Fine-Tuning Pre-Trained Large Language Models to Identify Jim Crow Laws in Virginia

Tolu Odukoya

Department of Politics, University of Virginia

Introduction

CAN MACHINE LEARNING ALGORITHMS IDENTIFY JIM CROW LAWS WITHIN OTHER LAWS PASSED IN A STATE?

During the height of Jim Crow, states enacted laws to segregate races and disenfranchise African Americans and other minorities using both overt and implicit language.

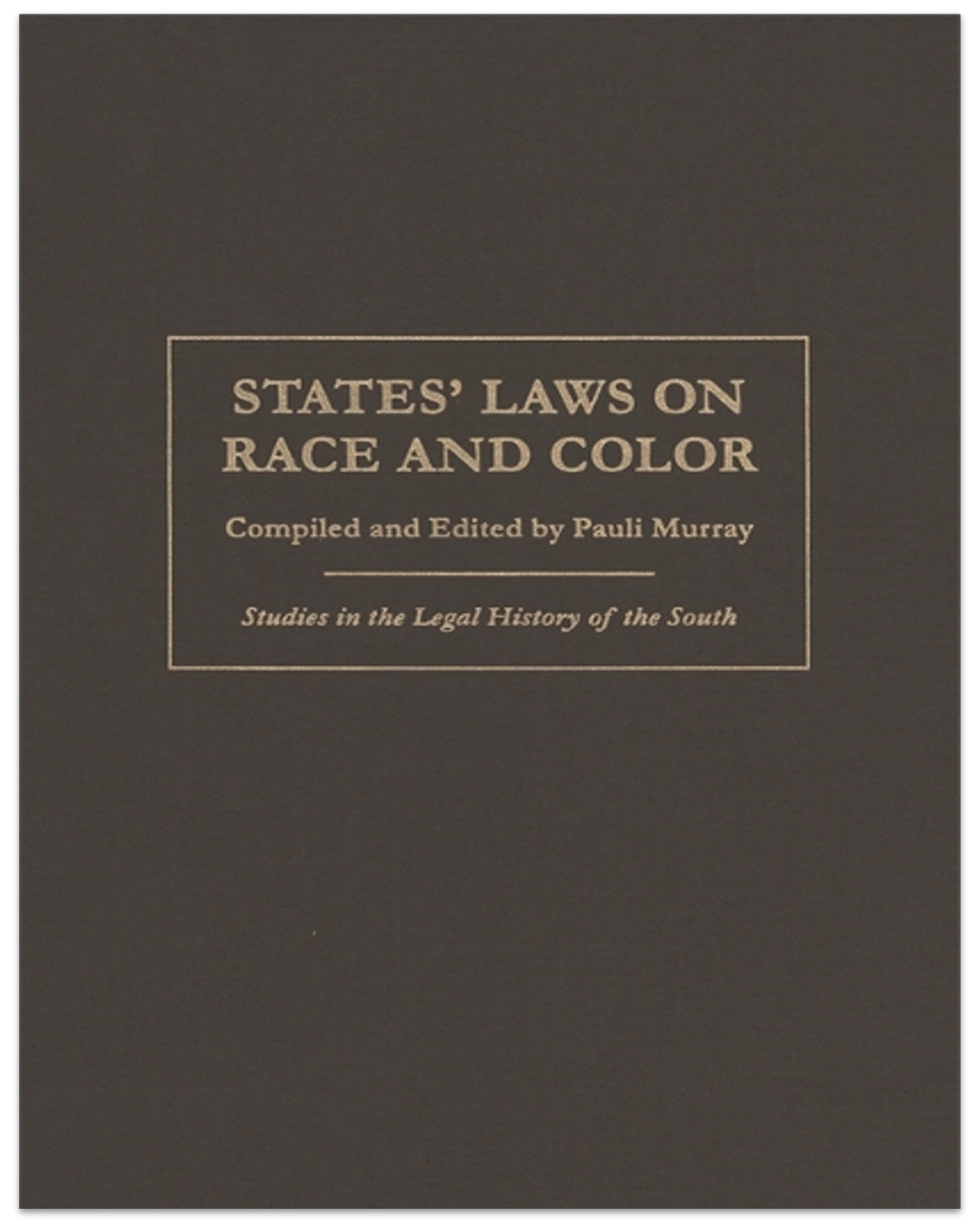
Searching legal volumes for words like "**colored**" AND "**white**" will only identify overt laws. This project uses a machine learning approach to provide a more comprehensive identification of different types of Jim Crow laws while reducing the need for close reading of legal volumes.

Objectives

- Create a corpus of laws passed between **Reconstruction and the Civil Rights Movement (1865 – 1967)**
- Create a **replicable process** for identifying Jim Crow laws
- Use **machine learning** to reduce human time and resource expenditure
- Create a **final corpus of identified Jim Crow laws**

Data

- Virginia laws **(1865 – 1967)**
- Data gathered from **HathiTrust**, **HeinOnline**, and **UVA Law Library**



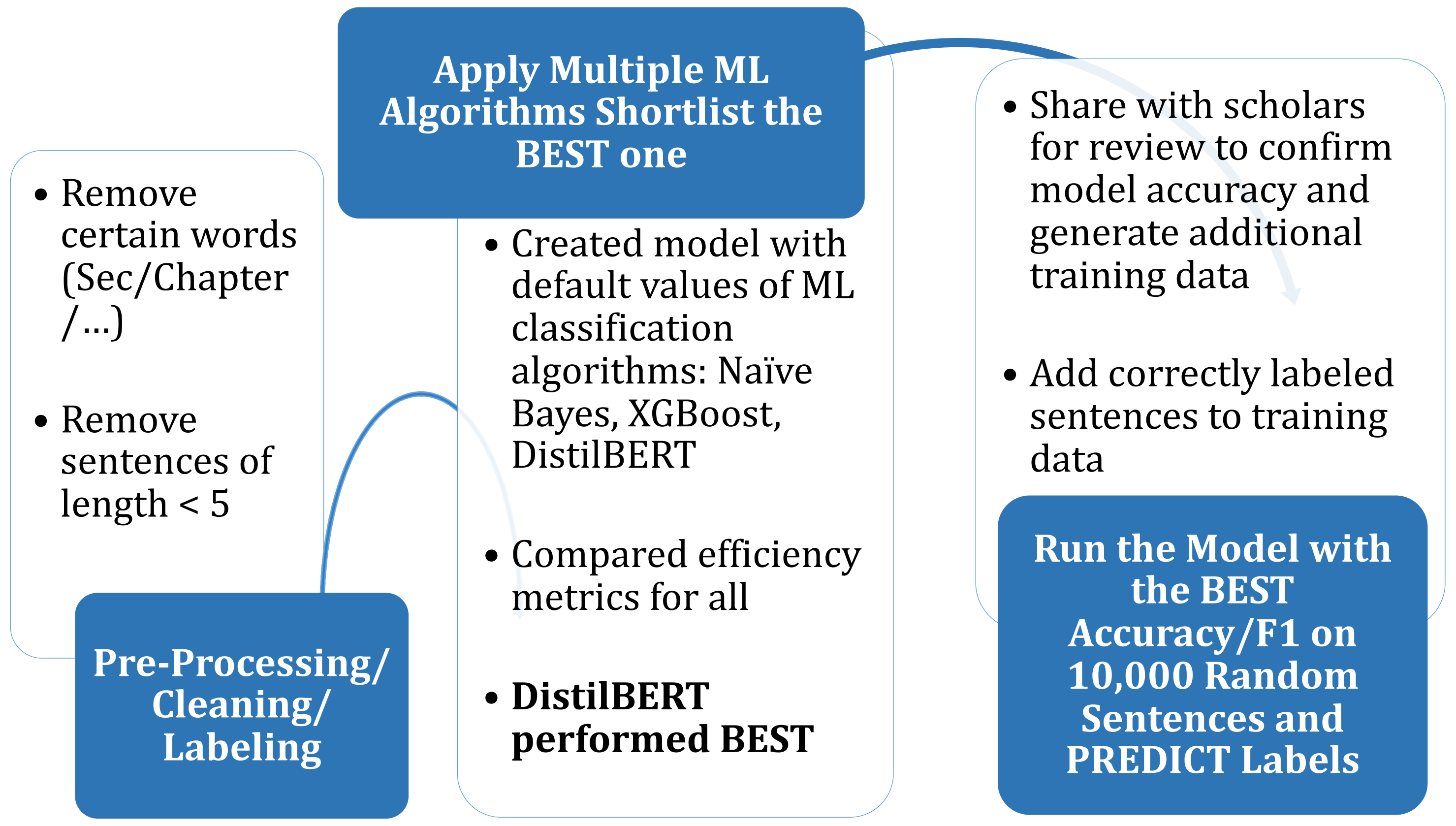
STATES' LAWS ON RACE AND COLOR

Compiled and Edited by Pauli Murray

Studies in the Legal History of the South

Methodology

- **Contour data** to remove margins and prepare for **OCR**
- Split data **into sentences,** resulting in **760,000 sentences** as the full corpus
- **Preprocess** corpus resulting in **470,000 sentences** as the final corpus
- **Python**, TensorFlow, LoRA
- Create a **training set** of **20,000** sentences labeled Jim Crow = "Yes," (1) and "No," (0)
- **Finetune DistilBERT** for classification
- GPU computing **resources on Rivanna**
- **Consultancy with UVA Research Computing** scientist Marcus Bobar



Pre-Processing/ Cleaning/ Labeling

- Remove certain words (Sec/Chapter /…)
- Remove sentences of length < 5

Apply Multiple ML Algorithms Shortlist the BEST one

- Created model with default values of ML classification algorithms: Naïve Bayes, XGBoost, DistilBERT
- Compared efficiency metrics for all
- **DistilBERT performed BEST**

Run the Model with the BEST Accuracy/F1 on 10,000 Random Sentences and PREDICT Labels

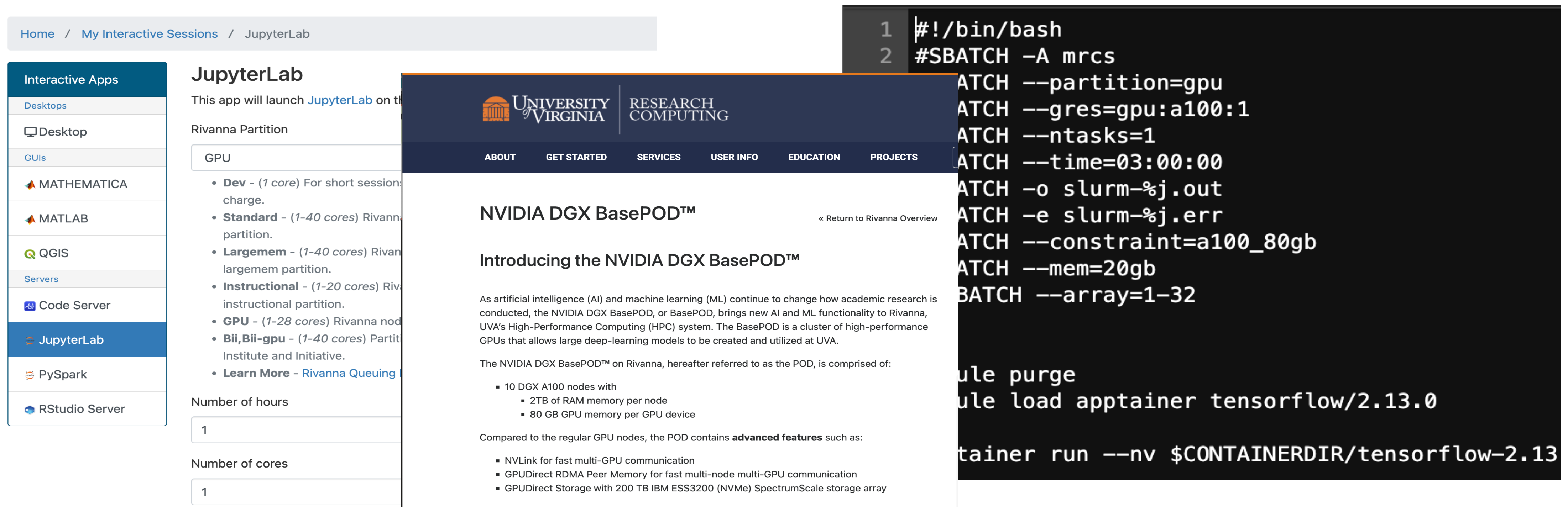
- Share with scholars for review to confirm model accuracy and generate additional training data
- Add correctly labeled sentences to training data

DistilBERT	Parameters
Batch Size	16
Epoch	6
Learning Rate	2e-5
Warmup Steps	5
Accuracy and F1	**0.99**

Final Model Parameters and Score

Research Computing Resources

- Over **200 hours of Research Computing**



Home / My Interactive Sessions / JupyterLab

JupyterLab

This app will launch JupyterLab on t

Rivanna Partition

GPU

- **Dev** - (*1 core*) For short session charge.
- **Standard** - (*1-40 cores*) Rivann partition.
- **Largemem** - (*1-40 cores*) Rivan largemem partition.
- **Instructional** - (*1-20 cores*) Riv instructional partition.
- **GPU** - (*1-28 cores*) Rivanna nod
- **Bii,Bii-gpu** - (*1-40 cores*) Partit Institute and Initiative.
- **Learn More** - Rivanna Queuing

Number of hours

1

Number of cores

1

NVIDIA DGX BasePOD™

« Return to Rivanna Overview

Introducing the NVIDIA DGX BasePOD™

As artificial intelligence (AI) and machine learning (ML) continue to change how academic research is conducted, the NVIDIA DGX BasePOD, or BasePOD, brings new AI and ML functionality to Rivanna, UVA's High-Performance Computing (HPC) system. The BasePOD is a cluster of high-performance GPUs that allows large deep-learning models to be created and utilized at UVA.

The NVIDIA DGX BasePOD™ on Rivanna, hereafter referred to as the POD, is comprised of:

- 10 DGX A100 nodes with
  - 2TB of RAM memory per node
  - 80 GB GPU memory per GPU device

Compared to the regular GPU nodes, the POD contains **advanced features** such as:

- NVLink for fast multi-GPU communication
- GPUDirect RDMA Peer Memory for fast multi-node multi-GPU communication
- GPUDirect Storage with 200 TB IBM ESS3200 (NVMe) SpectrumScale storage array

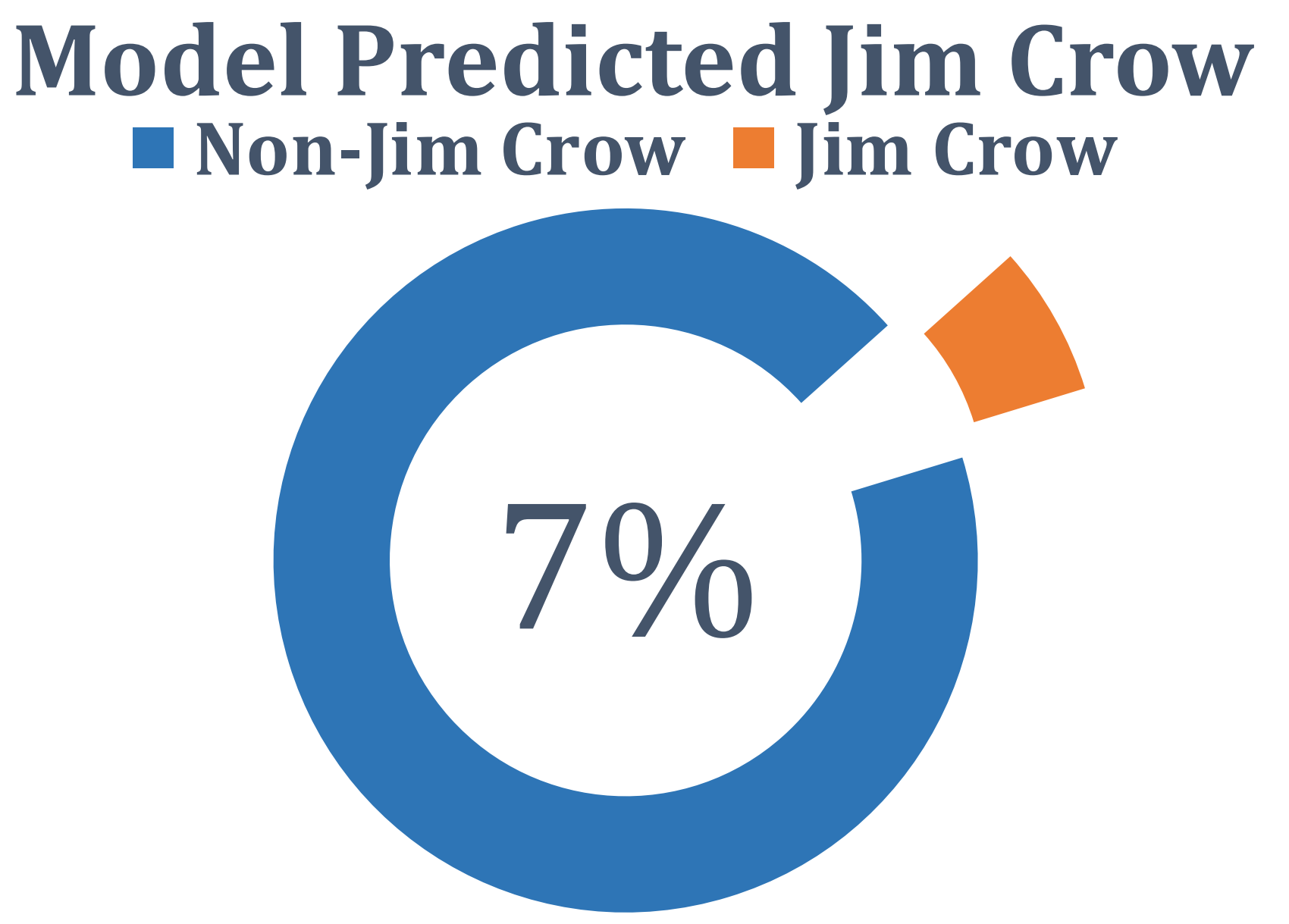
```
#!/bin/bash
#SBATCH -A mrcs
ATCH --partition=gpu
ATCH --gres=gpu:a100:1
ATCH --ntasks=1
ATCH --time=03:00:00
ATCH -o slurm-%j.out
ATCH -e slurm-%j.err
ATCH --constraint=a100_80gb
ATCH --mem=20gb
BATCH --array=1-32


ule purge
ule load apptainer tensorflow/2.13.0

tainer run --nv $CONTAINERDIR/tensorflow-2.13
```

Rivanna On Demand **Rivanna GPU POD** **SLURM Script**

Research Computing Resources

- **On-Demand GPU** sessions to train and fine-tune the model
- **GPU NVIDIA** pod provided access to **80GB memory** Rivanna devices for **training and inference**
- Used **GPU array and SLURM script for** model **prediction on 470,000** sentences
- SLURM GPU array **reduced inference timing from Over 30 hours to 2 hours**

Results



- Identified over **20,000 Jim Crow laws**

Research Significance

- Created the **first finetuned Large Language Model for Jim Crow classification**
- Provides a **reproducible process** for identifying **Jim Crow laws for other states**
- Created the **first machine-learning classified corpus** of Jim Crow Laws **for Virginia**

Acknowledgments

- **Mellon Foundation**
- **University of North Carolina On The Books Team**
- **UVA Law Library and Main Library On The Books Team**
- **Marcus Bobar**
- **Research Computing**