PreCog Recruitment Task

S Sreyas 2021111016 CSD

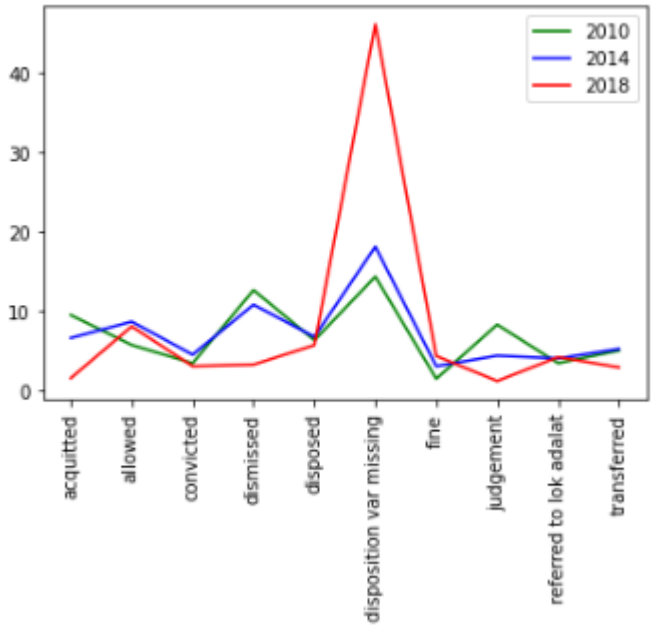
Introduction

Data about 80 million cases from different district courts across India has been analysed in this project. This database is a real-life database containing information about the cases, acts, sections, judges and several courts across India over the years 2010-2018.

Key Insights

Some key trends and insights I have retrieved from the data are as follows-

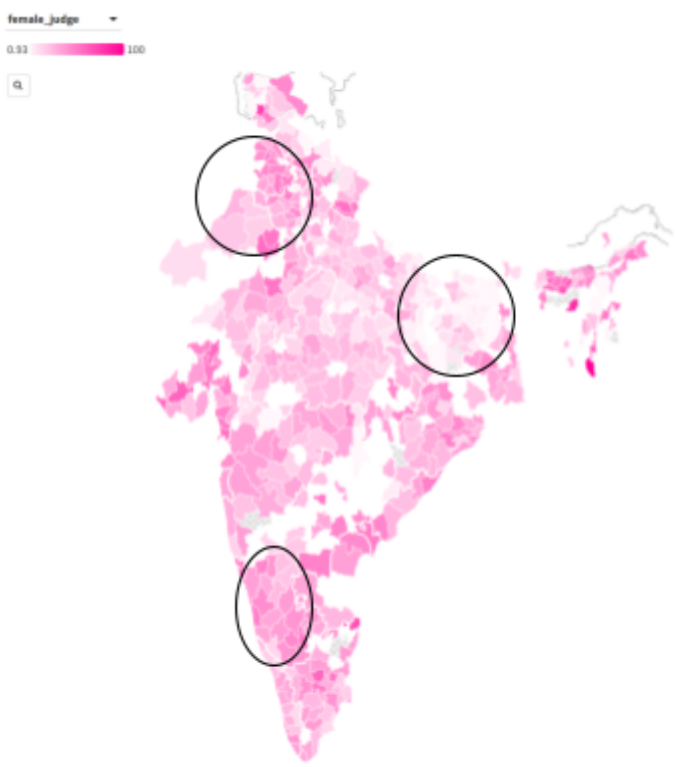
1. Common disposition trends -



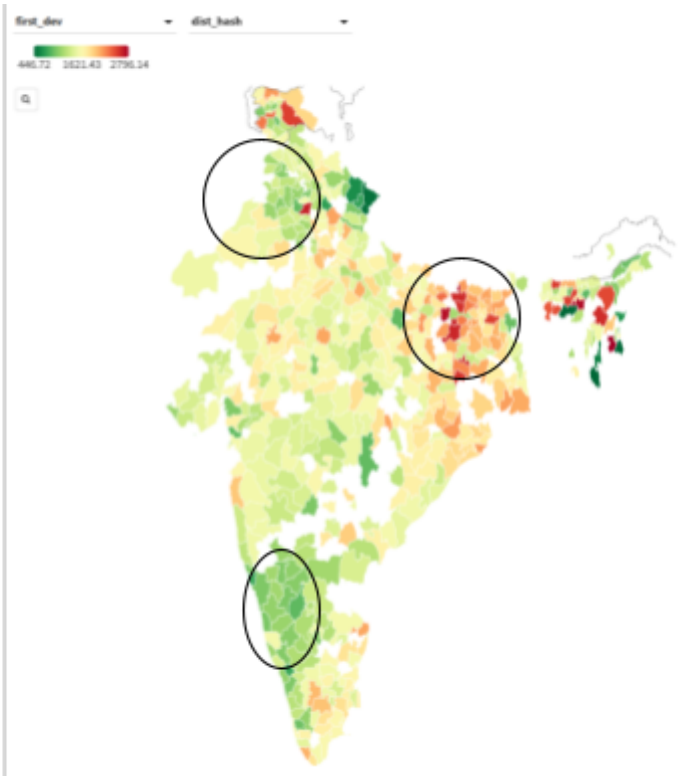
- From this graph, we can see a very consistent trend of increase in disposition “var missing” and a fall in the share of other dispositions.

- Going by this data, we can predict that **at least 50% of dispositions in 2020 would be "var missing"**
- The share of dispositions that fine the guilty has also increased. This partly explains the increase in court earnings by fine by about 40%. (Along with the number of dispositions given)

2. Women representation vs Case process time



Number of female judges per 100 judges in every district

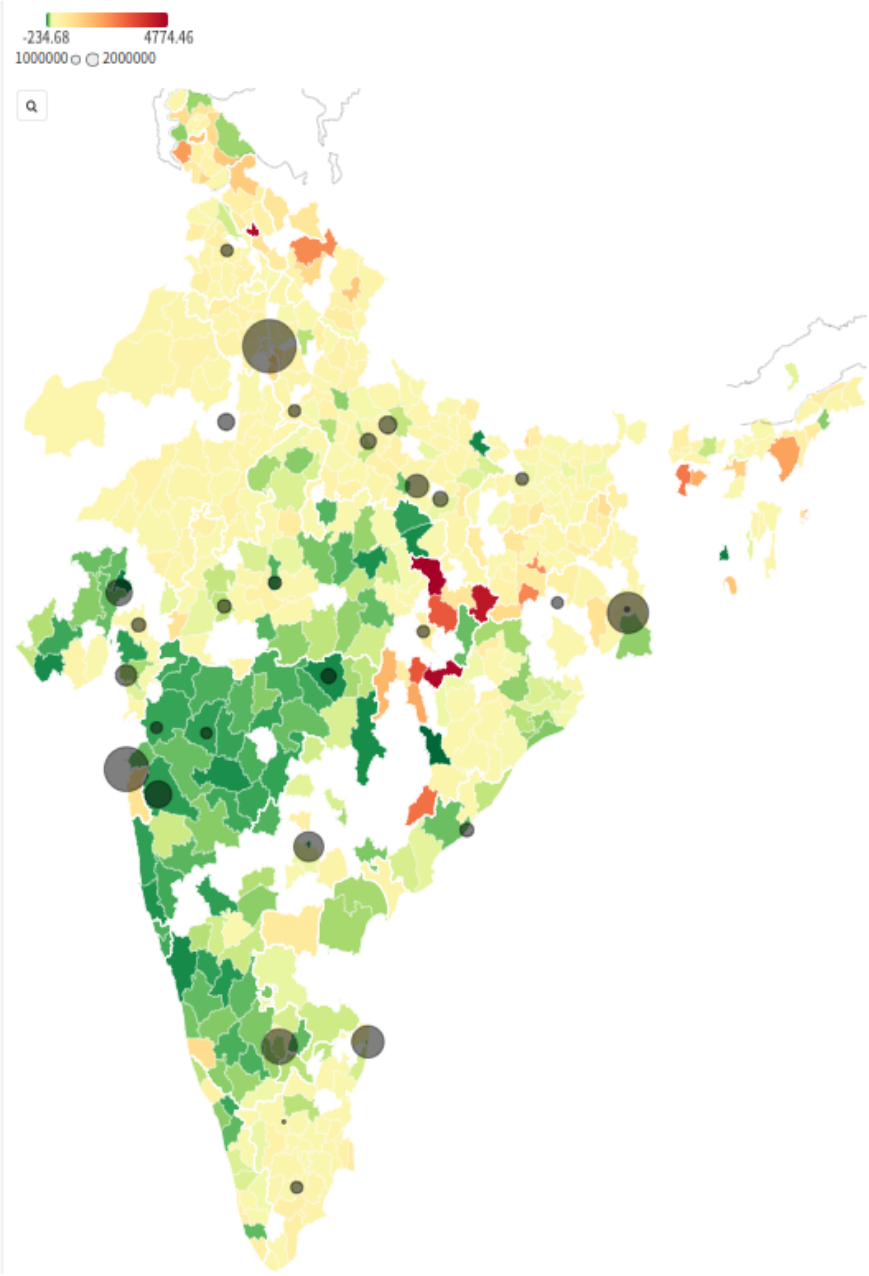


Districtwise expected duration from filling to final verdict (days)

- The map on the right is the **expected duration from filing to final verdict with 80% confidence** for each district
- From both graphs, we can see that increase in female representation results in faster processing of the cases
- This can be observed by looking at the highlighted areas.
    - In Karnataka and north-western Andhra (this dataset hardly has any data for Telangana), we see the ratio of female judges is comparatively high. We can also see that the expected duration of the cases is low.

- In areas of Bihar and Jharkhand, female representation is deficient. This can also be clearly observed from the high processing time for this area.
- Even within Punjab, districts with higher female representation process the cases faster than other districts.

3. Districtwise variation in cases filed in 2010 and 2018



- The map is a rough plot of the variation in the number of cases filed in 2010 and 2018 district-wise.
- The circles in the map are a reference to the population of the region.
- From this, we can infer that areas with high populations decrease the cases filed much better than other areas.
- This is also helped by the fact that the crime rates in urban areas are reducing.
- Some adjustment has been performed to the data to ensure that the data collection issue is resolved.
- There are much less data about 2010 cases compared to 2018 cases. To balance the data collection issues, a factor of 3.17 is multiplied with the number of district-wise cases in 2010.

Classification Model

1. Classification of the criminality of a case

- Below is a model trained to classify whether the case is criminal or not based on the acts, sections, number of sections and bailability of a case.
- This model has been trained on the entire dataset (and does **not** overfit) to predict if a case is criminal or not.
- **This model has an accuracy of 89.07%** (tested on an unbiased dataset) with a roughly equal share of false negatives and false positives.
- The data split used to train the model is **94 : 3 : 3 (train: dev: test)**
  This is because the dataset is significant, therefore, 3% of the data is enough for testing and development.
- Some level of hyperparameter testing has been done to decide the model to use and the number of units in each layer, the number of layers and the non-linearity function for each layer.
- RELU has been used for all layers except the final layer. The last layer uses Sigmoid (to ensure the value is between 0 and 1)

Using the saved model and testing

(apart from the testing done above)

[8]:

```
model = load_model('../input/model-asbc/model_criminal_ActsSections.h5')

df = pd.read_csv('../input/smalldataset/acts_sections1000000.csv')


# Preprocess the data
df['bailable_ipc'] = df["bailable_ipc"].fillna('3')
df = df[['section', 'act', 'bailable_ipc','criminal']]
df = df.dropna()  # Drop rows with missing values
X = df[['section', 'act', 'bailable_ipc']]  # Select features
y = df['criminal']  # Select target

le = LabelEncoder()
X['section'] = le.fit_transform(X['section'])
X['act'] = le.fit_transform(X['act'])
X['bailable_ipc'] = le.fit_transform(X['bailable_ipc'])

scores = model.evaluate(X, y)
print("\n%s: %.2f%%" % (model.metrics_names[1], scores[1]*100))
```

```
/opt/conda/lib/python3.7/site-packages/ipykernel_launcher.py:14: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy

/opt/conda/lib/python3.7/site-packages/ipykernel_launcher.py:15: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  from ipykernel import kernelapp as app
/opt/conda/lib/python3.7/site-packages/ipykernel_launcher.py:16: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
29279/29279 [==============================] - 61s 2ms/step - loss: 3.5353 - accuracy: 0.8907

accuracy: 89.07%
```

Console

2. Classifier to predict whether an offence is punishable or not

- This model is trained to predict if a case filed will be punished or not based on - The state, district, and court where the case is filed, gender of the judge, participation of women either as petitioner, defendant or advocate, and the first, most recent hearing dates.
- The following dispositions are considered to be **non-punishable**-
  Acquitted, cancelled, compromised, dismissed, rejected, withdrawn.
- The following dispositions are considered to be **punishable**-
  Bail order, bail refused, bail rejected, confession, convicted, execution, fine, prison, probation.
- RELU has been used for all layers except the final layer. The last layer uses Sigmoid (to ensure the value is between 0 and 1)
- The data split used for this training is 80:20 (train: test)
- **The model has an accuracy of ~86.78%** and has an **unbiased** confusion matrix.
- Better accuracy (~95%) can be achieved for models classifying if a case is acquitted or not, but it would be incomplete as any disposition without punishment is also being acquitted.

```
Epoch 1/3
97267/97267 [==============================] - 211s 2ms/step - loss: 0.4070 - accuracy: 0.8370
Epoch 2/3
97267/97267 [==============================] - 231s 2ms/step - loss: 0.3510 - accuracy: 0.8585
Epoch 3/3
97267/97267 [==============================] - 239s 2ms/step - loss: 0.3428 - accuracy: 0.8630
15198/15198 [==============================] - 24s 2ms/step - loss: 0.3326 - accuracy: 0.8678

accuracy: 86.78%
```