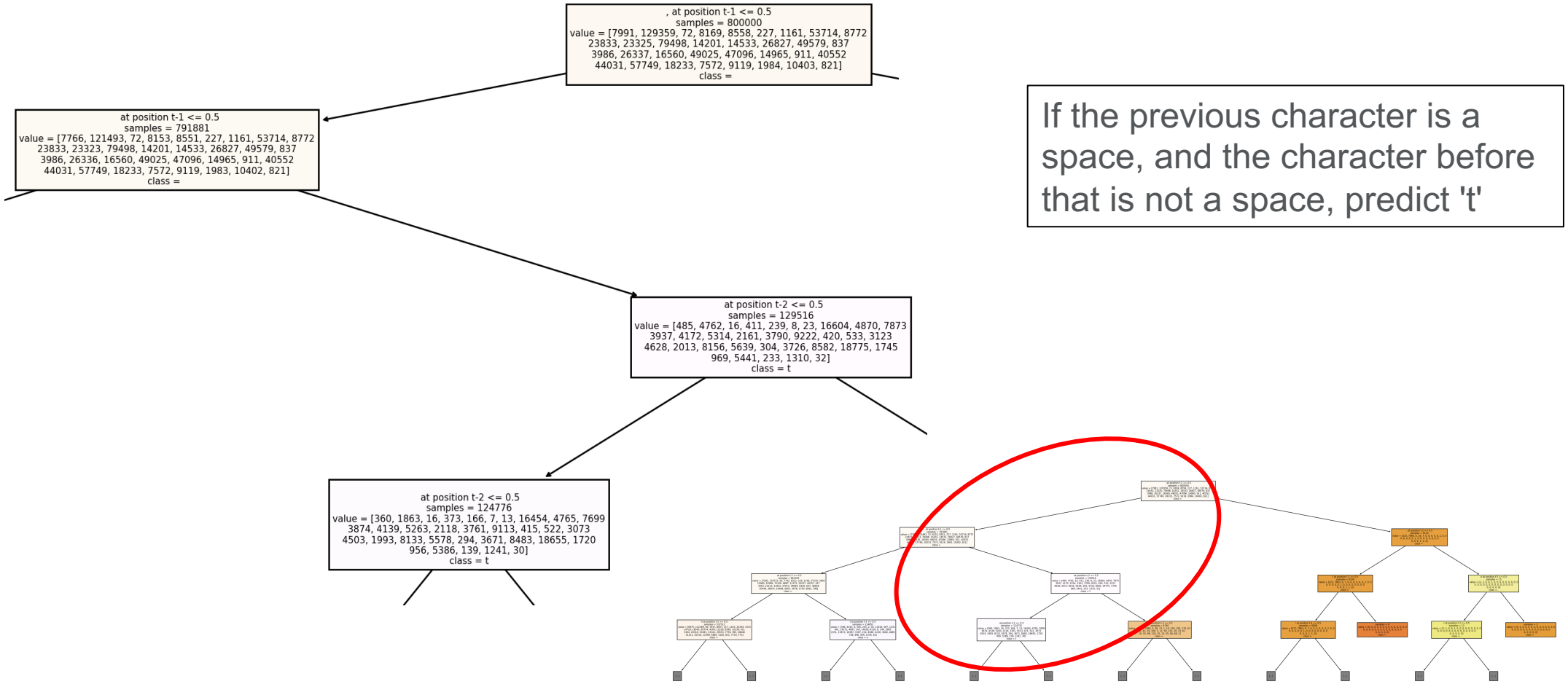




If the previous character is a comma, predict that the next character will be a space

MIT MANAGEMENT SLOAN SCHOOL

INTERPRETABILITY



If the previous character is a space, and the character before that is not a space, predict 't'

IMPACT AND CONCLUSION

- Our findings suggest that tree-based methods may be viewed as a viable alternative to neural network methods for next character prediction in small to medium sized datasets (<1M training examples).
- CART models can be trained extremely quickly, and are interpretable, allowing us to inspect the reasons why the model predicted what it did
  - This could be a step toward alternative architectures of more interpretable mid-sized language models
- Our experiments indicate that more data produces better results, and there is no reason why the process couldn't scale to larger datasets
  - However, increasing the context lengths of these models may require significant amounts of data

MIT MANAGEMENT SLOAN SCHOOL

APPENDIX: WHAT WE TRIED THAT DIDN'T WORK

- We tried several methods to try to make word embeddings using tree-based methods, none of which worked
- Idea: use a denser representation than the one-hot encodings where features of the vectors have semantic meaning
  - Train a model on initial token embeddings, then find the optimal token embeddings to maximize accuracy on that model, then retrain the model to fit those new vectors
- Methods for updating embeddings tried on CART trees:
  - "Mean embedding collection" – move the embeddings vectors closer to the value the tree output
  - "Pseudogradient descent" – make a small perturbation in every embedding element, move in the direction that minimizes the cross-entropy loss
  - "Local search with noise" – add a small random noise vector to the best-found embeddings and train a tree. Update the best tree-embeddings pair if the new tree and new embeddings give better performance

  All of the above methods were on par with random embedding vectors

MORE GENERATIONS

- **Top performing CART model (62% character accuracy):**
- 'Where is the best place to go skiing?\n\nsurface area of the triangle abc is simple language processes that release oxygen to generate longterm strategies for liquids adequality.'
- 'abcdefghijk fears and is chairs has excerpts to support their thoughts and extract if wealth, and prevent certain traits and loyalty.\n\n. promote green energy consumption of new destination or resoup platform.'
- 'The best place to go skiing is mostly driven by factoring outside and easier months.'