

# 2024 REU AI FOR JUSTICE – FINAL REPORT

SHREYA BALAJI, DAKOTA LIN, ANSHUMAN SINGH, KYLE TORRES

PI: PROF. DEANNA NEEDELL

MENTOR: DR. MINXIN ZHANG

CONSULTANTS: MIKE SEMANCHIK (TIC), MARISSA BLUESTINE (QUATTRONE)

**ABSTRACT.** The United States criminal justice system consistently has high incarceration rates and systemic racial disparities, which leads to large numbers of wrongful convictions, especially in minority groups. These issues highlight the urgent need for tools that can improve the fairness and accuracy in our legal processes. Our project aims to develop a reliable and transparent recommendation tool using Artificial Intelligence (AI) technology to assist innocence organizations by summarizing data, conducting complex searches, and identifying key features in appeal investigations, which will address the need for effective tools that prevent wrongful convictions. The dataset for this research project was compiled with guidance from the Innocence Center - exonerated murder cases were sourced from The National Registry of Exonerations, focusing on cases within the last ten years to standardize the data; non-exonerated murder cases were sourced by maintaining geographic diversity and matching the states represented in the exonerated cases. In total, we have 140 cases from 39 states. We implemented several existing algorithms, including Semi-NMF, Convex NMF, and SSNMF, and further extended them to develop a new method, Kernel SSNMF. Each algorithm was selected and modified to handle the mixed-sign and nonnegative data characteristics of our dataset, and we provided convergence proofs for these algorithms to ensure their reliability and applied them to classify the data, reconstruct feature matrices, and evaluate their performance through SVM classification. Our study found that Kernel SSNMF outperformed both Convex NMF and SSNMF in classification tasks on gene expression data and LLM embeddings. Particularly with polynomial kernels, Kernel SSNMF consistently achieved higher accuracy across different numbers of topics. Overall, the results of our study highlight gaps and opportunities for enhancing fairness and accuracy in the criminal justice system. By addressing the limitations of our dataset and refining our models, we can contribute to a better legal framework and create reforms in the justice system.

## 1. INTRODUCTION

### 1.1. Background.

**1.1.1. Injustice in Our Criminal Justice System.** The United States criminal justice system faces several challenges, including high incarceration rates that disproportionately affect minority communities. Marc Mauer, the assistant director of the Sentencing Project, highlights that 49% of prison inmates are African American, despite African Americans comprising only 13% of the overall population [1]. The “Prisoners in 2022 - Statistical Tables” report [2] indicates that of the over 1.2 million incarcerated individuals in 2022, 32% were Black, 23% were Hispanic, 2% were American Indian or Alaska Native, and 1% were Asian, Native Hawaiian, or Other Pacific Islander, whereas 31% were White. Estimates suggest that 5-10% of this population, or up to 120,000 individuals, are innocent. Further emphasizing the issue, a study by the University of Michigan Law School [3] found that if all defendants sentenced to death in the United States remained under the death sentence indefinitely, at least 4.1% would be exonerated. Additionally, the Criminal Justice Experience Survey (CJES) reported that wrongful convictions occur in 6% of criminal convictions leading to imprisonment in an intake population of state prisoners. These statistics show that there is an urgent need for reform in our criminal justice system to reduce wrongful convictions and ensure fairness in the criminal justice system

---

Date: August 19, 2025.

1.1.2. *The Innocence Center.* The Innocence Center [4] is a non-profit organization dedicated to exonerating individuals who have been wrongfully convicted of crimes. Although The Innocence Center was founded in 2023, the board and staff members have decades of innocence work experience from the California Innocence Project. This group has collectively exonerated over 40 individuals after extensive case reviews on vast amounts of data including inmate letters and case files. They work to rectify criminal injustice and improve the criminal justice system by advocating for changes in the policies and practices that lead to wrongful convictions. Moreover, the Innocence Center supports exonerated individuals as they adjust to life post-release, helping them reintegrate into society. They also educate the public about the causes and consequences of these wrongful convictions through partnerships with educational institutions, striving to create a more just legal system.

1.1.3. *The National Registry of Exonerations.* The National Registry of Exonerations [5], founded in 2012, is a detailed database that collects, analyzes, and disseminates information about individuals who have been wrongfully convicted of crimes and later exonerated. Their goal is to offer accurate, objective, transparent, and accessible information about these cases to the public. By relying entirely on publicly available data, they aim to educate policymakers and the general public about both the causes and consequences of wrongful convictions. This effort raises awareness about systemic issues within the criminal justice system, in hopes of preventing future wrongful convictions and advocating for extensive reforms. The Registry documents a wide range of information for each case, including the nature of the crime committed, the circumstances that led to the wrongful conviction, the legal process that led to the exoneration, and other contributing factors such as mistaken witness identification, false confession, perjury or false accusation, and inadequate legal defense. Each of the cases provides a thorough overview of the incident, with a specific focus on the errors leading to conviction and the steps resulting in the exoneration. In addition to maintaining this comprehensive database, the Registry produces annual reports which summarize trends and patterns in wrongful convictions over the year, highlighting areas that need to be reformed. Often, these reports delve into specific issues such as racial disparities in these wrongful convictions, the impact of forensic science errors, prosecutorial misconduct, and ineffective legal representation. These trends allow for the Registry to identify common factors that contribute to wrongful convictions, providing insights for reform efforts. The Registry also collaborates with innocence organizations, legal clinics, and academic institutions to support any ongoing efforts to seek justice for those who have been wrongfully accused. This joint effort helps facilitate the sharing of resources, expertise, and useful strategies to overturn wrongful convictions by implementing preventative measures. This also allows for the opportunity for the Registry to engage in more public education initiatives, by presenting at seminars and workshops to inform the public about the reality of wrongful convictions. Overall, through its data collection and analysis, reports, collaborative efforts, and public presence, the National Registry of Exonerations not only brings attention to individual cases of injustice but also contributes to a broader understanding of the systemic failures that lead to wrongful convictions. It plays a crucial role in efforts to improve the fairness and accuracy of the criminal justice system, advocating for changes to protect innocent people from being wrongfully convicted.

1.2. **The Purpose in Our Work.** With recent advancements in Artificial Intelligence (AI) and specifically with large language models (LLMs), we are able to use predictive tools to assist innocence organizations in their decision-making process. Our research goal is to develop a reliable and transparent recommendation tool using AI which will summarize data, conduct complex searches, and identify key features related to appeal investigations. Additionally, we will test our developed models against historical decisions, to ensure reliability in our models. In collaborating with the UCLA mathematics department, the Innocence Center, and the National Registry of Exonerations, our goal is to enhance the use of AI and Machine Learning (ML) technologies within the criminal justice system. We ultimately want to ensure that AI technologies used are fair, reliable, and transparent in order to mitigate any potential biases or discrimination that's inherent in these systems due to historical data. This project is about so much more

than technological advancement. It's also about securing justice for all and protecting humanity. Every person who is convicted of a crime deserves to go through a legal process which is fair and unbiased. Through our research, we hope to create a system that can make a difference in the lives of those who are wrongfully affected by the criminal justice system.

**1.3. Contributions.** The contributions of this paper are as follows:

- (1) We provide a method for converting open-source legal documents into text embeddings for classification tasks.
- (2) We introduce a new algorithm, Kernel Semi-Supervised Non-Negative Matrix Factorization (Kernel SSNMF), to analyze and classify data with many features, such as UCI Genomic Data or legal text embeddings.
- (3) We prove the convergence of our proposed algorithm.
- (4) We propose an alternative approach for performing classification tasks on Convex NMF and Kernel SSNMF by reconstructing the feature matrix.
- (5) We perform classification tasks on publicly available datasets with numerous features, including exonerated and non-exonerated documents, demonstrating that our algorithm outperforms SSNMF and Convex NMF.

**1.4. Organization.** In Section 2, we discuss the data preparation and highlight the features of our data such as demographic information and average number of years incarcerated. In Section 3, we introduce pre-existing algorithms and introduce our algorithm: Kernel SSNMF. We prove convergence and develop the theory behind how we performed classification tasks for Convex NMF and Kernel SSNMF. In Section 4, we discuss our experimental setup and evaluate the performance of our algorithm on open source data (eg. UCI Genomic Data, GEMLeR Colon Kidney Data) as well as our embeddings data set to classify wrongful incarceration. In Section 5, we discuss our results and potential future works for understanding wrongful incarcerations.

## 2. DATA PREPARATION

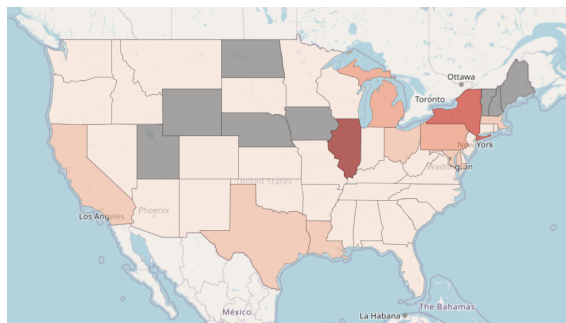
**2.1. Collecting the Data.** The Innocence Center provided our team with guidance to create the dataset used for this research project. Our goal was to compile a total of 100-200 documents (court opinions) of murder cases, with 50-100 documents representing exonerated cases and 50-100 documents representing non-exonerated cases. For the exonerated cases, we sourced our data points from The National Registry of Exonerations and then located the corresponding documents on Casetext or Westlaw. We filtered our data to include only murder cases where the date of exoneration occurred within the last ten years. This criterion served to standardize our data, as cases might differ depending on the type of crime committed and the particular laws in place at the time of the exoneration. We also ensured representation across the United States by attempting to include at least one case from each state and by excluding federal Supreme Court cases.

For the exonerated cases, we randomly selected one case from each state and then proceeded to randomly select from the remaining data to reach a total of 100 data points. We then searched for the corresponding court opinions on Casetext; if we could not find the case on Casetext, we checked Westlaw. However, if we could not locate the court opinion on either Casetext or Westlaw, we eliminated the case altogether. It is important to note that the appeals process consists of several stages, and each stage might correspond to a different court opinion. As such, we had to find a systematic way to select one opinion for those cases that had multiple court opinions. To do so, we chose the court opinion that was written closest to the date of conviction, as this opinion was the furthest removed from the date of eventual exoneration. This time difference is important because it minimizes the potential for bias in the court opinion that could be influenced by new exoneration evidence introduced at a later stage in trial.

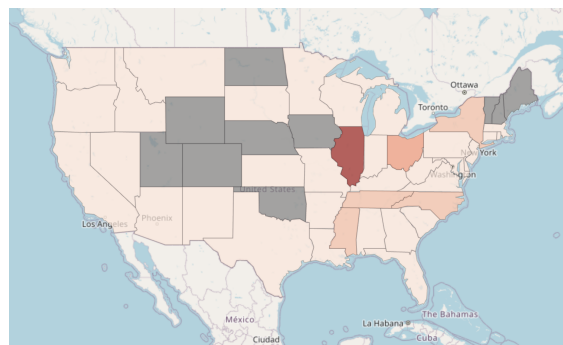
We repeated this process until we collected 70 exonerated cases, a sufficient number of data points that fit inside our desired range. For the non-exonerated cases, we aimed to preserve the geographic distribution of data points in the exonerated data, so we collected case data from the same states as those of the exonerated cases. In total, our data includes 140 cases (70 exonerated cases and 70 non-exonerated cases). However, because we were unable to find exonerated cases from certain states, our data is comprised of 39 states, excluding Colorado, Hawaii, Iowa, Maine, Nebraska, New Hampshire, North Dakota, Oklahoma, Utah, Vermont, and Wyoming.

**2.2. Exploratory Data Analysis.** In the present section, we focus on analyzing the given data from the National Registry of Exonerations as well as our sampled data. The original data from the National Registry of Exonerations consists of 3549 cases that represent the number of known exonerations in the United States since 1989. We filtered this data to only include murder cases that resulted in exonerations in the past 10 years, narrowing down the dataset to 629 cases. This abridged dataset served as the basis from which we sampled our data.

As mentioned in the Data Preparation section, we sampled 70 exoneration cases from the original dataset with the goal of incorporating cases from as many states as possible. To compare the geographical distribution of our sample data with that of the original data, we created two heat maps representing the states that were included in each of the datasets. The two maps are pictured below:



((a)) Heat map of original data depicting number of exoneration cases by state

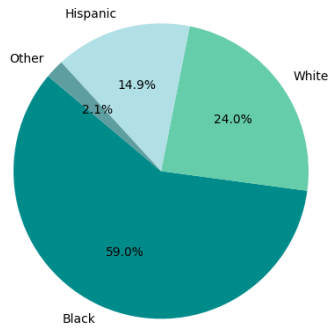


((b)) Heat map of sample data depicting number of exoneration cases by state

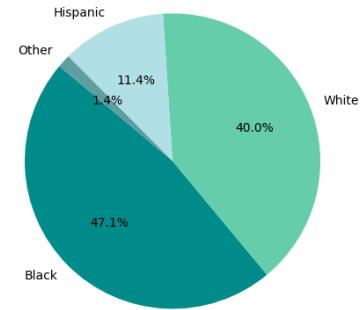
**Figure 1** Heat maps of exonerations by state in original dataset compared to sample dataset

We see that the original dataset is missing nine states (Hawaii, Iowa, Maine, Nebraska, New Hampshire, North Dakota, Utah, Vermont, and Wyoming); whereas, the sampled dataset is missing 11 states (Colorado and Oklahoma along with the nine other aforementioned states). Further, we note that the majority of the exoneration cases in the original dataset occurred in the Northeastern and Western parts of the country. While this pattern can still be seen in the sampled data, the extent of the geographical disparity is diminished. It is also interesting to note that Illinois had the most exoneration cases in each dataset (100 in the original dataset and 8 in the sampled dataset). This alignment makes sense because Illinois had 26 more exoneration cases than the next closest state (New York) in the original dataset, a sizeable margin.

To get a sense of how the demographic composition of the sample data compared to the original data, we created plots to visualize the racial distribution and age distribution of exonerees. It is important to note that we could only analyze data for exoneration cases because the data for non-exoneration cases did not include the relevant demographic information. Pie charts visualizing the racial distribution of exonerees are shown below:



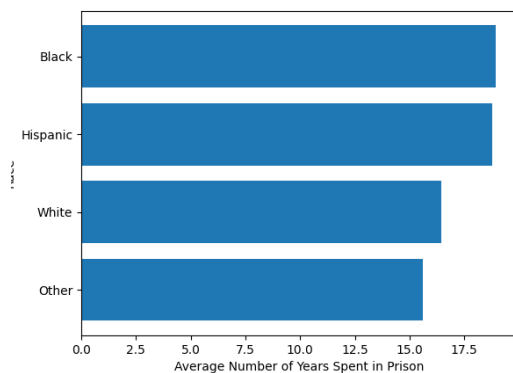
((a)) Racial distribution of exonerees in original data



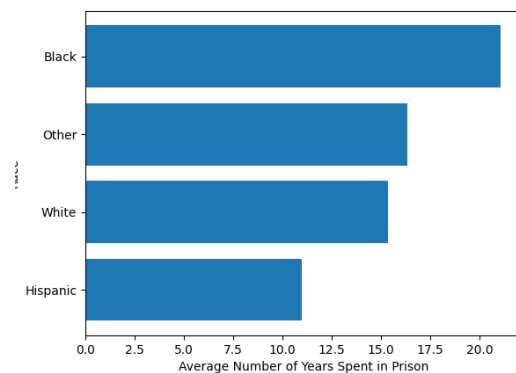
((b)) Racial distribution of exonerees in sampled data

**Figure 2** Racial distribution of exonerees in original dataset compared to sample dataset

We observe that the order of the plots is the same: most exonerees were Black, then White, then Hispanic, then "Other." This grouping demonstrates a glaring issue in America's legal system, where Black Americans represent more than half of the exoneration cases and consequent wrongful convictions. It is worth noting that there is a noticeable difference in the proportion of certain racial groups in the two datasets, particularly between Black and White exonerees. In the original dataset, Black exonerees comprised nearly 60% of the data; whereas, they made up 47% of the sample data. On the other hand, White exonerees made up 24% of the original data and 40% of the sample data. This disparity between racial groups in the different datasets can be attributed to natural variance in our sampling method. We were curious to see if racial bias was perpetuated in the time spent incarcerated among exonerees, so we created plots measuring incarceration length by race in both the original and sample dataset:



((a)) Average number of years incarcerated by race in original data



((b)) Average number of years incarcerated by race in sampled data

**Figure 3** Average time incarcerated by race in original dataset compared to sample dataset

From the above two plots, we see that the racial order changes. In both plots, Black and White exonerees face the longest and third-longest incarceration times, respectively. However, while Hispanic exonerees endure the second longest incarceration length in the original data, they have the shortest incarceration length in the sampled data (the racial group 'Other' faces the second longest incarceration

length in the sampled data). Further examination reveals that there is only one exoneree belonging to the 'Other' racial group, which skewed the order. Like the racial distribution plots above, this difference can be attributed to natural variance, but it is still important to take note.

### 3. ALGORITHMS

#### 3.1. Existing Algorithms Overview.

3.1.1. *Semi NMF*. Semi NMF is a variation of NMF where the basis matrix can have both positive and negative values while keeping the coefficient matrix non-negative. The data vectors are represented as column vectors of the input matrix. This flexibility in the algorithm allows Semi NMF to better capture the structure of complex mixed-sign data as the basis vectors are not restricted to non-negative values. This algorithm also enhances the interpretability of our data by having sparse representations with the non-negative coefficient matrix highlighting, which can help us identify the most significant features within the data.

The algorithm we used for semi NMF [6] is based on iterative updating methods, which alternatively updates the values for the basis matrix  $\mathbf{F}$  and the coefficient matrix  $\mathbf{G}$  until convergence. By repeatedly applying these update rules, the quality of the approximation improves, guaranteeing convergence to a locally optimal matrix factorization. This process ensures we reach a solution that closely approximates the original data matrix, meaning the factorization converges to a locally optimal solution.

First, we initialized  $\mathbf{G}$  and did a K-means clustering, which gives cluster indicators  $\mathbf{G} : \mathbf{G}_{ik} = 1$ , if  $x_i$  belongs to cluster  $k$  (or otherwise  $\mathbf{G}_{ik} = 0$ ). Then, we added a small constant (the value 0.2) to all elements of  $\mathbf{G}$ . Next, we updated  $\mathbf{F}$  while also fixing  $\mathbf{G}$  using the following rule:

$$\mathbf{F} = \mathbf{X}\mathbf{G}(\mathbf{G}^T\mathbf{G})^{-1} \quad (1)$$

Here,  $\mathbf{G}^T\mathbf{G}$  is a  $k \times k$  positive semidefinite matrix. Next, we updated  $\mathbf{G}$  while also fixing  $\mathbf{F}$  using the following rule:

$$\mathbf{G}_{ik} \leftarrow \mathbf{G}_{ik} \sqrt{\frac{(\mathbf{X}^T\mathbf{F})_{ik}^+ + [\mathbf{G}(\mathbf{F}^T\mathbf{F})^-]_{ik}}{(\mathbf{X}^T\mathbf{F})_{ik}^- + [\mathbf{G}(\mathbf{F}^T\mathbf{F})^+]_{ik}}} \quad (2)$$

We separate the positive and negative parts of a matrix  $A$  as follows:

$$\mathbf{A}_{ik}^+ = (|\mathbf{A}_{ik}| + \mathbf{A}_{ik})/2, \mathbf{A}_{ik}^- = (|\mathbf{A}_{ik}| - \mathbf{A}_{ik})/2. \quad (3)$$

Algorithm 1 is our pseudocode for Semi NMF.

---

**Algorithm 1** Semi Nonnegative Matrix Factorization (Semi-NMF)

---

```
1: procedure SEMINMF( $\mathbf{X}, k, N, \text{tol}$ )
2:   Input:
   •  $\mathbf{X}$ : Data matrix of dimensions  $p \times n$ 
   •  $k$ : Number of components
   •  $N$ : Maximum number of iterations
   •  $\text{tol}$ : Tolerance for convergence
3:   Initialization:
   •  $\mathbf{G}, \mathbf{F} \leftarrow \text{K-Means Clustering}(\mathbf{X}, k)$ 
   •  $\mathbf{X} \leftarrow \mathbf{X}^T$ 
4:   for  $i = 1, 2, 3, \dots, N$  do
5:     Update F while fixing G:
6:      $\mathbf{F} \leftarrow \mathbf{X}\mathbf{G}(\mathbf{G}^T\mathbf{G})^{-1}$ 
7:     Update G while fixing F:
8:      $\mathbf{G} \leftarrow \mathbf{G} \cdot \sqrt{\frac{(\mathbf{X}^T\mathbf{F})^+ + \mathbf{G}(\mathbf{F}^T\mathbf{F})^-}{(\mathbf{X}^T\mathbf{F})^- + \mathbf{G}(\mathbf{F}^T\mathbf{F})^+}}$ 
9:     Calculate norm  $\leftarrow \|\mathbf{X} - \mathbf{F}\mathbf{G}^T\|_F$ 
10:    if norm  $\leq \text{tol}$  then
11:      break
12:    end if
13:  end for
14:  Output:  $\mathbf{F}, \mathbf{G}$ 
15: end procedure
```

---

3.1.2. *Convex NMF*. Convex NMF is a variation of NMF where the basis vectors are made up of combinations of the input data columns, making them easier to interpret as weighted sums of data points, similar to centroids in clustering. This approach makes sure that the basis vectors are within the column space of the input data, which enhances the interpretability of the data. Additionally, Convex NMF can be applied to nonnegative data and mixed-sign data (where datasets have both positive and negative values), and tends to produce sparse factors. This sparsity highlights the most significant features in the data, which is useful in identifying patterns in a given dataset.

The algorithm we used for Convex NMF [6] is based on iterative updating methods, which alternatively updates the values for the basis matrix  $F$  and the coefficient matrix  $G$  until convergence. In each of the iterations, the values for  $F$  (represented by  $W$ ) and  $G$  are updated by multiplying their current values by a specific factor. By repeatedly applying these multiplicative update rules, the quality of the approximation is improved which guarantees convergence to a locally optimal matrix factorization. This process ensured that we reached a solution that could closely approximate the original data matrix, meaning that the factorization converged to a locally optimal solution. First, we applied Semi-NMF. Then we initialized  $G$  and  $W$  using K-means clustering because it provides a way to estimate initial cluster centroids which can be a good approximation for the basis vectors, which would ultimately help us achieve a faster and more stable convergence. Here,  $\mathbf{G}$  is known and we obtained a least square solution for  $\mathbf{W}$  by solving  $\mathbf{X} = \mathbf{X}\mathbf{W}\mathbf{G}^T$ , and therefore  $\mathbf{W} = \mathbf{G}(\mathbf{G}^T\mathbf{G})^{-1}$ .

Then, we updated  $G$  while also fixing  $W$  using the following rule:

$$\mathbf{G}_{ik} \leftarrow \mathbf{G}_{ik} \sqrt{\frac{[(\mathbf{X}^T\mathbf{X})^+ + \mathbf{W}]_{ik} + [\mathbf{G}\mathbf{W}^T(\mathbf{X}^T\mathbf{X})^-]_{ik}}{[(\mathbf{X}^T\mathbf{X})^-]_{ik} + [\mathbf{G}\mathbf{W}^T(\mathbf{X}^T\mathbf{X})^+]_{ik}}} \quad (4)$$

7

Next, we updated  $W$  while also fixing  $G$  using the following rule:

$$\mathbf{W}_{ik} \leftarrow \mathbf{W}_{ik} \sqrt{\frac{[(\mathbf{X}^T \mathbf{X})^+ \mathbf{G}]_{ik} + [(\mathbf{X}^T \mathbf{X})^- \mathbf{W} \mathbf{G}^T \mathbf{G}]_{ik}}{[(\mathbf{X}^T \mathbf{X})^- \mathbf{G}]_{ik} + [(\mathbf{X}^T \mathbf{X})^+ \mathbf{W} \mathbf{G}^T \mathbf{G}]_{ik}}} \quad (5)$$

These update steps are repeated until convergence, which is until the norm of the difference between the original data matrix  $X$  and its approximation  $X = XWG^T$  fall below a specified tolerance level. This process ensured that we reached a solution that could closely approximate the original data matrix, meaning that the factorization converged to a locally optimal solution. The following is our pseudocode for this Convex NMF algorithm:

---

**Algorithm 2** Convex Nonnegative Matrix Factorization (Convex NMF)

---

```

1: procedure CONVEXNMF( $\mathbf{X}, \mathbf{S}, N, \text{tol}$ )
2:   Input:
      •  $\mathbf{X}$ : Data matrix of dimensions  $p \times n$ 
      •  $\mathbf{S}$ : Topic matrix
      •  $N$ : Maximum number of iterations
      •  $\text{tol}$ : Tolerance for convergence
3:   Initialization:
      • Initialize  $\mathbf{G}$  using semi-NMF
      •  $\mathbf{W} \leftarrow \mathbf{G}(\mathbf{G}^T \mathbf{G})^{-1}$ 
4:   Iterative Updates:
      for  $i = 1, 2, 3, \dots, N$  do
5:     Update  $\mathbf{G}$  while fixing  $\mathbf{W}$ :
6:        $\mathbf{G}_{ik} \leftarrow \mathbf{G}_{ik} \sqrt{\frac{[(\mathbf{X}^T \mathbf{X})^+ \mathbf{W}]_{ik} + [\mathbf{G} \mathbf{W}^T (\mathbf{X}^T \mathbf{X})^- \mathbf{W}]_{ik}}{[(\mathbf{X}^T \mathbf{X})^- \mathbf{W}]_{ik} + [\mathbf{G} \mathbf{W}^T (\mathbf{X}^T \mathbf{X})^+ \mathbf{W}]_{ik}}}$ 
7:       if  $\|\mathbf{X} - \mathbf{XW} \mathbf{G}^T\| \leq \text{tol}$  then
8:         break
9:       end if
10:      Update  $\mathbf{W}$  while fixing  $\mathbf{G}$ :
11:         $\mathbf{W}_{ik} \leftarrow \mathbf{W}_{ik} \sqrt{\frac{[(\mathbf{X}^T \mathbf{X})^+ \mathbf{G}]_{ik} + [(\mathbf{X}^T \mathbf{X})^- \mathbf{W} \mathbf{G}^T \mathbf{G}]_{ik}}{[(\mathbf{X}^T \mathbf{X})^- \mathbf{G}]_{ik} + [(\mathbf{X}^T \mathbf{X})^+ \mathbf{W} \mathbf{G}^T \mathbf{G}]_{ik}}}$ 
12:      end for
13:   Output:  $\mathbf{G}, \mathbf{W}$ 
14: end procedure

```

---

3.1.3. *Semi Supervised Nonnegative Matrix Factorization (SSNMF)*. SSNMF factorizes a data matrix  $\mathbf{X} \in \mathbb{R}_+^{m \times n}$  (where  $m$  is the number of terms and  $n$  is the number of samples) and a label matrix  $\mathbf{Y} \in \mathbb{R}_+^{k \times n}$  (where  $k$  is the number of classes and each column of  $\mathbf{Y}$  is a binary vector) with the following objective function:

$$\min_{\mathbf{W}, \mathbf{H}} \|\mathbf{W} \odot (\mathbf{X} - \mathbf{A}\mathbf{S})\|_F^2 + \lambda \|\mathbf{L} \odot (\mathbf{Y} - \mathbf{B}\mathbf{S})\|_F^2 \quad (6)$$

where  $\lambda$  is a weight parameter.

Here,  $\mathbf{W} \in \mathbb{R}_+^{m \times n}$  is a binary weight matrix such that

$$\mathbf{W}_{ij} = \begin{cases} 1, & \text{if } \mathbf{X}_{ij} \text{ is observed} \\ 0, & \text{otherwise} \end{cases} \quad (7)$$



$\mathbf{A} \in \mathbb{R}_+^{m \times r}$  is the basis matrix for  $\mathbf{X}$  where  $r$  is the number of classes, and  $\mathbf{S} \in \mathbb{R}_+^{r \times n}$  is the feature matrix or the joint-factorization matrix.  $\mathbf{L}$  is a weight matrix which handles missing entries and is 1 if the label is known and 0 otherwise.  $\mathbf{B}$  is a basis matrix for the label matrix  $\mathbf{Y}$  such that  $\mathbf{B} \in \mathbb{R}_+^{k \times r}$ .

The updates are given as follows [7]:

$$\mathbf{A} \leftarrow \mathbf{A} \odot \frac{[\mathbf{W} \odot \mathbf{X}] \mathbf{S}^\top}{[\mathbf{W} \odot \mathbf{A} \mathbf{S}] \mathbf{S}^\top} \quad (8)$$

$$\mathbf{B} \leftarrow \mathbf{B} \odot \frac{[\mathbf{L} \odot \mathbf{Y}] \mathbf{S}^\top}{[\mathbf{L} \odot \mathbf{B} \mathbf{S}] \mathbf{S}^\top} \quad (9)$$

$$\mathbf{S} \leftarrow \mathbf{S} \odot \frac{\mathbf{A}^\top [\mathbf{W} \odot \mathbf{X}] + \lambda \mathbf{B}^\top [\mathbf{L} \odot \mathbf{Y}]}{\mathbf{A}^\top [\mathbf{W} \odot \mathbf{A} \mathbf{S}] + \lambda \mathbf{B}^\top [\mathbf{L} \odot \mathbf{B} \mathbf{S}]} \quad (10)$$

**3.2. Our Extension: Kernel SSNMF.** We then focused on our modification for the Convex NMF algorithm which, analogous to Convex NMF, restricts the matrix  $\mathbf{F}$  to be a convex combination of the data points of  $\mathbf{Z}$ , where  $\mathbf{Z}$  is the data matrix. Our entries for our data matrix were unconstrained and could be either positive or negative. We kernelized the data matrix by mapping the column vectors (data vectors) into a different space as follows:

$$\mathbf{x}_i \rightarrow \phi(\mathbf{x}_i), \quad \text{for } i = 1, 2, \dots, n$$

Additionally, we made the method semi-supervised by stacking the label matrix,  $\mathbf{Y}$ , scaled by some constant  $\lambda$ , with the kernelized matrix  $\phi(\mathbf{X})$ . Our desired matrix factorization problem became

$$\mathbf{Z} = \mathbf{Z} \mathbf{W} \mathbf{G}^\top,$$

where

$$\mathbf{Z} = \begin{bmatrix} \phi(\mathbf{X}) \\ \lambda \mathbf{Y} \end{bmatrix}.$$

$\mathbf{A} = \phi(\mathbf{X}) \mathbf{W}$  and  $\mathbf{B} = \lambda \mathbf{Y} \mathbf{W}$ , and  $\mathbf{S}$  is  $\mathbf{G}^\top$ . Here,  $\mathbf{A}$ ,  $\mathbf{B}$ , and  $\mathbf{S}$  have the same meanings for interpreting data as those for SSNMF 3.1.3.

Our objective function for minimization became

$$\min \|\mathbf{Z} - \mathbf{Z} \mathbf{W} \mathbf{G}^\top\|^2 = \text{Tr}(\mathbf{D} - 2\mathbf{D} \mathbf{W} \mathbf{G}^\top + \mathbf{G} \mathbf{W}^\top \mathbf{D} \mathbf{W} \mathbf{G}^\top),$$

where  $\mathbf{D} = \phi^\top(\mathbf{X}) \phi(\mathbf{X}) + \lambda^2 \mathbf{Y}^\top \mathbf{Y}$ .  $\phi^\top(\mathbf{X}) \phi(\mathbf{X})$  is our kernel matrix, so the objective function did not depend on  $\phi(\mathbf{X})$ , but it depended on the kernel matrix.

We followed the update rules for Convex NMF after kernelization. Our pseudocode is provided below:

**3.3. Convergence for Algorithms.** In the present subsection, we will discuss the necessary theorems that demonstrate the convergence of Semi-NMF and Convex NMF, and it will ultimately help us show that our algorithm converges.

**3.3.1. Convergence of Semi NMF.** The following theorems are provided from the Jordan paper [6] for the convergence of Semi-NMF 3.1.1 and in the next part for Convex NMF 3.3.2.

**Theorem 1.** 1) Fixing  $\mathbf{F}$ , the residual  $\|\mathbf{X} - \mathbf{F} \mathbf{G}^\top\|^2$  decreases monotonically (i.e. it is nondecreasing) under the update rule for  $\mathbf{G}$ . 2) Fixing  $\mathbf{G}$ , the update rule for  $\mathbf{F}$  gives the optimal solution to  $\min_{\mathbf{F}} \|\mathbf{X} - \mathbf{F} \mathbf{G}^\top\|^2$ .

**Proposition 1.** The limiting solution of the update rule of 2 satisfies the KKT condition.

**Proposition 2.** The residual of Semi-NMF is monotonically decreasing (nonincreasing) under the update rule for  $\mathbf{G}$  for fixed  $\mathbf{F}$ .

---

**Algorithm 3** Kernel Semi-Supervised Nonnegative Matrix Factorization (Kernel SSNMF)

---

```
1: procedure KERNELSSNMF( $\mathbf{X}, \mathbf{Y}, \mathbf{S}, \text{max\_iter}, \text{tol}, \text{kernel}, \lambda$ )
2:   Input:
      •  $\mathbf{X}$ : Data matrix of dimensions  $p \times n$ 
      •  $\mathbf{Y}$ : Label matrix
      •  $\mathbf{S}$ : Topic matrix
      • max_iter: Maximum number of iterations
      • tol: Tolerance for convergence
      • kernel: Kernel function
      •  $\lambda$ : Regularization parameter
3:   Initialization:
      • KernelizedMatrix  $\leftarrow \text{kernel}(\mathbf{X}, \text{kernel})$ 
      • Matrix  $\leftarrow \text{KernelizedMatrix} + \lambda^2 \mathbf{Y}^\top \mathbf{Y}$ 
      • Initialize  $\mathbf{S}$  using Semi-NMF with Matrix
4:   Iterative Updates:
5:   ( $\mathbf{G}, \mathbf{W}$ )  $\leftarrow \text{ConvexNMF}(\text{Matrix}, \mathbf{S}, \text{max\_iter}, \text{tol})$ 
6:   Output:  $\mathbf{G}, \mathbf{W}$ 
7: end procedure
```

---

3.3.2. *Convergence of Convex NMF.* The convex NMF algorithm converges as well due to similar theorems[6].

**Theorem 2.** Fixing  $\mathbf{G}$ , under the update rule for  $\mathbf{W}$  of 5, 1) the residual  $\|\mathbf{X} - \mathbf{X}\mathbf{W}\mathbf{G}^\top\|^2$  decreases monotonically (nonincreasing), and 2) the solution converges to a KKT fixed point.

**Proposition 3.** The limiting solution of the update rule for  $\mathbf{W}$  5 satisfies the KKT condition.

**Proposition 4.** The residual for Convex NMF decreases monotonically (that is, it is nonincreasing). Thus, the algorithm converges.

3.3.3. *Convergence of Kernel SSNMF.* Using the aforementioned convergence theorems, it is easy to show that Kernel SSNMF converges.

**Theorem 3.** Since  $\mathbf{Z} = \begin{pmatrix} \phi(\mathbf{X}) \\ \lambda \mathbf{Y} \end{pmatrix}$ , then, fixing  $\mathbf{G}$ , under the update rule for  $\mathbf{W}$  of 5, 1) the residual  $\|\mathbf{Z} - \mathbf{Z}\mathbf{W}\mathbf{G}^\top\|^2$  decreases monotonically (nonincreasing), and 2) the solution converges to a KKT fixed point.

*Proof.* The matrix  $\mathbf{Z}$  is a special matrix which can be partitioned into  $\phi(\mathbf{X})$  and  $\lambda \mathbf{Y}$ . Since we apply the iterative update rules for Convex NMF, we can extend the proofs from Theorem 1 2 which is applicable to any arbitrary matrix  $\mathbf{X}$  to conclude that our algorithm converges. □

**3.4. Theory for Reconstructing Matrix.** In this section, we build on Lee-Seung's[7] method for reconstructing the feature matrix  $\mathbf{S}_{\text{test}}$  for SSNMF and extend the analysis to reconstruct similar matrices for Convex NMF and Kernel SSNMF. Reconstructing the feature matrix is important because it helps us perform Support Vector Machine (SVM) classification for our algorithms.

For SSNMF, the paper reconstructs the feature matrix  $\mathbf{S}_{\text{test}}$  by computing  $\mathbf{S}_{\text{test}} = \mathbf{A}^\dagger \mathbf{S}_{\text{train}}$ , where  $\mathbf{S}_{\text{test/train}}$  is obtained from SSNMF on the training data set.

We now extend the analysis to construct feature matrices for Convex NMF and Kernel SSNMF.

3.4.1. *Reconstruction for Convex NMF.*

**Theorem 4.** Since  $\mathbf{A} = \mathbf{X}_{\text{train}} \mathbf{W}$ , then the  $\mathbf{S}_{\text{test}}$  matrix is given by

$$\mathbf{S}_{\text{test}} = \mathbf{A}^\dagger \mathbf{X}_{\text{test}},$$

where  $\mathbf{A}^\dagger$  denotes the Moore-Penrose pseudoinverse of  $\mathbf{A}$ , and

$$\mathbf{A}^\dagger = \begin{cases} \mathbf{W}^\dagger \mathbf{X}_{\text{train}}^T (\mathbf{X}_{\text{train}} \mathbf{X}_{\text{train}}^T)^{-1} & \text{if } \mathbf{X}_{\text{train}} \text{ is a wide matrix,} \\ \mathbf{W}^\dagger (\mathbf{X}_{\text{train}}^T \mathbf{X}_{\text{train}})^{-1} \mathbf{X}_{\text{train}}^T & \text{if } \mathbf{X}_{\text{train}} \text{ is a tall matrix.} \end{cases}$$

*Proof.* We considered the expression for  $\mathbf{A}$ . We took the pseudoinverse of  $\mathbf{A}$  to obtain  $\mathbf{W}^\dagger \mathbf{X}_{\text{train}}^\dagger$ . We then have two cases for the pseudoinverse of  $\mathbf{X}_{\text{train}}$ :

$$\mathbf{X}_{\text{train}}^\dagger = \begin{cases} \mathbf{X}_{\text{train}}^T (\mathbf{X}_{\text{train}} \mathbf{X}_{\text{train}}^T)^{-1} & \text{if } \mathbf{X}_{\text{train}} \text{ is a wide matrix,} \\ (\mathbf{X}_{\text{train}}^T \mathbf{X}_{\text{train}})^{-1} \mathbf{X}_{\text{train}}^T & \text{if } \mathbf{X}_{\text{train}} \text{ is a tall matrix.} \end{cases}$$

Upon substituting the appropriate pseudoinverse depending on whether the matrix is tall or wide, we obtained our desired result.  $\square$

Equivalently, we constructed an alternative expression for our test label matrix:

**Theorem 5.** Since  $\mathbf{X}_{\text{train}} = \mathbf{A} \mathbf{G}_{\text{train}}^T$ , then the  $\mathbf{S}_{\text{test}}$  matrix is given by

$$\mathbf{S}_{\text{test}} = \mathbf{A}^\dagger \mathbf{X}_{\text{test}},$$

where  $\mathbf{A}^\dagger$  denotes the Moore-Penrose pseudoinverse of  $\mathbf{A}$ , and

$$\mathbf{A}^\dagger = \begin{cases} \mathbf{G}_{\text{train}}^T \mathbf{X}_{\text{train}}^T (\mathbf{X}_{\text{train}} \mathbf{X}_{\text{train}}^T)^{-1} & \text{if } \mathbf{X}_{\text{train}} \text{ is a wide matrix,} \\ \mathbf{G}_{\text{train}}^T (\mathbf{X}_{\text{train}}^T \mathbf{X}_{\text{train}})^{-1} \mathbf{X}_{\text{train}}^T & \text{if } \mathbf{X}_{\text{train}} \text{ is a tall matrix.} \end{cases}$$

*Proof.* We rewrote the expression as  $\mathbf{X}_{\text{train}} = \mathbf{A} \mathbf{G}_{\text{train}}^T$ , where  $\mathbf{A}^\dagger = \mathbf{G}_{\text{train}}^T \mathbf{X}_{\text{train}}^\dagger$ . To compute the pseudoinverse of  $\mathbf{A}$ , we left-multiplied both sides by the pseudoinverse of  $\mathbf{A}$ , and right-multiplied both sides by the pseudoinverse of  $\mathbf{X}_{\text{train}}$  to obtain  $\mathbf{A}^\dagger = \mathbf{G}_{\text{train}}^T \mathbf{X}_{\text{train}}^\dagger$ . We then substituted the expression for the pseudoinverse as given in the theorem for a tall and wide matrix to obtain the desired result.  $\square$

### 3.4.2. Reconstruction for Kernel SSNMF.

**Theorem 6.** Since  $\mathbf{A} = \phi(\mathbf{X}_{\text{train}}) \mathbf{W}$ , then the  $\mathbf{S}_{\text{test}}$  matrix was given by

$$\mathbf{S}_{\text{test}} = \mathbf{A}^\dagger \phi(\mathbf{X}_{\text{test}}),$$

where  $\mathbf{A}^\dagger$  denotes the Moore-Penrose pseudoinverse of  $\mathbf{A}$ , and

$$\mathbf{A}^\dagger = \begin{cases} \mathbf{W}^\dagger (\phi(\mathbf{X}_{\text{train}})^T \phi(\mathbf{X}_{\text{train}}))^{-1} \phi(\mathbf{X}_{\text{train}})^T, & \text{if } \phi(\mathbf{X}_{\text{train}}) \text{ is a tall matrix,} \\ \mathbf{W}^\dagger \phi(\mathbf{X}_{\text{train}})^T (\phi(\mathbf{X}_{\text{train}}) \phi(\mathbf{X}_{\text{train}})^T)^{-1}, & \text{if } \phi(\mathbf{X}_{\text{train}}) \text{ is a wide matrix.} \end{cases}$$

*Proof.* We considered the expression for  $\mathbf{A}$ . We took the pseudoinverse of  $\mathbf{A}$  to obtain  $\mathbf{W}^\dagger \phi(\mathbf{X}_{\text{train}})^\dagger$ . We then have two cases for the pseudoinverse of  $\phi(\mathbf{X}_{\text{train}})$ :

$$\phi(\mathbf{X}_{\text{train}})^\dagger = \begin{cases} (\phi(\mathbf{X}_{\text{train}})^T \phi(\mathbf{X}_{\text{train}}))^{-1} \phi(\mathbf{X}_{\text{train}})^T, & \text{if } \phi(\mathbf{X}_{\text{train}}) \text{ is a tall matrix,} \\ \phi(\mathbf{X}_{\text{train}})^T (\phi(\mathbf{X}_{\text{train}}) \phi(\mathbf{X}_{\text{train}})^T)^{-1}, & \text{if } \phi(\mathbf{X}_{\text{train}}) \text{ is a wide matrix.} \end{cases}$$

Upon substituting the appropriate pseudoinverse depending on whether the matrix is tall or wide, we obtained our desired result.  $\square$

Equivalently, we constructed an alternative expression for our test label matrix:

**Theorem 7.** Since  $\mathbf{A} = \phi(\mathbf{X}_{\text{train}}) \mathbf{W}$ , then the  $\mathbf{S}_{\text{test}}$  matrix was given by

$$\mathbf{S}_{\text{test}} = \mathbf{A}^\dagger \phi(\mathbf{X}_{\text{test}}),$$

where  $\mathbf{A}^\dagger$  denotes the Moore-Penrose pseudoinverse of  $\mathbf{A}$ , and

$$\mathbf{A}^\dagger = \begin{cases} (\phi(\mathbf{X}_{\text{train}})^T \phi(\mathbf{X}_{\text{train}}))^{-1} \phi(\mathbf{X}_{\text{train}})^T, & \text{if } \phi(\mathbf{X}_{\text{train}}) \text{ is a tall matrix,} \\ \phi(\mathbf{X}_{\text{train}})^T (\phi(\mathbf{X}_{\text{train}}) \phi(\mathbf{X}_{\text{train}})^T)^{-1}, & \text{if } \phi(\mathbf{X}_{\text{train}}) \text{ is a wide matrix.} \end{cases}$$

*Proof.* We knew from our Kernel SSNMF theory that  $\phi(\mathbf{X}_{\text{train}}) = \mathbf{A}\mathbf{G}_{\text{train}}^T$ . To compute the pseudoinverse of  $\mathbf{A}$ , we left-multiplied both sides by the pseudoinverse of  $\mathbf{A}$ , and right-multiplied both sides by the pseudoinverse of  $\phi(\mathbf{X}_{\text{train}})$  to obtain  $\mathbf{A}^\dagger = \mathbf{G}_{\text{train}}^T \phi(\mathbf{X}_{\text{train}})^\dagger$ . We substituted the expression for the pseudoinverse as given in the theorem for a tall and wide matrix and obtained the desired result.  $\square$

However, for all practical purposes, we recommend applying Kernel SSNMF on tall rather than wide matrices. This is because applying Kernel SSNMF on a wide matrix entails computing  $\phi(\mathbf{X}_{\text{train}})$  individually which will be computationally expensive versus for a tall matrix where we compute the inner product with  $\phi(\mathbf{X}_{\text{test}})$  and mitigate the need for computing  $\phi(\mathbf{X}_{\text{train}})$  individually. Furthermore, we saved computational cost for a tall matrix because we computed the kernel matrix where the number of documents would be far less than the number of features, so the dimensions of the kernel matrix would be number of documents by number of documents.

**3.5. Text embeddings to text conversions.** In this subsection, we analyze a methodology to invert embeddings back to text for the GPT-ada 002 standard [8]. Given a text sequence of tokens  $x \in \mathbb{V}$ , a text encoder maps  $x$  to a fixed length embedding vector  $e \in \mathbb{R}$ . Text recovery relies on similarity with the ground truth [8], so the objective function becomes:

$$\hat{x} = \arg \max_x \cos(\phi(x), e)$$

The paper [8] uses a maximum likelihood estimator  $p(x|e; \theta)$ , by learning  $\theta$  via maximum likelihood:

$$\hat{x} = \arg \max_x \mathbb{E} \cos(\phi(x), e)$$

The method guesses an initial hypothesis and iteratively refines the hypothesis:

$$\begin{aligned} p(x^{(t+1)}|e) &= \sigma_{x^{(t)}} p(x^{(t)}|e) p(x^{(t)}|e, x^{(t)}, \hat{e}^{(t)}) \\ \hat{e}^{(t)} &= \phi(x^{(t)}) \end{aligned}$$

with a base case of the simple learned inversion:

$$p(x^{(0)}|e) = p(x^{(t+1)}|e, \phi, \phi, \phi(\phi))$$

The encoder inverts the inputs for some nonlinear activation function for an embedding  $e \in \mathbb{R}$ :

$$\text{EmbToSeq}(e) = W_2 \sigma(W_1, e)$$

where  $W_1 \in \mathbb{R}^d$  and  $W_2 \in \mathbb{R}^{\text{sd}_{\text{enc}}}$ .

$$\text{concat}(\text{EmbToSeq}(e), \text{EmbToSeq}(\hat{e}^{(t)}), \text{EmbToSeq}(e - \hat{e}^{(t)}), (w_1, \dots, w_n))$$

The Vec2Text package is utilized to convert embeddings to text for pre-trained transformers.

## 4. RESULTS

**4.1. Converting embeddings back to text.** Here, we perform inversions of the text embeddings to text to do analyze what specific information GPT is encoding in the embeddings. An example of a case we get before and after conversions as follows:

### Before:

"Donovan Allen challenges his aggravated first-degree murder conviction for killing his mother, Sharon Cox. He argues that there was insufficient evidence to show that the murder was premeditated during a robbery. Allen also claims that the trial court made numerous evidentiary errors and denied his motions for mistrial and a new trial. Additionally, he alleges violations of his right to counsel and prosecutorial

misconduct, resulting in cumulative errors that denied him a fair trial. The Court of Appeal affirms the conviction, finding no errors in the trial proceedings and upholding the facts of the case, which include Allen's 911 call reporting his mother's death. The case involves Allen, who was initially taken into custody on March 10, 2000, for the murder of Cox based on a statement he provided describing the events. He was advised of his Miranda rights, assigned an attorney, and was not cooperative with the police. Allen was charged with first-degree murder on March 13, 2000. Later, on April 9, 2001, Allen dictated another statement admitting to killing Cox. Allen claimed that he felt coerced into giving the confession and that he was promised a brief visit with his fiancée if he cooperated in locating a cashbox linked to the case. Allen testified that the police were unresponsive to his requests for an attorney and that he only signed a waiver form for his rights without expressing a desire to talk. Ultimately, Allen's testimonies raise questions about the voluntariness of his confession, the validity of the Miranda waiver, and potential coercion by the police in obtaining the statement."

**After:** "The jury challenges his first-degree murder conviction and his conviction for the murder of his mother. The jury also challenges his first-degree murder conviction and his conviction for the murder of his mother. The jury argues that Anderson's misdemeanor conviction was not justified by a wrongful premeditation of evidence. The jury also argues that the evidence was not sufficient to justify the murder of his mother. In a summary judgment, the jury affirmed Anderson's motion for summary judgment, denying the motion for summary judgment. The jury also argues that the jury affirmed the jury's motion"

We next perform LDA and various other tasks on the text to analyze for specific information being encoded in these embeddings' retrieved text.

**4.2. Experimental Setup.** The methodology section provides us a theoretical support for using Kernel SSNMF in classification tasks. Specifically, we followed the testing procedure below for all our experiments on both external data and the embedding data we generated.

- The original data matrix  $\mathbf{X}$  is split into train set  $\mathbf{X}_{\text{train}}$  and test set  $\mathbf{X}_{\text{test}}$ . 80-20 split is used here. The label vectors  $\mathbf{y}_{\text{train}}$  and  $\mathbf{y}_{\text{test}}$  are generated accordingly. Note that for regular SSNMF we also need to pass in the one-hot encoded binary label matrices, denoted here as  $\mathbf{Y}_{\text{train}}$  and  $\mathbf{Y}_{\text{test}}$ .
- Training data  $\mathbf{X}_{\text{train}}$  and  $\mathbf{y}_{\text{train}}$  (or  $\mathbf{Y}_{\text{train}}$ ) are passed into the selected topic modeling algorithm, and a reduced feature matrix, denoted as  $\mathbf{S}_{\text{train}}$ , is returned.
- $\mathbf{S}_{\text{test}}$  is constructed using either Theorem 6 or 7.
- $\mathbf{S}_{\text{train}}$  can be used to train a classifier, for which we used SVM with grid search. We then computed classification accuracy using testing data  $\mathbf{y}_{\text{test}}$  and SVM predictions for  $\mathbf{S}_{\text{test}}$ .
- To maintain consistency, we have set regularization parameters  $\lambda = 1$  and maximum number of iterations to be 1,000 universally.

**4.3. Kernel SSNMF General Performance.** To verify that Kernel SSNMF works for dimension reduction and classification tasks, we compared its performance with benchmark algorithms including Convex NMF and regular SSNMF on some real datasets from external sources. Here we provide brief descriptions of these data, according to the sites on which they are published:

- Gene Expression Cancer RNA-Seq [9]: This multivariate, fully-labeled dataset from UCI Machine Learning Repository is part of the RNA-Seq (HiSeq) PANCAN data set. It is a random extraction of gene expressions of patients having different types of tumor: BRCA, KIRC, COAD, LUAD, and PRAD. It has 801 instances and 20,531 features. Due to the large dimension of this data, we randomly extracted 200 instances to save computation costs.
- GEMLeR Colon Kidney [10]: This bivariate, fully-labeled dataset can be found at OpenML. The GEMLeR repository provides a collection of gene expression datasets that can be used for benchmarking gene expression oriented machine learning algorithms. Each gene expression sample

in GEMLeR repository comes from a large publicly available expO (Expression Project For Oncology) repository by International Genomics Consortium. This data has 546 instances and 10,935 numeric features. We also randomly extracted 200 instances from this data.

Both datasets contain high-dimensional numeric features and were standardized during pre-processing. The experiments performed on these datasets resulted in the accuracy scores shown by Table 1 and 2, where we compared the performance of Kernel SSNMF (with linear and polynomial kernels, since they tended to significantly outperform RBF and sigmoid kernels in these cases) with Convex NMF and SSNMF. As number of topics increases, the accuracy of Kernel SSNMF steadily increases and stabilizes, with Polynomial Kernel SSNMF achieving a very high level of accuracy. The experiments demonstrated that Kernel SSNMF is not only comparable to but can also outperform the benchmark algorithms with appropriate choice of number of topics, which provided practical support for our next step of evaluating its capability to learn the LLM embeddings of legal documents.

	Convex NMF	SSNMF	Kernel SSNMF (Linear)	Kernel SSNMF (Poly)
<b>5 Topics</b>	0.55	0.85	0.45	0.9
<b>10 Topics</b>	0.8	0.95	1.0	1.0
<b>15 Topics</b>	0.8	0.975	0.975	1.0
<b>20 Topics</b>	0.75	0.975	0.9	1.0
<b>25 Topics</b>	0.725	1.0	0.975	1.0

**Table 1** SVM Testing Accuracy on UCI Genomic Data

	Convex NMF	SSNMF	Kernel SSNMF (Linear)	Kernel SSNMF (Poly)
<b>5 Topics</b>	0.925	0.95	0.925	0.95
<b>10 Topics</b>	0.825	0.95	0.925	0.925
<b>15 Topics</b>	0.95	0.95	0.975	0.95
<b>20 Topics</b>	0.95	0.95	0.95	0.95
<b>25 Topics</b>	0.875	0.95	0.975	0.95

**Table 2** SVM Testing Accuracy on GEMLeR Colon Kidney Data

#### 4.4. Predicting Wrongful Convictions.

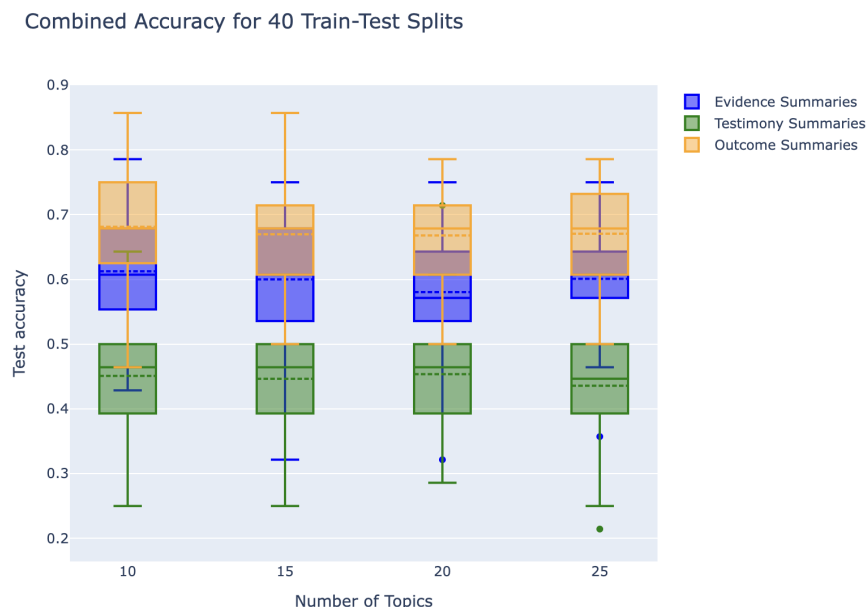
4.4.1. *Embeddings Data Preparation.* After collecting all of the case data, we extracted the text data from .pdf files by converting them to .txt files. We then preprocessed the data by removing common stop words and punctuation from the text and performed lemmatization to make sure the text was concise and effective.

We entered the following three prompts to get different summaries from ChatGPT:

- Prompt 1: *"Summarize evidence from this case."*
- Prompt 2: *"Evaluate how the accuracy and reliability of eyewitness testimony influenced the outcome of this case, considering factors such as the witnesses' credibility, consistency, and potential biases."*
- Prompt 3: *"Give me a good summary for this case to help the judge decide whether exonerated or non-exonerated."*

We took the summaries and put them into GPT-3.5 to convert each summary text into an embedding vector of length 1536 which captures semantic meaning for each of the 140 documents. We obtained a data matrix of dimension  $1536 \times 140$ , where 1536 contributes to the number of features in the data, and each document summary is a column vector to get a tall matrix.

To assess the classification accuracy of the embeddings corresponding to each summary type, we used Convex NMF. For each type of summary embedding, we implemented Convex NMF across five different random states to ensure consistent results. We also experimented with different numbers of topics in the Convex NMF input, ranging from 10 to 25 topics. Figure 4 shows the boxplots depicting the classification accuracy of the embeddings for each type of summary across various random states and numbers of topics.



**Figure 4** Classification accuracy using Convex NMF for various summary embeddings is shown for five different train-test splits and varying number of topics. The outcome summaries demonstrate the best classification accuracy followed by evidence summaries and then testimony summaries.

From Figure 4, we observe that the summaries resulting from the outcome prompt (3) perform significantly better than the summaries obtained from the other two prompts when we run Convex NMF. The median classification accuracy for the embeddings from the outcome prompt seems to be hovering around 0.7 across the various number of topics, which is higher than the 0.6 median accuracy and 0.45 median accuracy attained by the evidence and testimony embeddings, respectively. It is important to note the potential for drastic differences in classification accuracy that depend solely on the quality of the input prompt. For instance, the outcome embeddings performed significantly better than the testimony embeddings (the difference between their median accuracies was 0.25), indicating that the outcome prompt produced summaries that had more of an impact on classification than the testimony prompt had.

Of the three summary prompts we analyzed, the one that returned the best classification accuracy was the outcome Prompt (3), which read *"Give me a good summary for this case to help the judge decide whether exonerated or non-exonerated."* Because this prompt performed the best, we used the corresponding embeddings data from this prompt in the rest of our analysis. An example summary response to this prompt from the exonerated case Commonwealth v. Barnes is shown below:

*"The appeal case in Virginia occurred in Chesapeake in the Commonwealth, with the decision given on May 10, 2005 by Judge James Benton. The appeal was from the Circuit Court of the City of Portsmouth, involving Johnny Morrison with Judge James Broccoletti, and Zoby Broccoletti as the defense attorneys. The appellant was Kathleen Martin, Assistant*

*Attorney General, against the appellee, Jerry Kilgore, Attorney General. During the trial, the jury convicted Lamar Edward Barnes of murder and malicious wounding charges. The case concerned events at Mark King's residence involving Barnes, Mike Artis, and Amy M. Barnes argued that the trial judge made errors in admitting testimony about the victim's pregnancy and with the jury's verdict deliberation. However, the Court found the errors to be harmless and within the judge's discretion, upholding the conviction."*

This was a case for a non-exonerated individual:

*"In this case, the appellant, Jane Moran Williamson, appealed a decision from the circuit court of Mingo County, West Virginia, where she was sentenced to life in the penitentiary for first-degree murder. The appellant argued that the verdict was not supported by substantial evidence and that the prosecuting attorney abused their position during the trial. The appellant also claimed that new evidence was discovered, there was a material variance in the charges contained in the indictment, and that certain items seized from her car should have been suppressed. Additionally, the appellant alleged prosecutorial misconduct during the trial. However, upon review, the court found no reversible error and affirmed the judgment of the circuit court."*

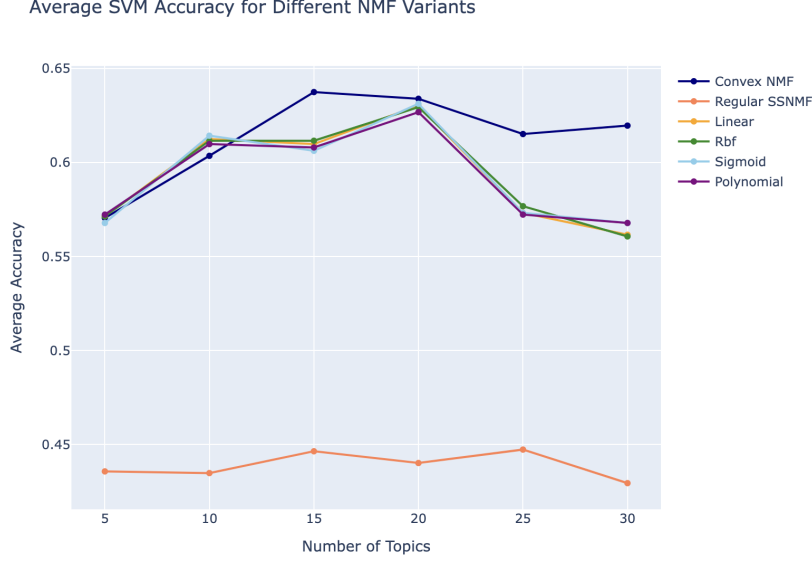
4.4.2. *Kernel SSNMF Performance on Embeddings.* To reduce randomness that may arise in the train-test split process, we repeated our experiments on 6 random states.

Figure 5 shows the results of applying the testing procedure to the embeddings, from which we could tell that while Convex NMF outperforms regular SSNMF for the selected range of topics, both are generally lower than Kernel SSNMF regardless of the choice of kernels. Further, Kernel SSNMF has higher accuracy for each quartile and is more resistant to small changes in the number of topics, with a lower chance of dropping below 0.5. Figure 6 gives a clearer comparison by computing the average test accuracy of 6 random states for each algorithms with a selected number of topics.



**Figure 5** SVM accuracy of different NMF variants (Convex NMF, regular SSNMF, and Kernel SSNMF with different kernels) on LLM embeddings with multiple random states. Each scatter point indicates the SVM testing accuracy of a random state with corresponding choice of algorithm and number of topics.





**Figure 6** Average SVM accuracy of different NMF variants (Convex NMF, regular SSNMF, and Kernel SSNMF with different kernels) on LLM embeddings for multiple random states.

The experimental results demonstrated that Kernel SSNMF is a functional algorithm with the capacity to more accurate and reliable predictions of wrongful convictions given LLM embeddings data. Some benefits of Kernel SSNMF are that it converges quickly, does not impose non-negative constraints on the data input, effectively incorporates label information as in SSNMF, and has more flexibility in fine-tuning parameters. However, although Kernel SSNMF performed relatively well on our data compared to other algorithms, there are still areas for improvement, specifically with the quality of data.

## 5. DISCUSSION

**5.1. Future Work.** Overall, our study provides valuable insights into the disparities and challenges within the criminal justice system and contributes to ongoing efforts to address these injustices and enhance the fairness and accuracy of the criminal justice system using AI and ML techniques. However, the limitations of our dataset indicate that further research into expanding the data and refining our models is necessary. Our dataset comprises 140 cases, which has supported our research so far. Nevertheless, the relatively small size of this data restricts the generalizability of our results. Therefore, expanding our dataset to include a broader and more diverse range of cases will allow for us to gain deeper insights and to improve the reliability of our findings.

Additionally, our exploration of document-level embeddings revealed limitations in their effectiveness for deep exploration of the data. While these embeddings are useful in certain contexts, they did not provide enough detail for our specific research needs. Therefore, future research should investigate better approaches to text embeddings that can capture details within the legal documents more effectively.

Furthermore, our current method of Term Frequency-Inverse Document Frequency (TF-IDF) resulted in a lower accuracy than anticipated, which suggests that we need to further explore alternative techniques for converting embeddings back into text. Identifying these methods will greatly help in improving the overall performance and interpretability of our models.

Lastly, future work should also focus on incorporating advanced LLMs specifically trained on legal texts. As observed in Section 4.4.1, the quality of the prompt given to the LLM affects the relevance of the resulting embeddings, which can significantly impact the effectiveness of eventual classification tasks.

A more specialized Legal LLM has the potential to provide more relevant summaries and embeddings, which can, in turn, lead to higher classification accuracy in predicting wrongful convictions.

## REFERENCES

- [1] M. Mauer, "The crisis of the young african american male and the criminal justice system 1," in *Impacts of incarceration on the African American family*, pp. 199–218, Routledge, 2018.
- [2] "Prisoners in 2022 – Statistical Tables | Bureau of Justice Statistics."
- [3] S. R. Gross, B. O'brien, C. Hu, and E. H. Kennedy, "Rate of false conviction of criminal defendants who are sentenced to death," *Proceedings of the National Academy of Sciences*, vol. 111, no. 20, pp. 7230–7235, 2014.
- [4] "The Innocence Center - Securing Freedom For The Innocent."
- [5] "Exoneration Detail List."
- [6] C. H. Ding, T. Li, and M. I. Jordan, "Convex and semi-nonnegative matrix factorizations," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 1, pp. 45–55, 2010.
- [7] H. Lee, J. Yoo, and S. Choi, "Semi-supervised nonnegative matrix factorization," *IEEE Signal Processing Letters*, vol. 17, no. 1, pp. 4–7, 2009.
- [8] J. X. Morris, V. Kuleshov, V. Shmatikov, and A. M. Rush, "Text embeddings reveal (almost) as much as text," *arXiv preprint arXiv:2310.06816*, 2023.
- [9] S. Fiorini, "gene expression cancer RNA-Seq." UCI Machine Learning Repository, 2016. DOI: <https://doi.org/10.24432/C5R88H>.
- [10] G. Stiglic and P. Kokol, "Stability of ranked gene lists in large microarray analysis studies," *Journal of biomedicine biotechnology*, vol. 2010, p. 616358, 2010.