An Intelligent Judiciary System

Jheel Mehta
College of Computer and Information Science
Northeastern University
Boston, MA 02115
`mehta.jh@husky.neu.edu`

Abhilash Kadupatil
College of Computer and Information Science
Northeastern University
Boston, MA 02115
`kadupatil.a@husky.neu.edu`

1 Introduction

1.1 Problem we are trying to solve

The United States is renowned for having one of the most sophisticated judicial systems in the world. The U.S. legal system depends on the adversarial system of justice where litigants present their case in front of the neutral party, which allows the judge or jury to determine the truth about the conflict. The truth, however, is that the judiciary system is flawed.
Few of these common problems faced by our current judiciary system are highlighted below.

- As humans, we can't help but see the world through the lens of our own experiences. Research shows that most of the judges are male and mostly white which implies not all demographics are represented in the system and leads to racial discrimination while making a verdict.
- Corruption is one of the issues that plague the court system.
- Human state of mind or other human issues like personal conflict, prejudice, and moods can lead to biased and unfair trials.

Prejudice and bias can be an important factor when *serving justice*. Hence, an objective machine learning algorithm, trained upon years of human experience can go a long way in creating an unbiased judiciary system.

1.2 Why this problem is so hard

Human Experience + Objective AI Algorithm = Unbiased Judiciary System

However easy this solution might be with the technical advancement we have reached, it has its fair share of obstacles to overcome.
Some the major reasons why this problem is so hard to solve are:

1. Algorithmical Bias:

   "Algorithms neutrality" is a popular idea in media. We believe that thanks to artificial intelligence, human bias and prejudice will finally be fought and dealt with.

   That's how it should be in theory. AI doesn't care of one's skin color, sex, or age, and determines whether the person is guilty based solely on his/her actions and facts. However, that's not the case in real life. The thing is, in order to develop an AI algorithm, data sets are used. The information is collected and "fed" to AI to humanize its reasoning.

   What comes out as a side effect is famous as "algorithmical bias". It turns out, AI is almost as objective to gender and racial bias as a human is. The reason to that is that AI learns from what humans have written, filmed, and recorded.

   Also, it is humans who create these models and hence they can easily create a biased model by giving more weight to features like race, gender, age etc while making a decision.

2. Lack of Transparency:

   Another grave issue is the Lack of Transparency. It is well known that government agencies don't develop AI algorithms they use by themselves. They outsource the process to private businesses, which means that the purchaser knows the process of machine decision-making only to a limited degree based on what the owner tells them.

   It is one thing is when no one understands the algorithm of Google Rankings or Netflix recommended movies, we can live without this knowledge. It's not as if people's fates are at stakes like in judicial system.

   To implement AI into legal practices, it is very important that it's model of reasoning is made transparent to a reasonable extent.

1.3 Our contribution

There are a lot of issues to solve before we can successfully integrate ML algorithms in our Judicial Systems. The problem we are trying to solve here is the Lack of Transparency.
Our Model extracts top features from the given dataset and tries to determine their relation with the output, which in this case is whether the convict is guilty or not. Based on these features an assisting system can be made incorporating NLP to build a summary to explain the reasoning behind the decision made.
As we know getting legal aid is very expensive, this system can help people understand their case better before getting involved in the process of appealing and fighting the case in the court.

2 Related Work

The research paper by *Lim et al*[1] talks about the threshold based fairness. They have examined the performance of Naive Bayes in a criminal justice application on a real-world dataset and found that it produced substantially disparate false positive rates for different racial groups.

The research paper by *Fersini et al.* takes an interesting approach to ML-based judiciary system. It highlights the working of a complete intelligent judiciary system (JUMAS)[2]. Its feature set includes Automatic template filling, semantic enrichment of the judicial folder through audio and video processing, enhanced transcription process, help judges, prosecutors not only to save time but also to enhance the quality of their judicial decisions and actions.

The study by *Tim Brennan et al*[5] examines the statistical validation of a risk—need assessment system (Correctional Offender Management Profiling for Alternative Sanctions; COMPAS) that incorporates a range of theoretically relevant criminogenic factors and key factors emerging from meta-analytic studies of recidivism. COMPAS's automated scoring provides decision support for correctional agencies for placement decisions, offender management, and treatment planning.

3 Model/Method

3.1 Overview of our method

Initially, we used Logistic regression for our model. It is the appropriate regression analysis to conduct when the dependent variable is dichotomous (binary). Like all regression analyses, the logistic regression is a predictive analysis. Logistic regression is used to describe data and to explain the relationship between one dependent binary variable and one or more nominal, ordinal, interval or ratio-level independent variables. In our case the output is binary so Logistic Regression fit our requirements.

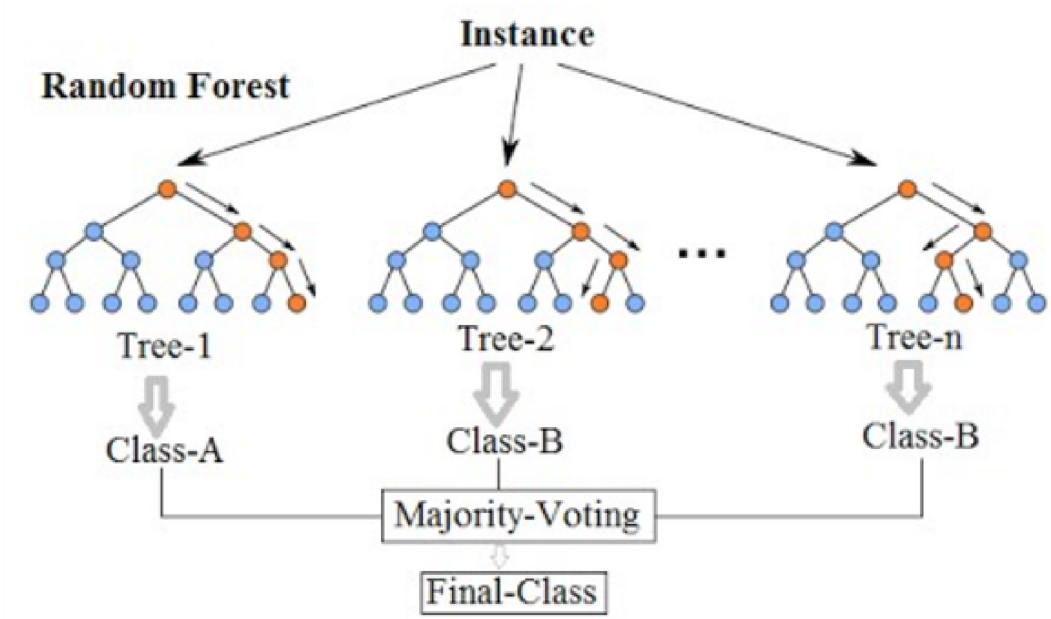
But the accuracy was not as high as expected and since the stakes are high for the problem we are trying to solve, we had to find another model to achieve higher accuracy. So we implemented Random Forest. Random Forest is a flexible, easy to use machine learning algorithm that produces, even without hyper-parameter tuning, a great result most of the time. It is also one of the most used algorithms because it's simplicity and the fact that it can be used for both classification and regression tasks. The accuracy achieved was high enough for us to use in our final model.

Later we used recursive Feature Elimination to extract top features which would, in turn, help us build the reasoning for the desired output. Our model can be summarized in the following steps:

1. Clean the dataset to fill in missing values, enumerate the data and categorize it.
2. Use Random forest on 80 percent of the given dataset to train our model.
3. Test the model on the remaining 20 percent of the data.
4. Finally uses Recursive Feature Elimination to extract top features.

3.2 Formal Definition

1. Random Forest:
   Random forest builds multiple decision trees and merges them together to get a more accurate and stable prediction. Random Forest adds additional randomness to the model while growing the trees. Instead of searching for the most important feature while splitting a node, it searches for the best feature among a random subset of features. This results in a wide diversity that generally results in a better model.



   One of the big problems in machine learning is overfitting, but most of the time this won't happen that easy to a random forest classifier. That's because if there are enough trees in the forest, the classifier won't overfit the model.
   The main limitation of Random Forest is that a large number of trees can make the algorithm too slow and ineffective for real-time predictions. But in our case accuracy is of higher importance and time is not a grave issue in terms of making a decision the legal system works in a systematic fashion, so this limitation is not a deal breaker in our case.

2. Recursive Feature Elimination:
   Given an external estimator that assigns weights to features (e.g., the coefficients of a linear model), the goal of recursive feature elimination (RFE) is to select features by recursively considering smaller and smaller sets of features. First, the estimator is trained on the initial set of features and the importance of each feature is obtained either through a coef-attribute or through a feature-importance- attribute. Then, the least important features are pruned from the current set of features. That procedure is recursively repeated on the pruned set until the desired number of features to select is eventually reached.

4 Experiment

4.1 Dataset

The U.S. Supreme Court Database 2018 Release 2 (SCDB18) is the data used in this study, provided by the Washington University Law. The Supreme Court database is the definitive source for researchers, students, journalists, and citizens interested in the analysis of the United States Supreme Court. [4]

The database has two different sets, namely, Case Centered and Justice Centered data. In the Case Centered data the unit of analysis is the case; i.e., each row of the database contains information about an individual dispute. These data should be used unless the votes of the individuals' justices are of interest. The Justice Centered data include a row for each justice participating in each dispute.

Cleansing of data

- The dataset was transformed and formatted to standardize the data.
- Missing values for were filled with the zero and outliers were removed.
- Some features were enumerated e.g. the feature *chief* was assigned unique numerical for a unique chief name.
- Categorizing of data into different brackets based on the type of data. e.g data related to year were categorized by decade.

4.2 Baselines

For our baseline model, we considered Naive Bayes Classifier. It was the simplicity of the model which got our attention in the first place. Naive Bayes model is easy to build and particularly useful for very large data sets. Along with simplicity, Naive Bayes is known to outperform even highly sophisticated classification methods. We started with Bernoulli Naive Bayes Classifier. It assumes that all our features are binary such that they take only two values. We shortlisted 30 features from the dataset for this baseline model. Our initial training of the model on 80% of the data and on the test set of remaining 20% of data gave an accuracy of 62.5%.

4.3 Quantitative results

To improve the accuracy of the baseline model we tried Forward Feature Selection, with this approach we were able to bring up the accuracy of the Bernoulli Naive Bayes models accuracy to 65.2 % We also tried other different Naive Bayes models to get a sense of the accuracy and reliability of the model. Our initial finding for our study with a different model is as follows:

Model	Accuracy
Naive Bayes : Gaussian	61.1 %
Naive Bayes : Multinomial	38.3 %
Naive Bayes : Bernoulli	62.5 %

Table 1: Accuracy of base Naive Bayes Models

We retrained the above models with selective features which we believed would provide better accuracy & result. The performance of each of these models was then evaluated on the test data. The results are as follows:

We expanded the search for better accuracy by implementing a Logistic regression model and Random Forest Model. Logistic Regression provides an accuracy of 59.9 %, which was less than the accuracy of our baseline model. In contrast, Random forest model seems to provide very good accuracy for our current dataset. The models had the parameter set to default values.

We tried improving upon both these models by performing Hyperparameter tuning.

1) K-Fold Cross Validation

Model	Accuracy
Naive Bayes : Gaussian	88.9 %
Naive Bayes : Multinomial	71 %
Naive Bayes : Bernoulli	65.2 %

Table 2: Accuracy of base Naive Bayes Models with (Selected Features)

Model	Accuracy
Logistic regression	59.9 %
Random Forest Model	96 %

Table 3: Accuracy of base models

2) GridSearch Cross Validation

Both Logistic Regression and Random Forest showed very little improvement with Hyperparameter tuning. Few of the Random Forest parameters for which we tried to get optimal values

1). **n_estimators** : represents the number of trees in the forest.

2). **max_depth** : represents the depth of each tree in the forest.

3). **min_samples_split** : represents splitting of data into training and test data

4). **min_samples_leaf** : The minimum number of samples required to be at a leaf node.

Model	Accuracy
Logistic regression	60.2 %
Random Forest Model	96.7 %

Table 4: Accuracy of base models with Hyperparameter tuning

Feature Selection was our next approach. We tried two feature selection/elimination approaches

1). Forward Feature Selection

2). Recursive Feature Elimination

with these two optimizations, we were able to get a decent improvement to our base models accuracy. The performance of the logistic regression model increased roughly 3% and that of the random forest model by 2%. The accuracy of the models with RFE is as follows:

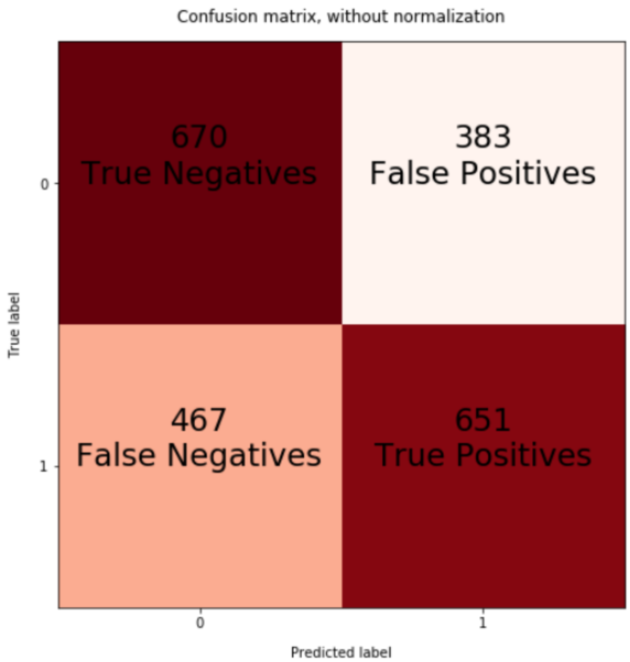
4.4 Qualitative results

4.4.1 Visualizing Model

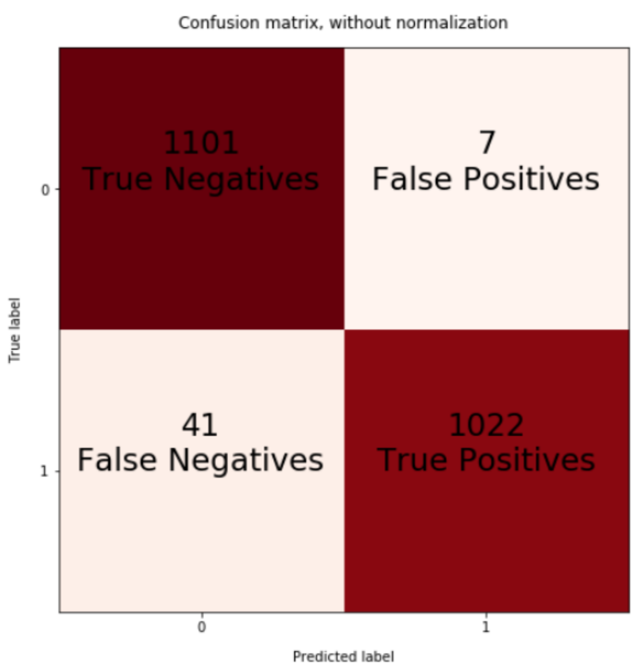
To visualize the performance of a model, we built a confusion matrix. A confusion matrix is a table that is often used to describe the performance of a classification model (or "classifier") on a set of test data for which the true values are known.

Model	Accuracy
Logistic regression	62.9 %
Random Forest Model	98.3 %

Table 5: Accuracy of base models with Recursive Feature Elimination



The Confusion matrix for Logistic Regression shows that True Negatives and True Positives are high but there is a considerable number of false positives and negatives.

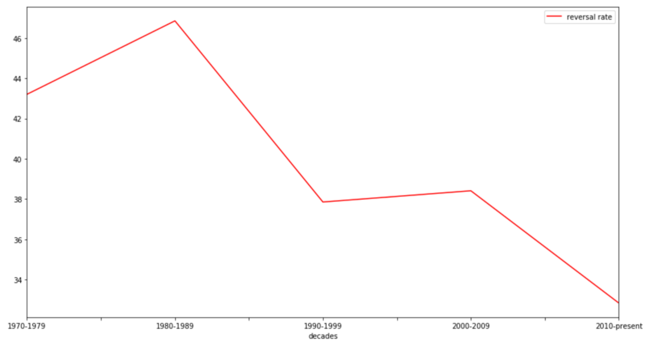


The Confusion matrix for Random Forest shows that True Negatives and True Positives are very high. The number of false positives and negatives are very less. This implies that the model has a very high recall and precision

4.4.2 Insights

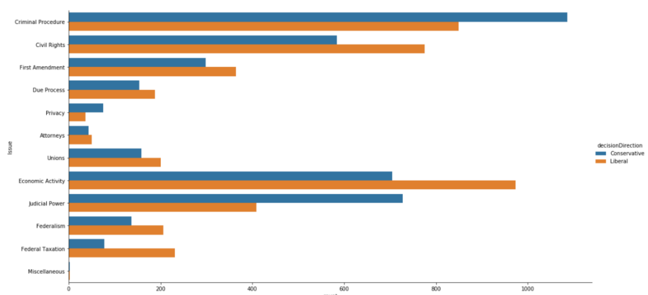
Now that we had a pretty good and accurate model, we ventured further to derive some useful insights from the data. These insights would serve as a foundation for the intelligent judiciary system we hope to build.

- **Reversal Rate**



This decline in Reversal Rates indicate that we are gradually overcoming issues like variability and inconsistencies in the laws and the courts are now firm about the constitutional provisions and important legal issues to be considered while giving a judgment.

- **Decision Direction**



In recent years, there has been a high percentage of liberal opinions. But voting liberally on a state law for drawing and quartering jaywalkers is different from voting to free a murderer. Thus as we can see, only in criminal proceedings, Privacy cases and Judicial power cases, the court seems to be mostly conservative

References

1 Dominick Lim & Torin Rudeen (2017) Minority Report: ML Fairness in Criminality Prediction. Standford University.

2 Fersini, Elisabetta & Messina, Vincenzina & Archetti, Francesco & Cislaghi, Mauro. (2013). Semantics and Machine Learning: A New Generation of Court Management Systems. Communications in Computer and Information Science. 272 CCIS. 382-398. 10.1007/978-3-642-29764-9-26.

3 Hyatt, Jordan & Berk, Richard. (2015). Machine Learning Forecasts of Risk to Inform Sentencing Decisions. Federal Sentencing Reporter. 27. 222-228. 10.1525/fsr.2015.27.4.222.

4 Harold J. Spaeth, Lee Epstein, Andrew D. Martin, Jeffrey A. Segal, Theodore J. Ruger, and Sara C. Benesh. 2018 Supreme Court Database, Version 2018 Release 02. URL:

http://Supremecourtdatabase.org

5 Brennan, Tim & Dieterich, William & Ehret, Beate. (2009). Evaluating the predictive validity of the COMPAS Risk and Needs Assessment System. Criminal Justice and Behavior - CRIM JUSTICE BEHAV. 36. 21-40. 10.1177/0093854808326545.