Incarceration, Charged Type and Sentence Length Predictions

Jiaqin Wu, Chuyuan Zhong, Li Zheng

PPOL564: Final Class Status Update. 11.30.2022

Outline

- Motivation
- Research Questions
- Data
- Methods
- Results thus far
- Limitations/Next Steps

Motivation

- The criminal justice system infiltrated by entrenched institutional racism in American society has led to numerous injustices, in terms of disparities in sentence length and the size and composition of the incarcerated population.
- Even as the pandemic slows down the sentencing process, and the moderate policies during the pandemic reduce incarceration rates in places like Los Angeles, Baltimore, and Philadelphia the factors that drive the sentencing outcome seem to have become more complicated.

Research Questions

- What factors are important in structuring the sentencing outcome?
  - To be specific, what factors influence on judge's decision related to **whether or not to be incarcerated, felony class of the charge at disposition,** and **how long the sentence will be**?

Research Questions

- Predict whether the defendant will be incarcerated or not
- Predict the felony class of the charge at the disposition
- Predict the sentence length of each defendant

Our plan

- We plan to build different machine learning models that take into account the race and gender of defendants, judges' predicted gender, neighborhood livability, and historical incarceration rates as key variables, while also introducing a time indicator to examine how COVID-19 influence court decisions, to predict sentencing outcomes.
- We compare and analyze the models' results and find what features are important for our prediction.
- By doing this, we are able to identify what variables are most relevant in contributing to the sentencing outcomes.

Data Sources

- **sentencing_asof0405.csv Data Set** (from Cook County Open Data)
  - The basic information of defendants
  - Website: https://datacatalog.cookcountyil.gov
- **Police_Stations.csv Data Set** (from the Chicago Data Portal website)
  - The basic information of police stations in Chicago
  - Website: https://data.cityofchicago.org/Public-Safety/Police-Stations/z8bn-74gv
- **Boundaries - Neighborhoods.geojson** Data Set (from the Chicago Data Portal website)
  - The JSON file of the Boundaries of Neighborhoods in Chicago
  - Website: https://data.cityofchicago.org/Facilities-Geographic-Boundaries/Boundaries-Neighborhoods/bbvz-uum9

Methods: Data Acquisition

For our project, we used the Cook County Open Data to access the data set, specifically looking into the sentencing_asof0405 data set, which was initially released on 13th February 2018 but was updated on 28th September 2022, We used the latest versions. Furthermore, we also looked at the Police_Stations data set to use the precise locations of police stations for each police district in Chicago. To visualize the neighborhood distribution in Chicago, we also download the geojson file from the Chicago Data Portal website.

Methods: data cleaning

We selected 14 variables from the data set when predicting whether the defendants will be incarcerated, which charged class they will be sentenced and how long their sentence length will be:

- CHARGE_COUNT: the number of charges associated with the defendant in one case
- DISPOSITION_CHARGED_CLASS: the ultimate class of the charge at the disposition
- AGE_AT_INCIDENT: age of the defendant at the date of the incident
- is_changed_offense: whether the offense's type of defendant is changed
- race variables
  - is_black_derived: whether the race of the defendant is black
  - is_hisp_derived: whether the race of the defendant is Hispanic
  - is_white_derived: whether the race of the defendant is white
  - is_other_derived: other race

Methods: data cleaning

We selected 14 variables from the data set when predicting whether the defendants will be incarcerated, which charged class they will be sentenced and how long their sentence length will be:

- gender variables
  - is_defendant_male: whether the gender of the defendant is male
  - is_judge_male: whether the gender of the judge who sentenced the defendant is male
- is_covid: whether the arrest date of the defendant is in the COVID-19 period
- Livability: the livability of the neighborhood of the police stations which arrest the defendants
- his_prop: the proportion of history arrest number of different police stations
- senlength_derived: the sentence length of the defendant (unit: year)
- Incar: whether the defendants were incarcerated according to their commitment type

Methods: data cleaning

- **DISPOSITION_CHARGED_CLASS**
  - subset to the felonies type, including Class M, Class X, Class 1, Class 2, Class 3, and Class 4.
  - encode categorical values of prediction outcomes and rename the felonies type as int type from number 1 - 6 (from least severe to most severe) according to the order of severity.

Code for creating 'DISPOSITION_CHARGED_CLASS' variable

```
new_df = new_df[new_df.DISPOSITION_CHARGED_CLASS.\
isin(['1','2','3','4','X','M'])]
species_dict={
     '4': 1,
     '3': 2,
     '2': 3,
     '1': 4,
     'X' : 5,
     'M' :6}
new_df['DISPOSITION_CHARGED_CLASS'] = new_df.
    DISPOSITION_CHARGED_CLASS.map(species_dict)
```

Methods: data cleaning

- **is_changed_offense**
  - create a binary variable according to the offense type and the updated offense type of each defendant

Methods: data cleaning

- **race variables**
  - create binary variables according to the defendants' races

Methods: data cleaning

- **is_defendant_male**
  - create a binary variable according to the defendants' genders

Methods: data cleaning

- **is_judge_male**
  - select the first name of the judges
  - use gender.Detector() package to predict the judges' genders
  - subset to the mostly male, male, mostly female, female
  - create a binary variable according to the gender of the judge

Code for creating 'is_judge_male' variable

```
# Load the packages
!pip install gender_guesser
import gender_guesser.detector as gender
# Get the first name of judgers
sentencing['FN_judge'] = [i.split(' ')[0] for i in
    sentencing['SENTENCE_JUDGE']]
# Predict the judgers' genders
gd = gender.Detector()
sentencing['Gender_judge'] = sentencing['FN_judge'].apply
    (str.capitalize).map(lambda x: gd.get_gender(x))
# Rename the is_judge_gender
sentencing1 = sentencing[sentencing.Gender_judge.isin(['
    female','male','mostly_female','mostly_male'])]
sentencing1['is_judge_male'] = np.where(sentencing1.
    Gender_judge.isin(['mostly_male','male']), True,
    False)
```

Methods: data cleaning

- **is_covid**
  - extract the arrest year from the arrest date column
  - create a binary variable according to the arrest year(compared with 2020)

Code for creating 'is_covid' variable

```
# Use regex to clean up the date columns
sentencing1["ARREST_DATE"]  = [re.sub(r'2[1-9]([0-9]+)',
    r"20\1", str(date))
if bool(re.search('\/2[1-9][0-9]+', str(date)))
else str(date)
for date in sentencing1["ARREST_DATE"]]
# Transfer date to datetime
sentencing1["ARREST_DATE"] = pd.to_datetime(
    sentencing_cleaned1.ARREST_DATE)
# Extract the year
sentencing1["arrest_year"] = sentencing1["ARREST_DATE"].
    dt.year
# Create is_covid variable
sentencing1["is_covid"] = np.where(sentencing1["
    arrest_year"]>=2020,True,False)
```

Methods: data cleaning

- **Livability**
  - search the livability index from website(https://www.areavibes.com)
  - add the livability index with the neighborhood where the police stations are
  - merge the police station data set with the sentencing data set according to the district name

Code for creating 'Livability' variable

```
new_df = sentencing[~sentencing['LAW_ENFORCEMENT_UNIT'].
    isnull()]
new_df_count = new_df.groupby('LAW_ENFORCEMENT_UNIT').agg
    (COUNT = ('CASE_ID','count')).\
sort_values('COUNT',ascending = False).reset_index()
new_df_up = new_df_count[new_df_count.
    LAW_ENFORCEMENT_UNIT.str.contains('District')].\
reset_index().drop(columns=['index'])
new_df_up['DISTRICT NAME'] = [new_df_up.
    LAW_ENFORCEMENT_UNIT.str.split(' - ')[i][1] for i in
    range(len(new_df_up))]
new_df_con = pd.merge(new_df_up,
                      police_stations[['DISTRICT NAME','
    LATITUDE','LONGITUDE','NEIGHBORHOODS']],
                      how = 'inner',
                      on = 'DISTRICT NAME')
```

Code for creating 'Livability' variable

```
new_df_con['Livability'] = [62, 51, 60, 56, 58, 64, 58,
    56, 66, 55, 54, 57, 63, 68, 73, 73, 55, 63, 76, 76,
    67, 69]
sentencing2 = sentencing1[(~sentencing1.
    LAW_ENFORCEMENT_UNIT.isnull())&(~sentencing1.
    arrest_year.isna())]
sentencing_conc = pd.merge(sentencing2,
                           new_df_con[['Livability','
    LAW_ENFORCEMENT_UNIT']],
                           how='inner',
                           on='LAW_ENFORCEMENT_UNIT')
```

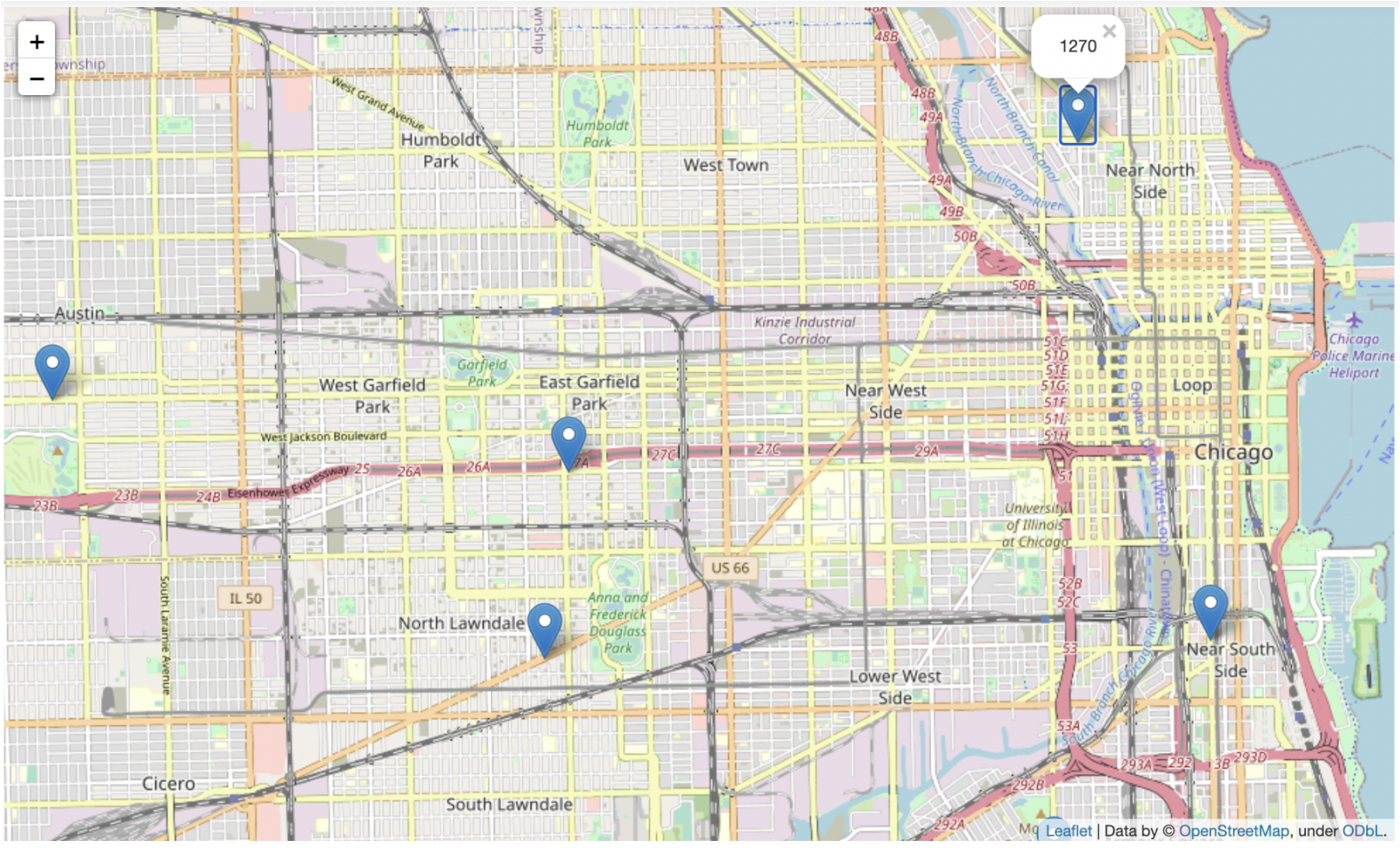
Methods: data cleaning

- **his_prop**
  - get the proportion of different police station history arrest number
  - merge the police station data set with the sentencing data set according to the district name

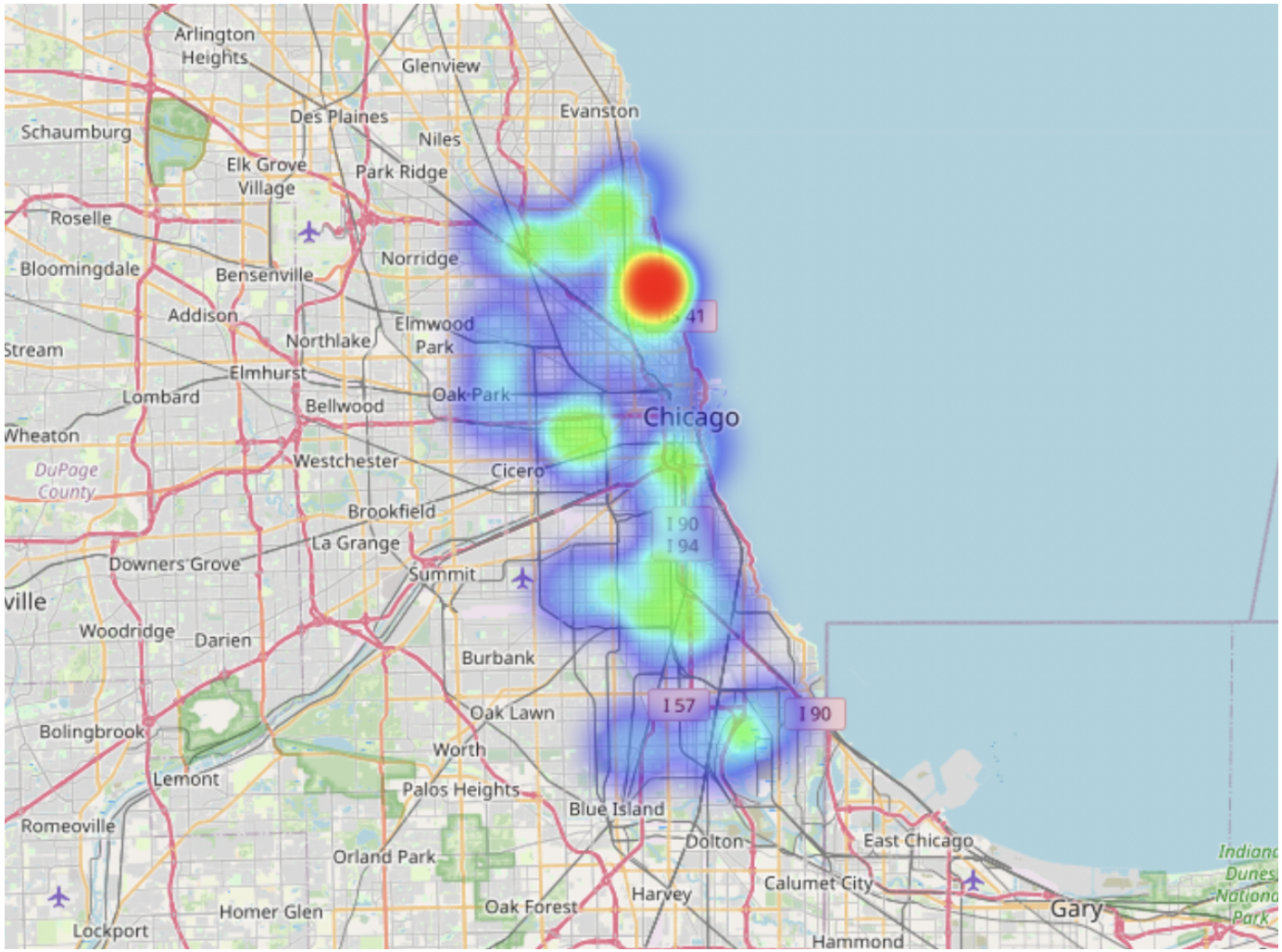
Code for creating 'his_prop' variable

```
new_df_con['his_prop'] = new_df_con.COUNT / new_df_con.
    COUNT.sum()
sentencing_conc = pd.merge(sentencing_conc,
                           new_df_con[['his_prop','
    LAW_ENFORCEMENT_UNIT']],
                           how='inner',
                           on='LAW_ENFORCEMENT_UNIT')
```

Visualize the police station distribution in Chicago according to the number of defendants



Visualize the police station distribution in Chicago according to the number of defendants



Methods: data cleaning

- **senlength_derived**
  - subset the Commitment Unit to certain types and commitment term to non-NA value
  - transform the sentence length of defendants to year(s)

Methods: data cleaning

- **Incar**
  - subset the defendants' COMMITMENT_TYPE == 'Illinois Department of Corrections as incarceration
  - create a binary variable according to the COMMITMENT_TYPE

Describing the analytic process

Predict whether the defendant will be incarcerated or not

- Use 13 independent variables to predict the dependent variable 'Incar'.
- Choose a logistic regression model to get the relationship between independent variables and dependent variable.
- Choose four models (Logistic Regression, K-Nearest Neighbor, Decision Tree, and Random Forest) to predict, and compare the prediction results to understand the difference between models

Introduction of different models

Linear Regression Model: The linear regression model is often fitted using the least squares approach and used linear predictor functions whose unknown model parameters are estimated from the data.

Logistic Regression Model: The parameters of a logistic regression are most commonly estimated by maximum-likelihood estimation (MLE) and logistic regression models are often used to predict the binary variables.

KNN Model: The k-nearest neighbors algorithm, also known as KNN, is a non-parametric, supervised learning classifier, which uses proximity to make classifications or predictions about the grouping of an individual data point.

Introduction of different models

Decision Tree Model: The decision tree model is the model of computation in which an algorithm is considered to be basically a decision tree, i.e., a sequence of queries or tests that are done adaptively, so the outcome of the previous tests can influence the test is performed next.

Random Forest Model: The random forest model is an ensemble learning method for classification, regression and other tasks that operates by constructing a multitude of decision trees at training time. For classification tasks, the output of the random forest is the class selected by most trees. Random decision forests correct for decision trees' habit of overfitting to their training set.

Select the train and test data sets

Predict whether the defendant will be incarcerated or not

- We use train test split(X,y,random state=6) to split the train and test data sets
- The shape of train and test data sets
  - The Training Features Shape is (35471, 14)
  - The Training Labels Shape is (35471,)
  - The Testing Features Shape is (11824, 14)
  - The Testing Labels Shape is (11824,)

View of regression results

Suppose the logistic regression coefficient confidence interval default to 5 percent

variable	coef	std err	t-value	p-values	coef-0.025	coef-0.975
charge_cou	-0.0377	0.016	-2.310	0.021	-0.070	-0.006
class	-0.1790	0.011	-16.472	0.000	-0.200	-0.158
age	0.0125	0.001	15.089	0.000	0.011	0.014
changed_off	-0.1999	0.052	-3.878	0.000	-0.301	-0.099
black	-2.0940	0.138	-15.191	0.000	-2.364	-1.824
hisp	-2.8473	0.251	-11.352	0.000	-3.339	-2.356
other	-2.8930	0.145	-19.993	0.000	-3.177	-2.609
white	-2.9038	0.150	-19.417	0.000	-3.197	-2.611
de_male	1.0384	0.033	31.435	0.000	0.974	1.103
ju_male	-0.0889	0.022	-3.986	0.000	-0.133	-0.045
covid	-0.7203	0.208	-3.466	0.001	-1.128	-0.313
livability	0.0082	0.002	4.270	0.000	0.004	0.012
his_prop	0.7361	0.110	6.666	0.000	0.520	0.953
senlength	0.5030	0.010	48.287	0.000	0.483	0.523

View of prediction results

Actual	LR	KNN	DT	RF	DT_R	RF_R
0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0
1	1	1	1	1	1	1
1	1	1	1	1	1	1
1	1	1	1	1	1	1
1	1	1	1	1	1	1
1	1	1	1	1	1	1
0	0	0	0	0	0	0

Prediction accuracy of different models

Model Type	F1-Score	Precision	Recall
Logistic Regression	0.61	0.61	0.62
KNN Classifier	0.80	0.80	0.80
DecisionTreeClassifier	0.85	0.85	0.85
Random Forest Classifier	0.86	0.86	0.86
Decision Tree Classifier Refined	0.85	0.85	0.85
Random Forest Classifier Refined	0.86	0.86	0.86

$$F_1 = (\frac{recall^{-1} + precision^{-1}}{2})^{-1} = 2 * \frac{precision * recall}{precision + recall}$$

Prediction accuracy of logistic regression model

Actual Class	Predicted Class		Class Accuracy
	Not Incarceration	Incarceration	
Not Incarceration	2747	2580	0.55
Incarceration	1951	4546	0.67

Prediction accuracy of KNN model

Actual Class	Predicted Class		Class Accuracy
	Not Incarceration	Incarceration	
Not Incarceration	4427	900	0.79
Incarceration	1520	4977	0.80

Prediction accuracy of decision tree model

Actual Class	Predicted Class		Class Accuracy
	Not Incarceration	Incarceration	
Not Incarceration	4560	767	0.83
Incarceration	1051	5446	0.86

Prediction accuracy of random forest model

Actual Class	Predicted Class		Class Accuracy
	Not Incarceration	Incarceration	
Not Incarceration	4795	532	0.85
Incarceration	1182	5315	0.86

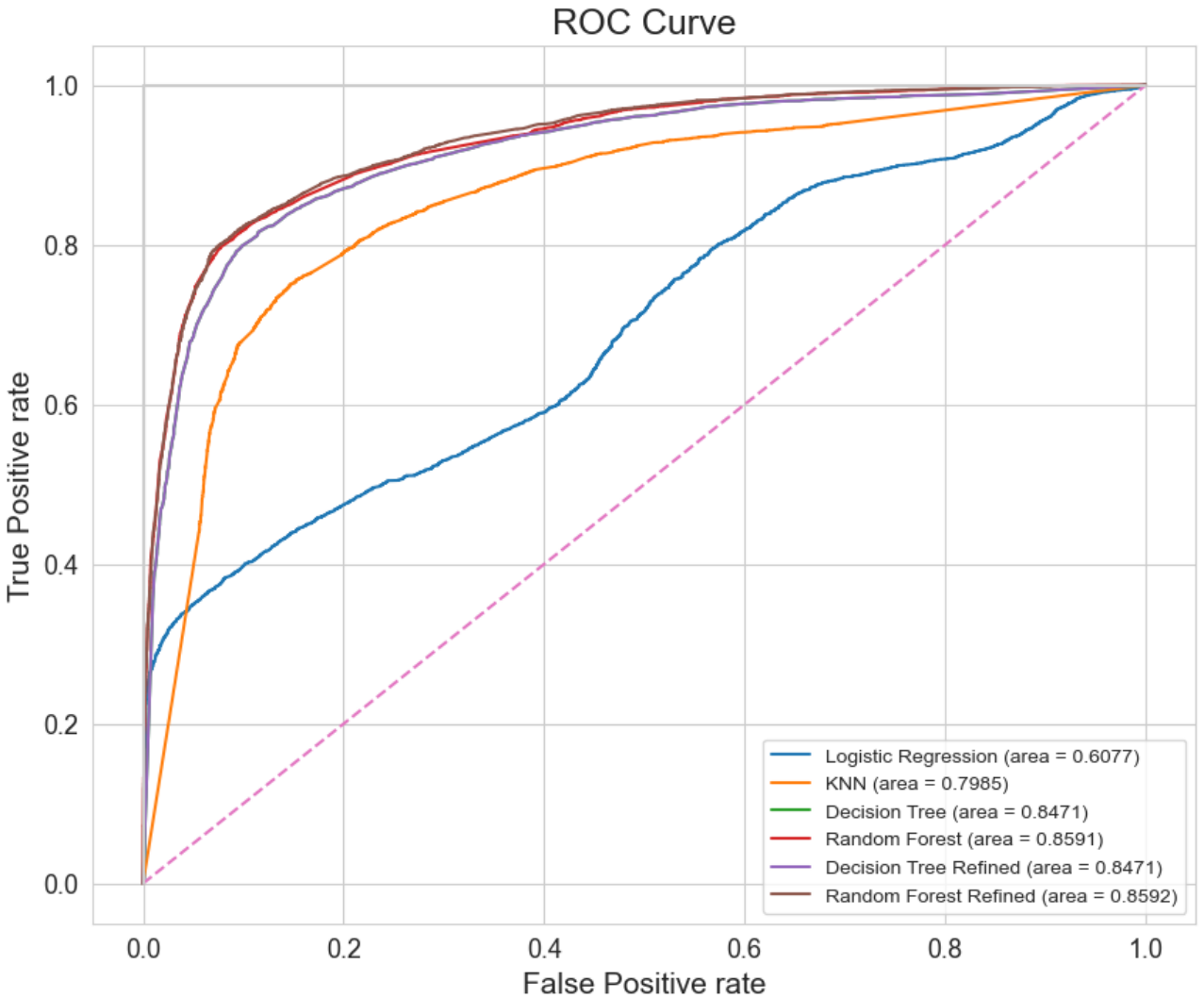
Prediction accuracy of decision tree(refined) model

Actual Class	Predicted Class		Class Accuracy
	Not Incarceration	Incarceration	
Not Incarceration	4560	767	0.83
Incarceration	1051	5446	0.86

Prediction accuracy of random forest(refined) model

Actual Class	Predicted Class		Class Accuracy
	Not Incarceration	Incarceration	
Not Incarceration	4732	595	0.85
Incarceration	1067	5430	0.87

Results: ROC Curve area(accuracy of prediction)

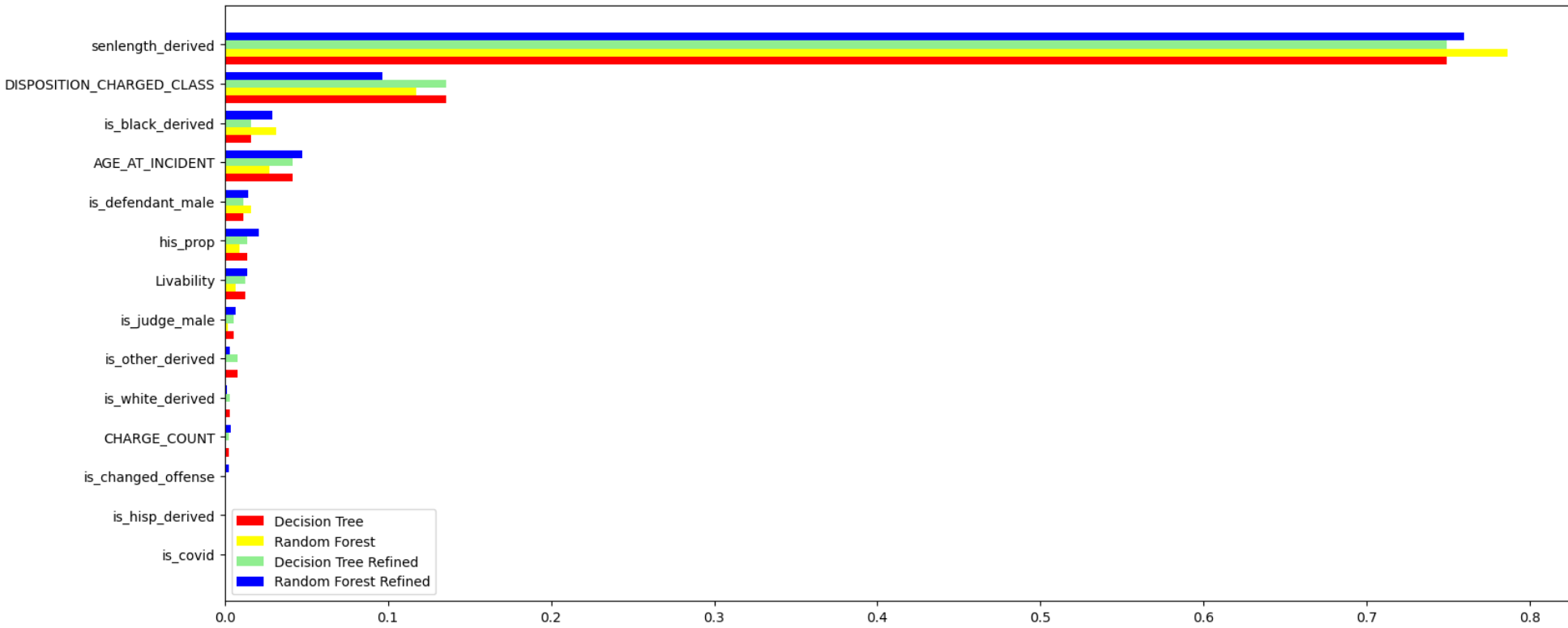


ROC Curve areas of different models

Model Type	ROC Curve Area
Random Forest Classifier Refined	0.8592
Random Forest Classifier	0.8591
Decision Tree Classifier Refined	0.8471
Decision Tree Classifier	0.8471
KNN Classifier	0.7985
Logistic Regression	0.6077

Following are a few thumb rules:
0.90-1 = excellent (A)
0.80-0.90 = good (B)
0.70-0.80 = fair (C)
0.60-0.70 = poor (D)
0.50-0.60 = fail (F)

Results: Feature Importance Comparison



Describing the analytic process

Predict the defendant's sentence length

- Use 13 independent variables to predict the dependent variable 'senlength_derived'.
- Choose a linear regression model to get the relationship between the independent variables and the dependent variable.
- Choose four models (Logistic Regression, K-Nearest Neighbor, Decision Tree, and Random Forest) to predict and compare the prediction results of these four models.

The relationship between independent variables and dependent variable

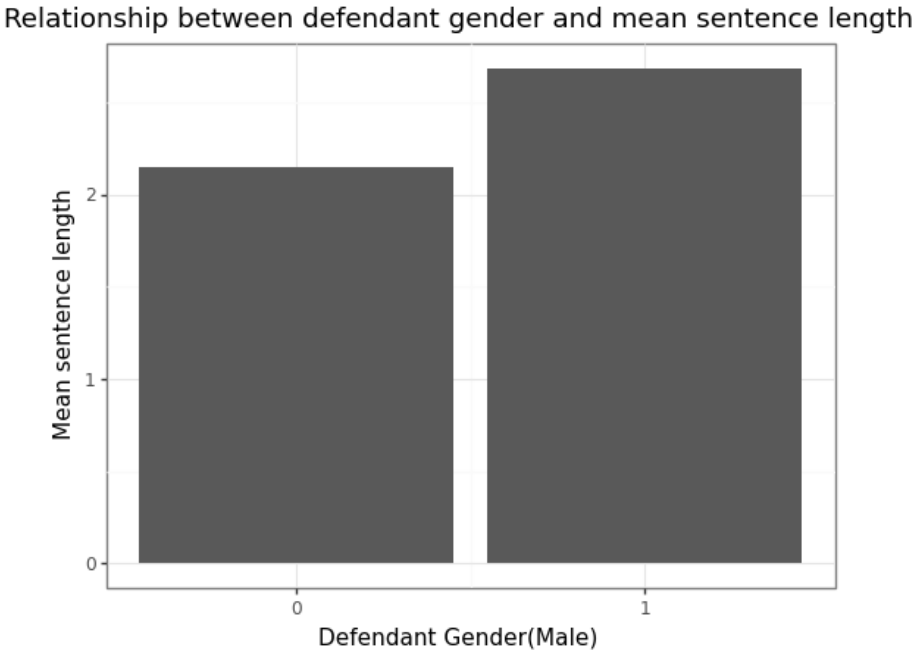


Abbildung 1: Defendant Gender

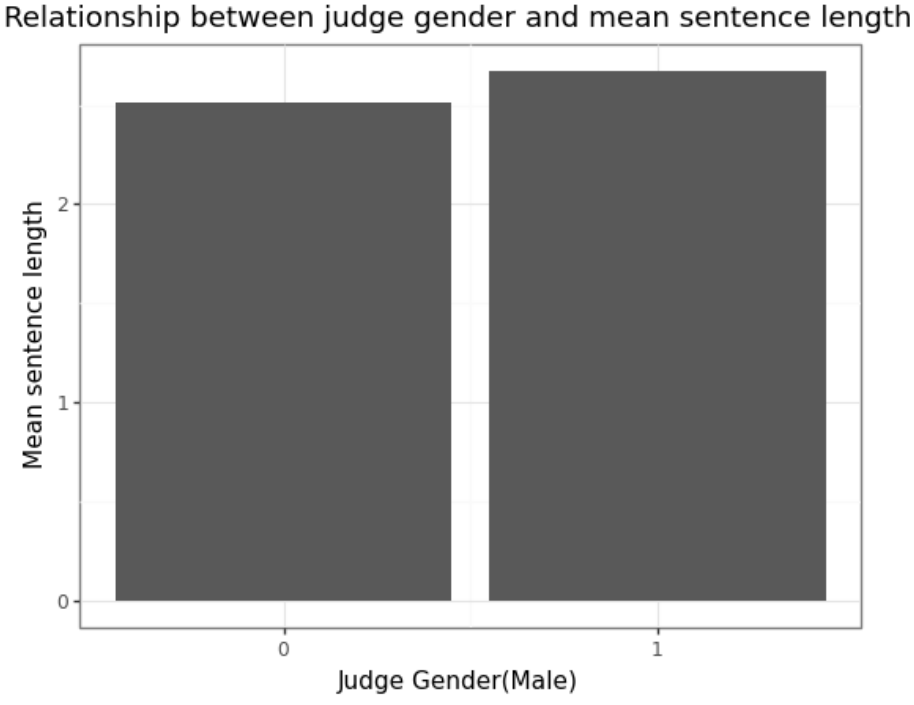


Abbildung 2: Judge Gender

The relationship between independent variables and dependent variable

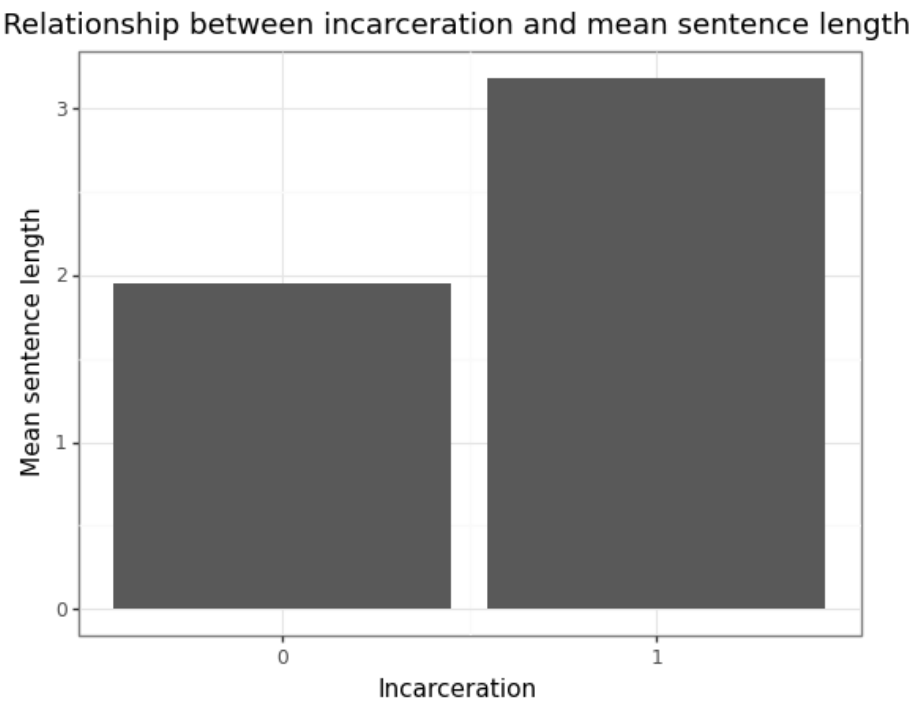


Abbildung 3: Incarceration

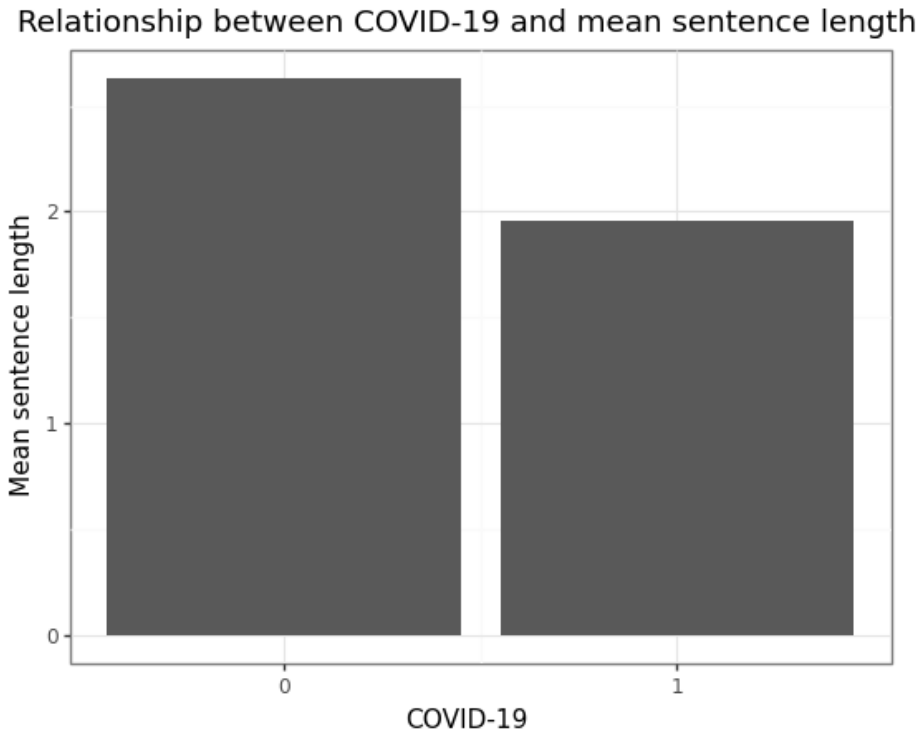


Abbildung 4: COVID-19

View of regression results

Suppose the linear regression coefficient confidence interval default to 5 percent

variable	coef	std err	t-value	p-values	coef-0.025	coef-0.975
const	-0.016224	0.141946	-0.114	0.909	-0.294441	0.261993
charge_cou	0.004093	0.019722	0.208	0.836	-0.034563	0.042748
age	0.007129	0.000987	7.223	0.000	0.005195	0.009063
changed_off	-0.320902	0.058826	-5.455	0.000	-0.436201	-0.205602
black	-0.043763	0.058491	-0.748	0.454	-0.158407	0.070881
hisp	-0.041964	0.196648	-0.213	0.831	-0.427397	0.343470
other	0.053643	0.064778	0.828	0.408	-0.073322	0.180608
white	0.015860	0.070702	0.224	0.823	-0.122718	0.154437
de_male	0.139637	0.038630	3.615	0.000	0.063922	0.215351
ju_male	0.094098	0.026645	3.532	0.000	0.041874	0.146322
covid	-0.211914	0.245951	-0.862	0.389	-0.693981	0.270152
livability	-0.001130	0.002277	-0.496	0.620	-0.005592	0.003333
his_prop	-0.593837	0.132144	-4.494	0.000	-0.852841	-0.334833
incar	0.902852	0.024662	36.609	0.000	0.854514	0.951190
class	1.182286	0.009963	118.666	0.000	1.162758	1.201814

R-squared: 0.270

Adjusted R-squared: 0.270

View of prediction results

Actual	Ridge	Lasso	Elastic	KNN	DT	RF
1.000000	1.964501	2.161997	2.066604	1.357143	1.441176	1.727166
1.500000	2.107172	2.161997	2.066604	2.214286	1.991935	1.729837
1.000000	1.954694	2.161997	2.066604	2.559524	2.012821	1.729837
1.000000	1.877743	2.161997	2.066604	1.714286	1.659910	1.729837
2.000000	3.484370	3.262895	3.393662	2.000000	1.905405	2.045453
2.500000	1.739701	2.161997	2.066604	1.571429	1.393045	1.882024
2.000000	1.510122	2.161997	2.066604	1.928571	1.794118	1.882024
2.000000	2.091617	2.161997	2.066604	1.928571	1.363636	1.727166
0.378082	1.327317	2.161997	2.066604	2.000000	1.985714	1.884695
1.000000	2.063364	2.161997	2.066604	1.642857	1.843750	1.728067

Test Score of each model

Model Type	Train Score	Test Score
Random Forest Regression	59.92	55.04
Decision Tree Regression	64.83	51.19
KNN Regression	54.46	35.76
ElasticNet Regression	-109.94	-110.12
Lasso Regression	-114.07	-114.35
Ridge Regression	-115.40	-115.92

Comparison of different models

Model Type	Tr. RMSE	Tr. R-Squared	Te. RMSE	Te. R-Squared
Ridge Regression	2.564153	0.262252	2.693230	0.243813
Lasso Regression	2.713009	0.174109	2.835030	0.162090
ElasticNet Regression	2.674562	0.197351	2.798033	0.183817
KNN Regression	2.014643	0.544575	2.482347	0.357598
Decision Tree Regression	1.770321	0.648338	2.163875	0.511857
Random Forest Regression	1.889937	0.599211	2.076634	0.550425

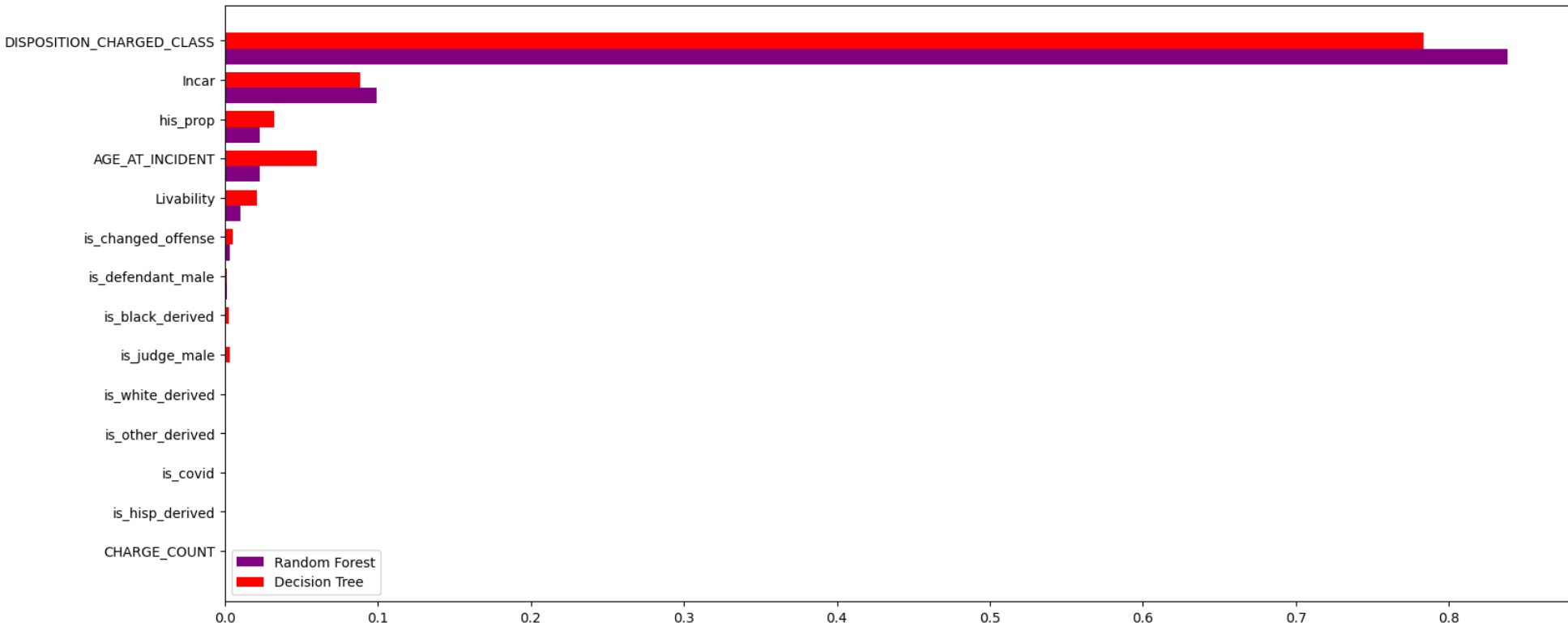
When we have more samples, reconstructing the error distribution using Root Mean Squared Error (RMSE) is considered to be more reliable.

As compared to mean absolute error, RMSE gives higher weightage and punishes large errors.

RMSE metric is given by:

$$RMSE = \sqrt{\frac{\sum_{i=1}^{N}(Predicted_i - Actual_i)^2}{N}}$$

The comparison of feature importances of Random Forest and Decision Tree



Describing the analytic process

Prediction of the deposition charged class of defendants

- Our prediction results will map to the following classes of felonies in DESPOSITION_CHARGED_CLASS:
  - 1. Class M felonies (First-degree murder)
  - 2. Class X felonies
  - 3. Class 1 felonies
  - 4. Class 2 felonies
  - 5. Class 3 felonies
  - 6. Class 4 felonies
- Possible factors: Gender, race, covid effect, living condition, local crime rate, whether the offense category has changed, whether to be incarcerated, sentence length, of charges of one defendant, Age of defendant at the date of the incident.

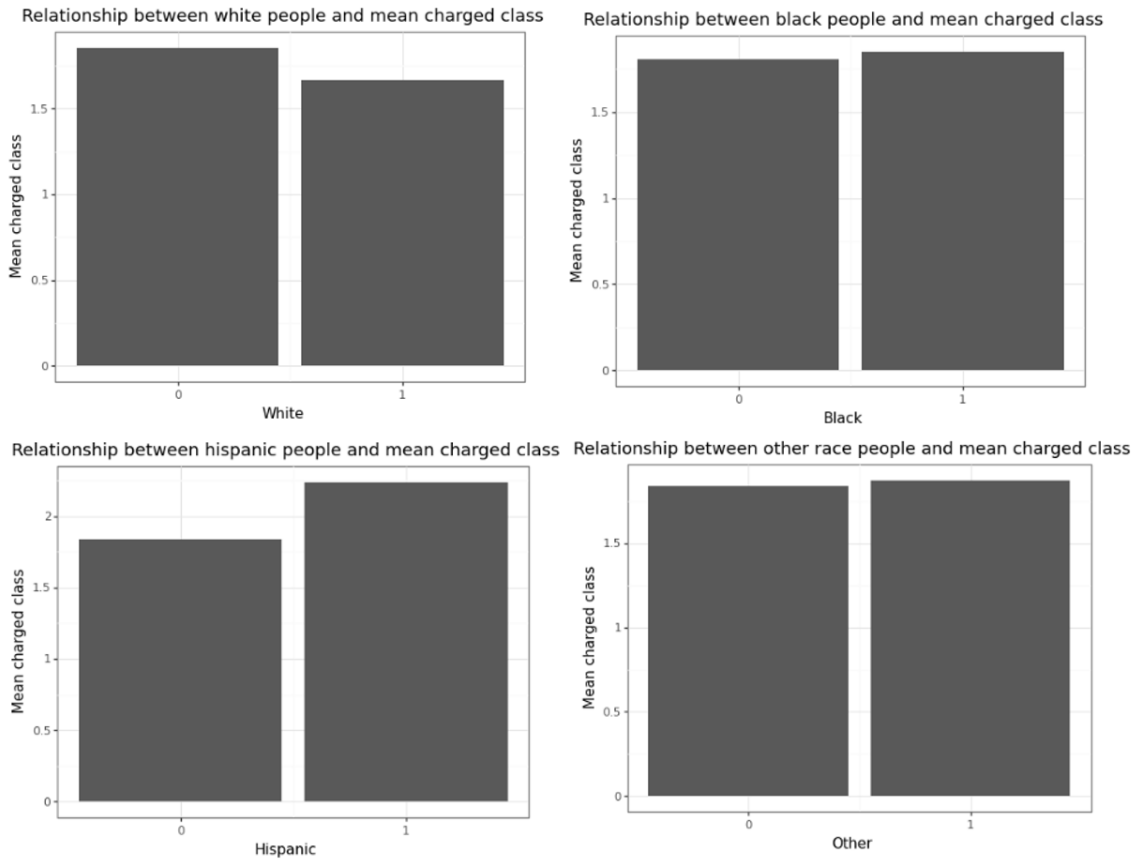
Describing the analytic process

Prediction of the deposition charged class of defendants

- Including variables: “is_black_derived”, “is_hip_derived”, “is_white_derived”, “is_other_derived”, *predicted “is_judge_male” and “is_defendant_male”, “his_prop”, “is_changed_offense”, “is_covid”, “Livability”, “senlength_derived”, “CHARGE_COUNT”, “AGE_AT_INCIDENT”.

The relationship between a single variable and the predicted outcome

- Defendant's Race
- Hispanic defendants are more likely to be classified as higher felonies than other ethnic groups.
- White defendants are more likely to be classified as lower felonies than defendants of other races.



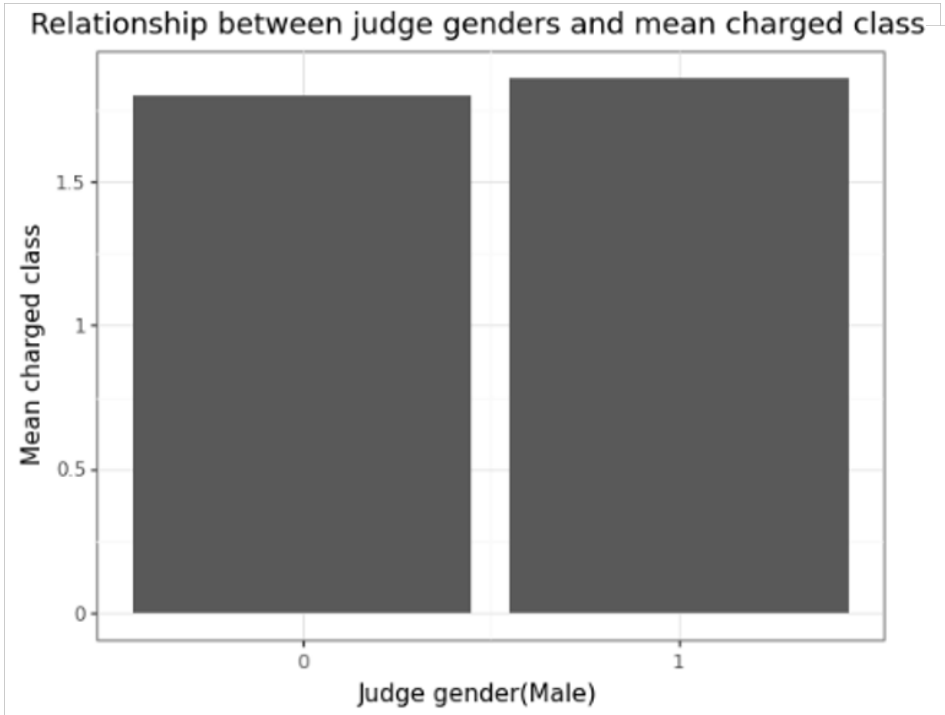


Abbildung 5: Gender of judge vs. mean charged class

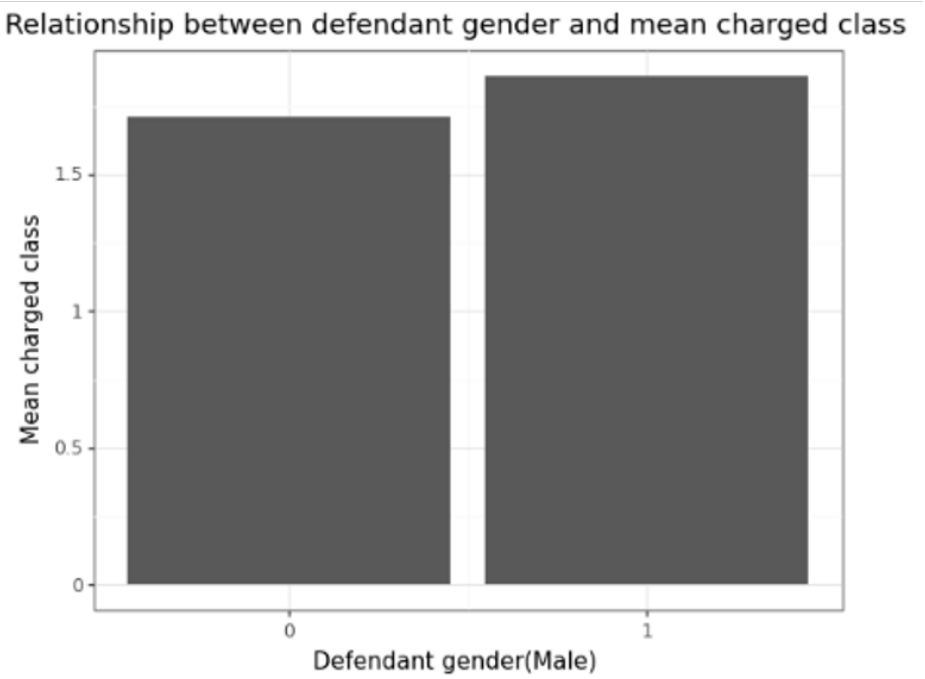
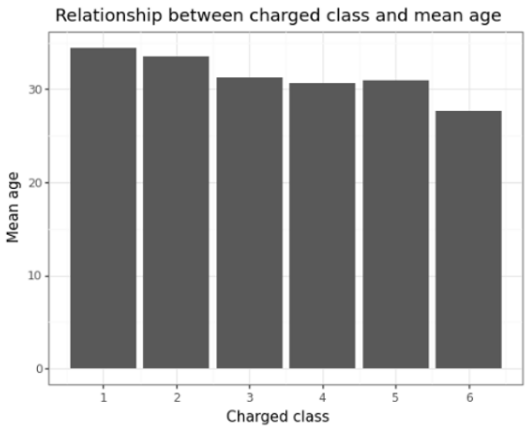
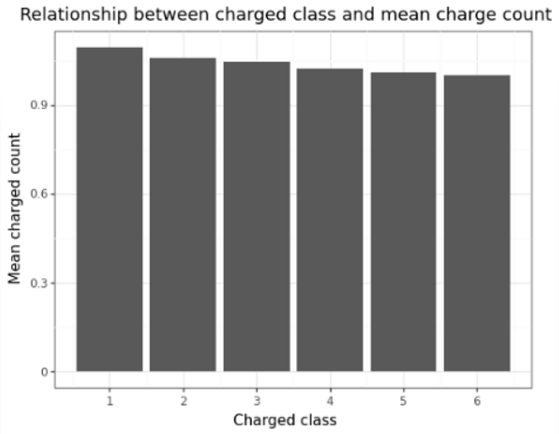
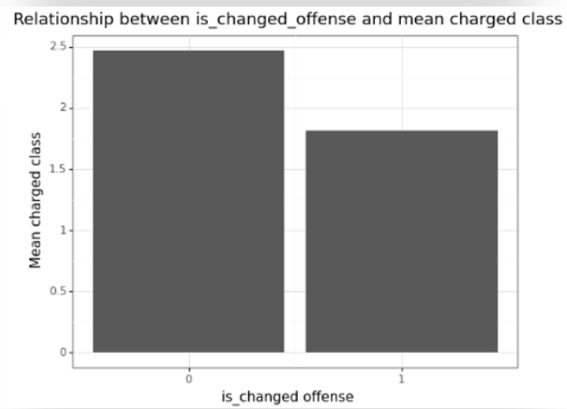


Abbildung 6: Gender of defendant vs. mean charged class





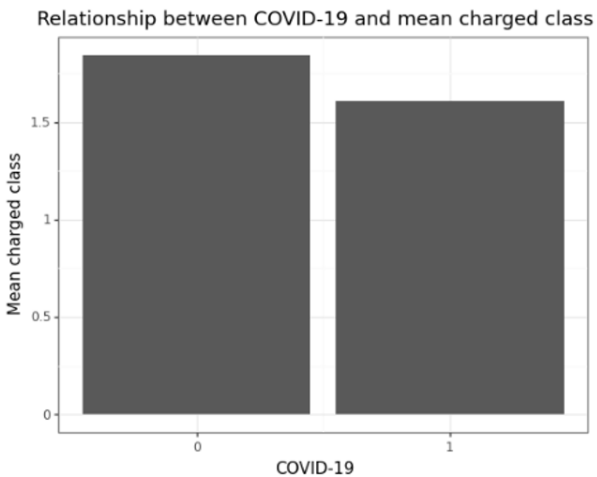


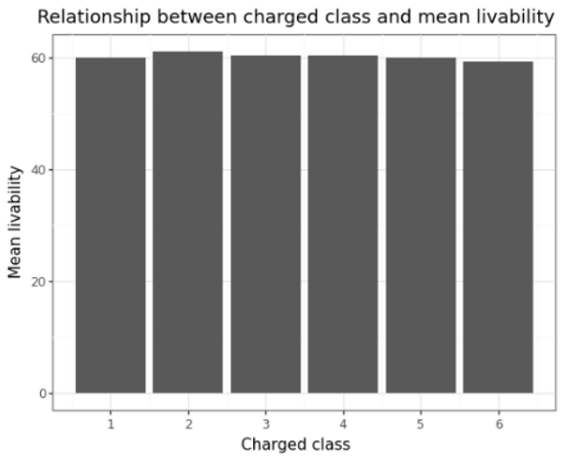
Age at incident, charge count, charge update

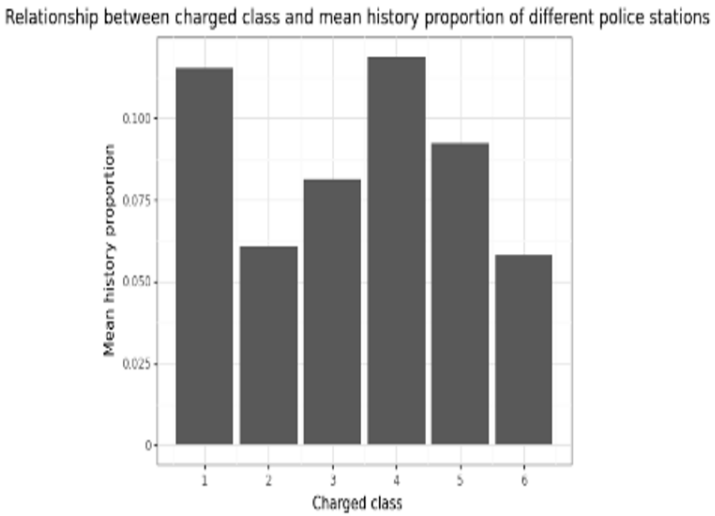
- Negative relationship between age at incident and class of offense
- The more charges, the lower the criminal offense level.
- The change of offense category might lead to a lower crime class level.

Covid, livability, historical crime rate

- Post-pandemic felonies were classified as less severe than pre-pandemic felonies.
- There is no relationship between livability and charged class
- High: Class 4 and Class 1; Low: Class 3 and Class M.

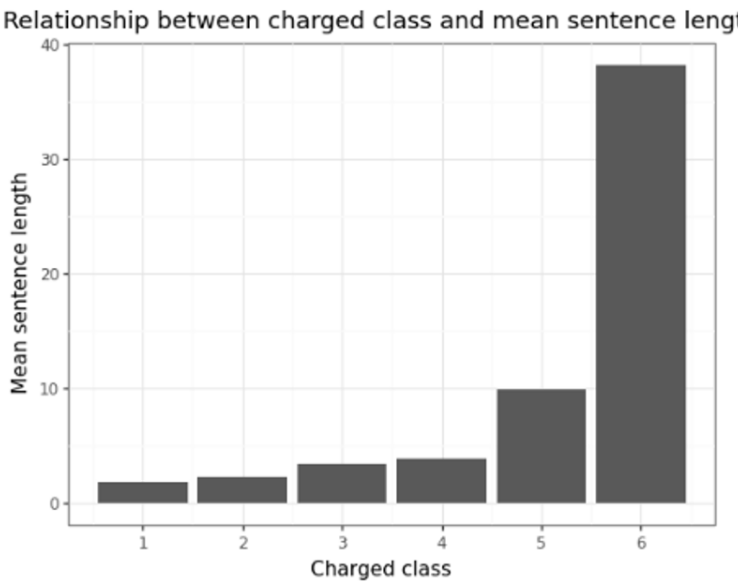


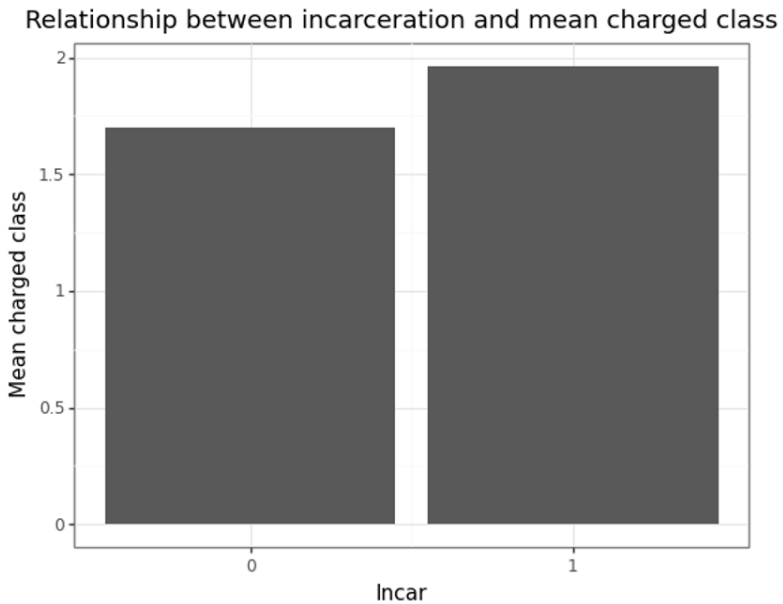




Sentence length, incarceration

Obviously, the length of the sentence is directly proportional to the class of felonies.





View of Regression Results

We can see is_defendant_male, is_covid have no relevance with y variable,
is_judge_male has less relevance with y variable.
There are multilinearlity problems.

	params	std err	t	p-values	coef_0.025	coef_0.975
const	1.547381	0.057072	27.113	0.000	1.435518	1.659243
CHARGE_COUNT	-0.069650	0.007985	-8.723	0.000	-0.085300	-0.054000
AGE_AT_INCIDENT	-0.011211	0.000397	-28.255	0.000	-0.011989	-0.010433
is_changed_offense	-0.380875	0.023778	-16.018	0.000	-0.427481	-0.334269
is_black_derived	0.385768	0.023633	16.323	0.000	0.339447	0.432090
is_hisp_derived	0.599746	0.079631	7.532	0.000	0.443669	0.755824
is_other_derived	0.334120	0.026202	12.752	0.000	0.282763	0.385476
is_white_derived	0.227747	0.028628	7.955	0.000	0.171635	0.283858
is_defendant_male	-0.011620	0.015654	-0.742	0.458	-0.042302	0.019063
is_judge_male	0.022584	0.010797	2.092	0.036	0.001422	0.043746
is_covid	-0.072404	0.099655	-0.727	0.468	-0.267730	0.122921
Livability	0.004656	0.000922	5.048	0.000	0.002848	0.006463
his_prop	-0.724473	0.053450	-13.554	0.000	-0.829236	-0.619710
senlength_derived	0.194100	0.001636	118.666	0.000	0.190894	0.197306
Incar	0.034603	0.010132	3.415	0.001	0.014744	0.054462

```
## Get the accuracy of test data set
log_reg.score(X_test,y_test)
```

```
0.6627198917456022
```

Abbildung 7: Logistic Regression Model

```
## The score of test model
knn_clf.score(X_test,y_test)
```

```
0.6932510148849798
```

Abbildung 8: KNN Model

- Left:0.713
- Right is the refined random forest model that has a slightly higher accuracy (0.715)

Random Forest Model

```
param_grid = {
    'criterion':['entropy','gini'],
    'max_depth':[5, 6, 7, 8],
    'n_estimators':[11,13,15],
    'max_features':[0.3,0.4,0.5],
    'min_samples_split':[4,8,12,16]
}

import sklearn.ensemble as ensemble
from sklearn.ensemble import RandomForestClassifier
rf_clf = ensemble.RandomForestClassifier()
grid_search = GridSearchCV(rf_clf,param_grid)
grid_search.fit(X_train, y_train)
```

```
param_grid = {
    'criterion':['entropy','gini'],
    'max_depth':[8,9,10,11,12,13],
    'n_estimators':[15,17,19,21,23],
    'max_features':[0.5,0.6,0.7,0.8],
    'min_samples_split':[1,2,3,4]
}

import sklearn.ensemble as ensemble
from sklearn.ensemble import RandomForestClassifier
rf_clf1 = ensemble.RandomForestClassifier()
grid_search = GridSearchCV(rf_clf1,param_grid)
grid_search.fit(X_train, y_train)
```

- Left:0.713
- Right is the refined random forest model that has a slightly higher accuracy (0.715)

Decision Tree Model

- Left: 0.672
- Right is the refined random forest model that has a higher accuracy (0.672), random_state = 42

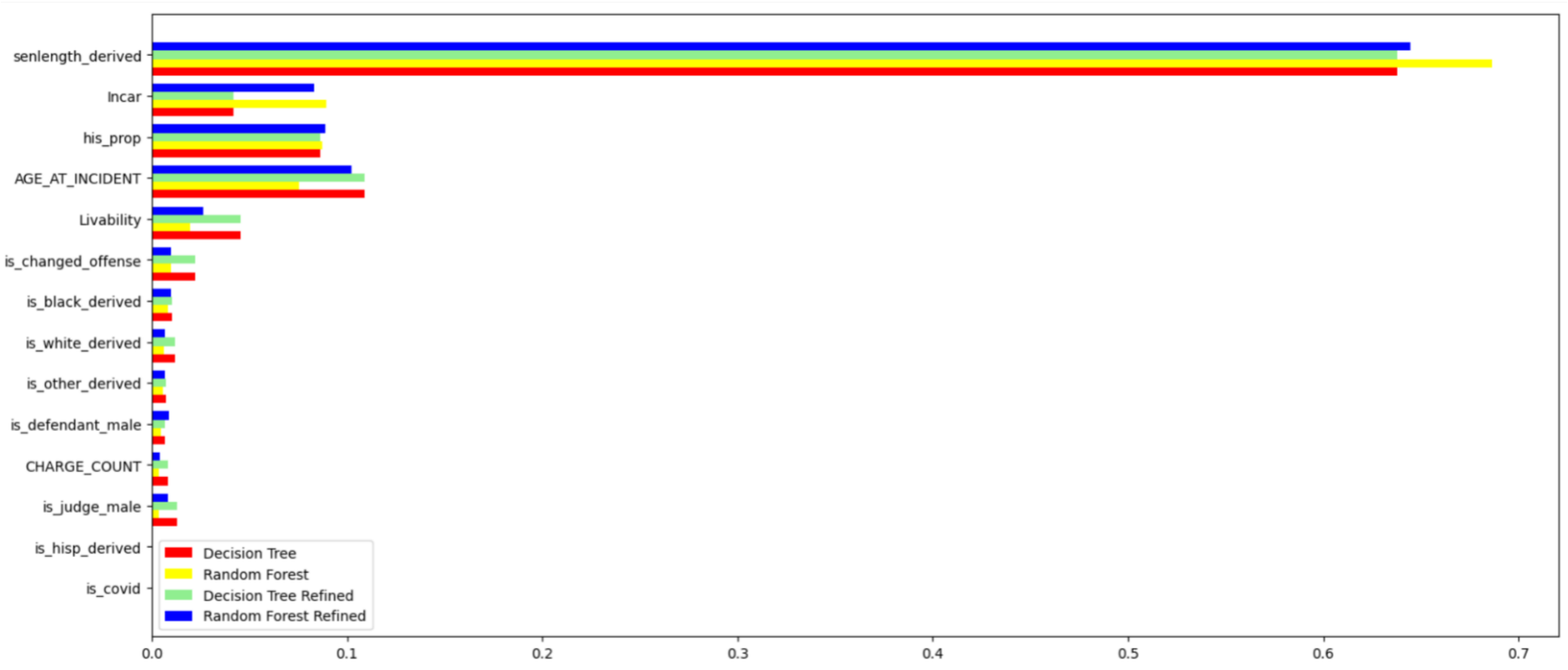
```
from sklearn.tree import DecisionTreeClassifier
dt_clf= DecisionTreeClassifier(random_state=6)
```

```
from sklearn.model_selection import GridSearchCV
param_grid = [
    {
        'max_features':['auto','sqrt','log2'],
        'min_samples_split':[2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18],
        'min_samples_leaf':[1,2,3,4,5,6,7,8,9,10,11]
    }
]
grid_search = GridSearchCV(dt_clf,param_grid)

grid_search.fit(X_train,y_train)
```

```
from sklearn.model_selection import GridSearchCV
param_grid = [
    {
        'max_features':['auto','sqrt','log2'],
        'min_samples_split':[1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20],
        'min_samples_leaf':[1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20]
    }
]
grid_search = GridSearchCV(dt_clf,param_grid)

grid_search.fit(X_train,y_train)
```

The accuracy of different models

Model Type	Score
Random Forest Model Refined	0.715325
Random Forest Model	0.712618
KNN Model	0.693251
Decision Tree Model	0.671685
Decision Tree Model Refined	0.671685
Logistic Regression Model	0.662720


senlength_derived
Incar
his_prop
AGE_AT_INCIDENT
Livability
is_changed_offense
is_black_derived
is_white_derived
is_other_derived
is_defendant_male
CHARGE_COUNT
is_judge_male
is_hisp_derived
is_covid
Decision Tree
Random Forest
Decision Tree Refined
Random Forest Refined
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7

The limitation of the prediction model

The big difference in the data set division

Charged type	Amount
4	29578
2	7250
1	5110
3	3845
X	1430
M	82

Limitations

- Missing data of existing dataset: about 60% of the law enforcement unit data is missing, which is one of the most important factors that would affect the incarceration condition, the charged type, and the sentence length.
- Other necessary related data: historical crime records, location of crimes
- Encoding categorical values issues: offense categories
- The size of the sample: Larger and more sophisticated data sets are needed to make predictions.

Next steps

- Modify the parameters of the model to make the features better participate in the prediction, and find a more suitable machine learning model.
- Screen the significance of variables before the prediction and eliminate the variables with low correlation.
- Consider the interaction effects of variables.