Python Assignment

Laveti BhanuPrakash
220583

April 2024

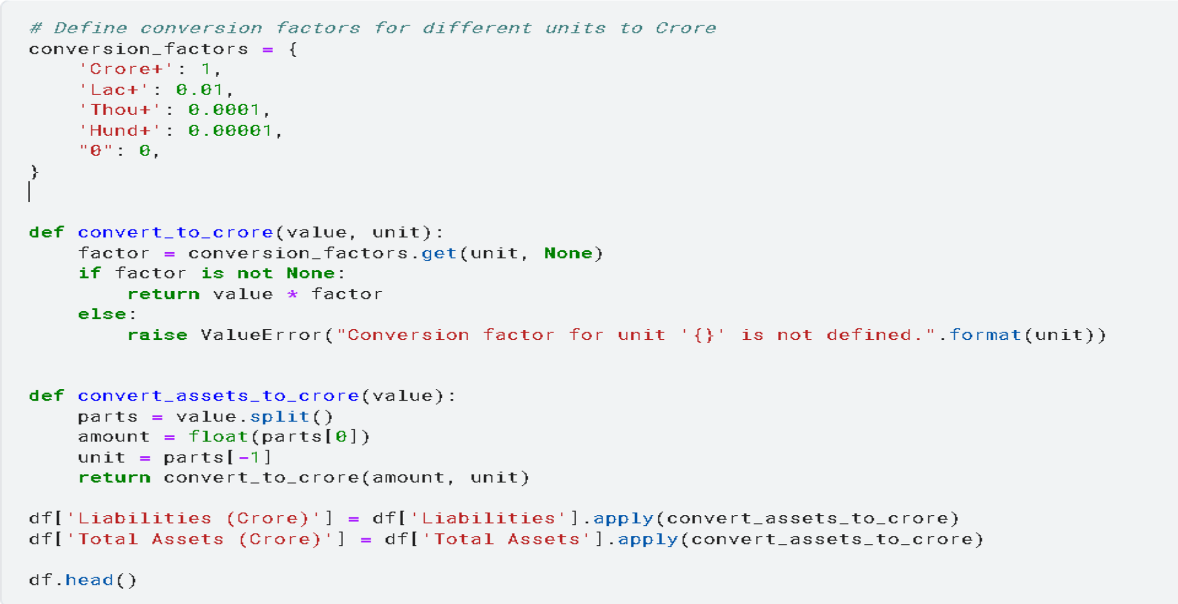
1 Methodology

In this section, we outline the methodology employed for preprocessing the dataset and preparing it for model training.

1.1 Data Preprocessing Steps

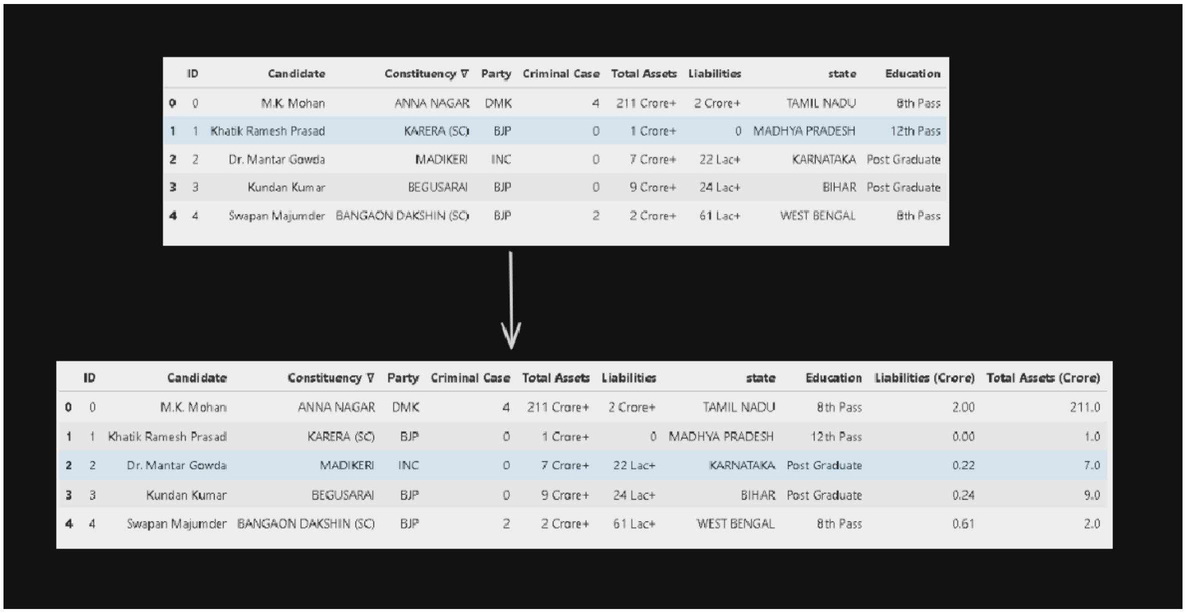
Data preprocessing involves transforming raw data into a clean, organized, and structured format that is suitable for analysis or model training. The primary goal of data preprocessing is to ensure that the data is of high quality, consistent, and ready to be used by machine learning algorithms.

1. **Converting Liabilities and Total Assets** :



```python
# Define conversion factors for different units to Crore
conversion_factors = {
    'Crore+': 1,
    'Lac+': 0.01,
    'Thou+': 0.0001,
    'Hund+': 0.00001,
    "0": 0,
}


def convert_to_crore(value, unit):
    factor = conversion_factors.get(unit, None)
    if factor is not None:
        return value * factor
    else:
        raise ValueError("Conversion factor for unit '{}' is not defined.".format(unit))


def convert_assets_to_crore(value):
    parts = value.split()
    amount = float(parts[0])
    unit = parts[-1]
    return convert_to_crore(amount, unit)

df['Liabilities (Crore)'] = df['Liabilities'].apply(convert_assets_to_crore)
df['Total Assets (Crore)'] = df['Total Assets'].apply(convert_assets_to_crore)

df.head()
```

Figure 1: code

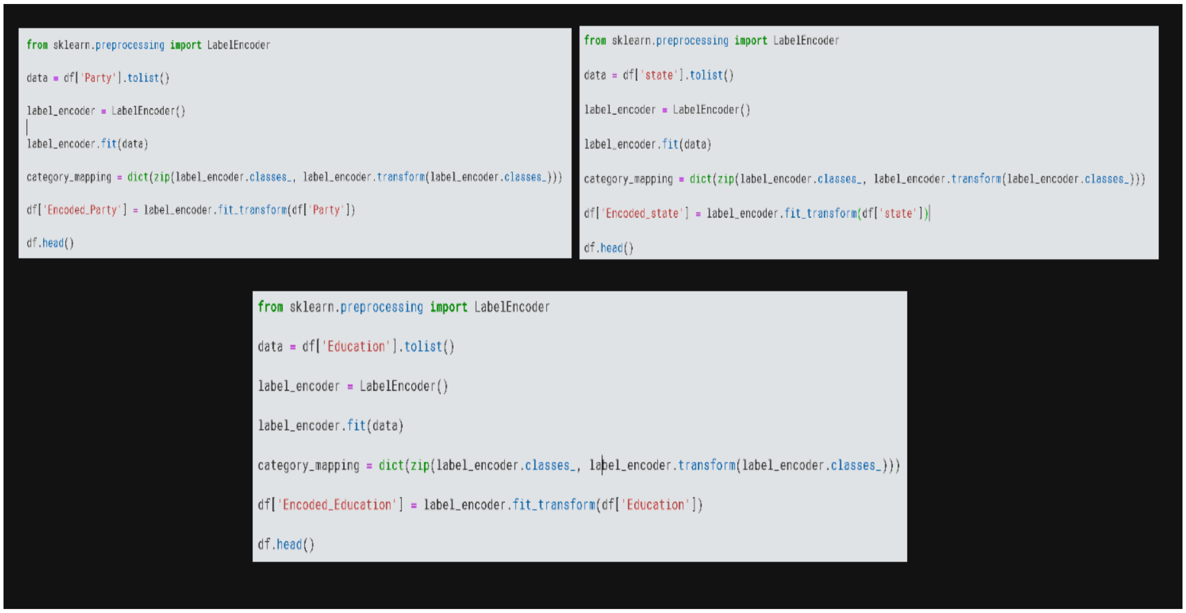


	ID	Candidate	Constituency ∇	Party	Criminal Case	Total Assets	Liabilities	state	Education
0	0	M.K. Mohan	ANNA NAGAR	DMK	4	211 Crore+	2 Crore+	TAMIL NADU	8th Pass
1	1	Khatik Ramesh Prasad	KARERA (SC)	BJP	0	1 Crore+	0	MADHYA PRADESH	12th Pass
2	2	Dr. Mantar Gowda	MADIKERI	INC	0	7 Crore+	22 Lac+	KARNATAKA	Post Graduate
3	3	Kundan Kumar	BEGUSARAI	BJP	0	9 Crore+	24 Lac+	BIHAR	Post Graduate
4	4	Swapan Majumder	BANGAON DAKSHIN (SC)	BJP	2	2 Crore+	61 Lac+	WEST BENGAL	8th Pass

	ID	Candidate	Constituency ∇	Party	Criminal Case	Total Assets	Liabilities	state	Education	Liabilities (Crore)	Total Assets (Crore)
0	0	M.K. Mohan	ANNA NAGAR	DMK	4	211 Crore+	2 Crore+	TAMIL NADU	8th Pass	2.00	211.0
1	1	Khatik Ramesh Prasad	KARERA (SC)	BJP	0	1 Crore+	0	MADHYA PRADESH	12th Pass	0.00	1.0
2	2	Dr. Mantar Gowda	MADIKERI	INC	0	7 Crore+	22 Lac+	KARNATAKA	Post Graduate	0.22	7.0
3	3	Kundan Kumar	BEGUSARAI	BJP	0	9 Crore+	24 Lac+	BIHAR	Post Graduate	0.24	9.0
4	4	Swapan Majumder	BANGAON DAKSHIN (SC)	BJP	2	2 Crore+	61 Lac+	WEST BENGAL	8th Pass	0.61	2.0

Figure 2: Transformation

2. **Encoding Education, States, Party column of trainData[3]**:

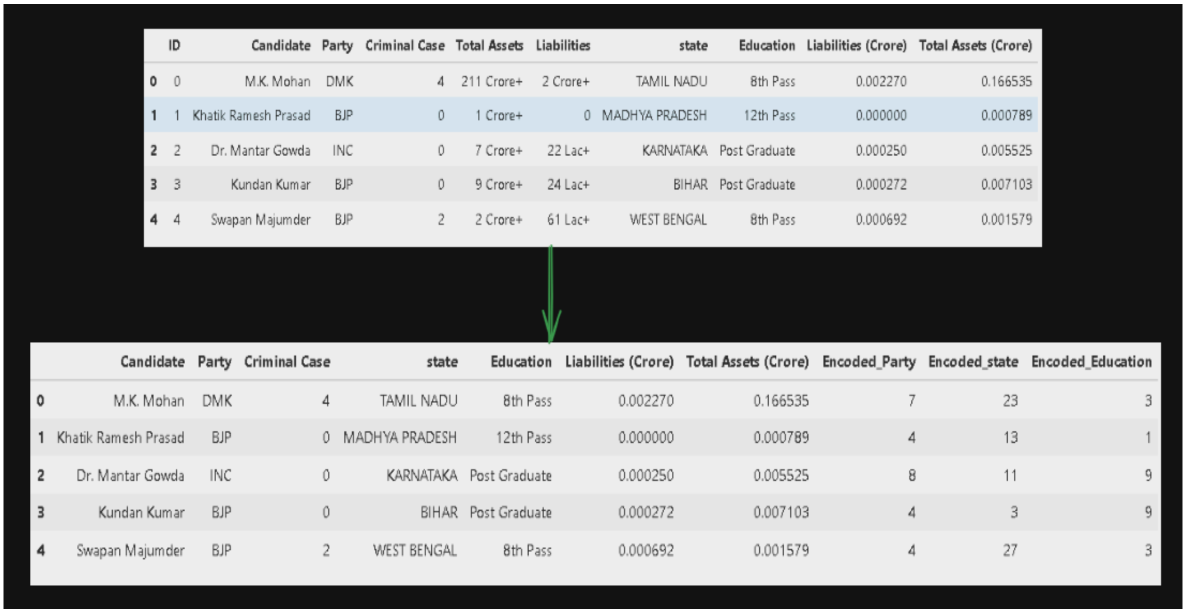


```
from sklearn.preprocessing import LabelEncoder

data = df['Party'].tolist()

label_encoder = LabelEncoder()

label_encoder.fit(data)

category_mapping = dict(zip(label_encoder.classes_, label_encoder.transform(label_encoder.classes_)))

df['Encoded_Party'] = label_encoder.fit_transform(df['Party'])

df.head()
```

```
from sklearn.preprocessing import LabelEncoder

data = df['state'].tolist()

label_encoder = LabelEncoder()

label_encoder.fit(data)

category_mapping = dict(zip(label_encoder.classes_, label_encoder.transform(label_encoder.classes_)))

df['Encoded_state'] = label_encoder.fit_transform(df['state'])

df.head()
```

```
from sklearn.preprocessing import LabelEncoder

data = df['Education'].tolist()

label_encoder = LabelEncoder()

label_encoder.fit(data)

category_mapping = dict(zip(label_encoder.classes_, label_encoder.transform(label_encoder.classes_)))

df['Encoded_Education'] = label_encoder.fit_transform(df['Education'])

df.head()
```

Figure 3: code

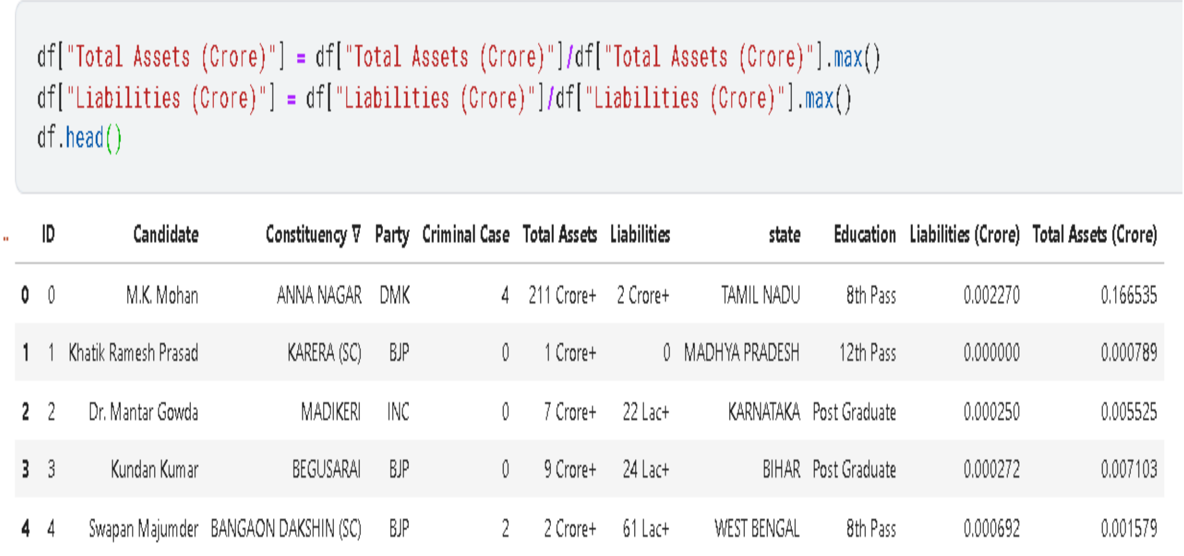


	ID	Candidate	Party	Criminal Case	Total Assets	Liabilities	state	Education	Liabilities (Crore)	Total Assets (Crore)
0	0	M.K. Mohan	DMK	4	211 Crore+	2 Crore+	TAMIL NADU	8th Pass	0.002270	0.166535
1	1	Khatik Ramesh Prasad	BJP	0	1 Crore+	0	MADHYA PRADESH	12th Pass	0.000000	0.000789
2	2	Dr. Mantar Gowda	INC	0	7 Crore+	22 Lac+	KARNATAKA	Post Graduate	0.000250	0.005525
3	3	Kundan Kumar	BJP	0	9 Crore+	24 Lac+	BIHAR	Post Graduate	0.000272	0.007103
4	4	Swapan Majumder	BJP	2	2 Crore+	61 Lac+	WEST BENGAL	8th Pass	0.000692	0.001579

	Candidate	Party	Criminal Case	state	Education	Liabilities (Crore)	Total Assets (Crore)	Encoded_Party	Encoded_state	Encoded_Education
0	M.K. Mohan	DMK	4	TAMIL NADU	8th Pass	0.002270	0.166535	7	23	3
1	Khatik Ramesh Prasad	BJP	0	MADHYA PRADESH	12th Pass	0.000000	0.000789	4	13	1
2	Dr. Mantar Gowda	INC	0	KARNATAKA	Post Graduate	0.000250	0.005525	8	11	9
3	Kundan Kumar	BJP	0	BIHAR	Post Graduate	0.000272	0.007103	4	3	9
4	Swapan Majumder	BJP	2	WEST BENGAL	8th Pass	0.000692	0.001579	4	27	3

Figure 4: Transformation

3. **Normalization for trainData [1]:**

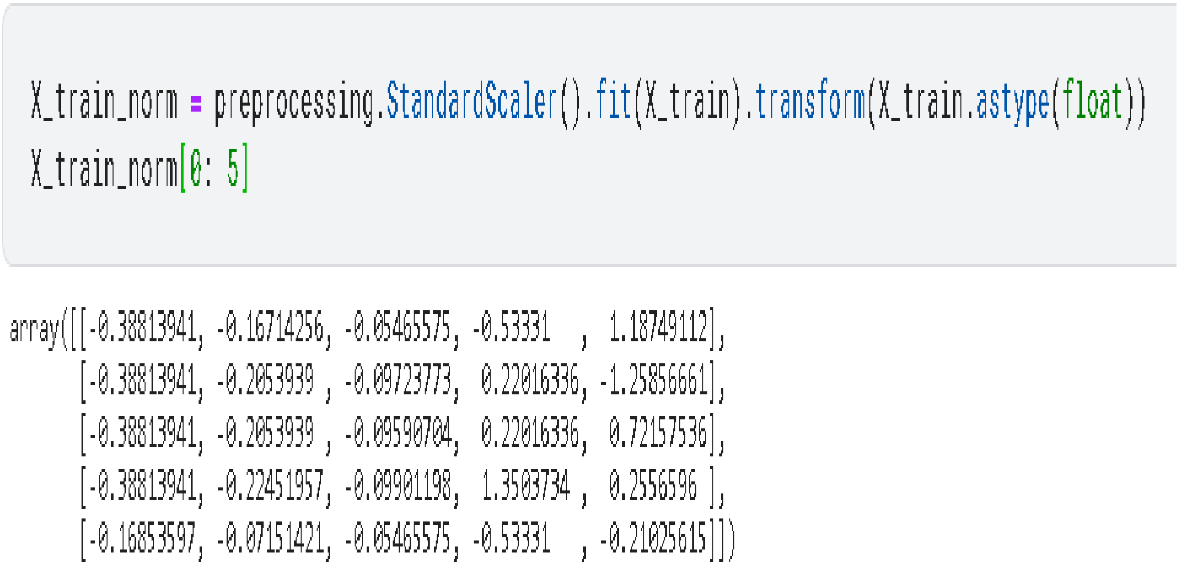


```
df["Total Assets (Crore)"] = df["Total Assets (Crore)"]/df["Total Assets (Crore)"].max()
df["Liabilities (Crore)"] = df["Liabilities (Crore)"]/df["Liabilities (Crore)"].max()
df.head()
```

	ID	Candidate	Constituency ∇	Party	Criminal Case	Total Assets	Liabilities	state	Education	Liabilities (Crore)	Total Assets (Crore)
0	0	M.K. Mohan	ANNA NAGAR	DMK	4	211 Crore+	2 Crore+	TAMIL NADU	8th Pass	0.002270	0.166535
1	1	Khatik Ramesh Prasad	KARERA (SC)	BJP	0	1 Crore+	0	MADHYA PRADESH	12th Pass	0.000000	0.000789
2	2	Dr. Mantar Gowda	MADIKERI	INC	0	7 Crore+	22 Lac+	KARNATAKA	Post Graduate	0.000250	0.005525
3	3	Kundan Kumar	BEGUSARAI	BJP	0	9 Crore+	24 Lac+	BIHAR	Post Graduate	0.000272	0.007103
4	4	Swapan Majumder	BANGAON DAKSHIN (SC)	BJP	2	2 Crore+	61 Lac+	WEST BENGAL	8th Pass	0.000692	0.001579

Figure 5: code

4. **Standardization for trainData [1]:**



```
X_train_norm = preprocessing.StandardScaler().fit(X_train).transform(X_train.astype(float))
X_train_norm[0: 5]
```

```
array([[-0.38813941, -0.16714256, -0.05465575, -0.53331   ,  1.18749112],
       [-0.38813941, -0.2053939 , -0.09723773,  0.22016336, -1.25856661],
       [-0.38813941, -0.2053939 , -0.09590704,  0.22016336,  0.72157536],
       [-0.38813941, -0.22451957, -0.09901198,  1.3503734 ,  0.2556596 ],
       [-0.16853597, -0.07151421, -0.05465575, -0.53331   , -0.21025615]])
```

Figure 6: code

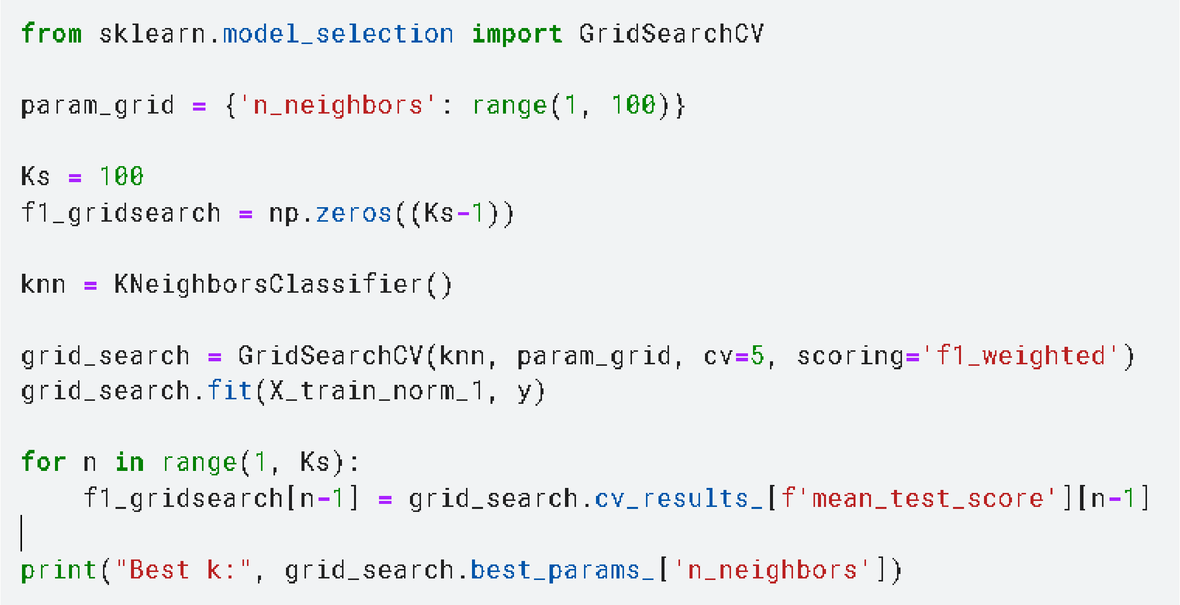
All the above methods are also applied for Test Dataset.

2 Experiment Details[2]

In this section, we provide details about the classifiers used for model.

Model	Hyperparameters
Knn	n_neighbors=[1, 100], weights='uniform', algorithm='auto' with GridSearchCv
Knn	n_neighbors=[1, 100], weights='uniform', algorithm='auto' with 80% TrainData, 20% TestData

2.0.1 GridSearchCv[4]



```
from sklearn.model_selection import GridSearchCV

param_grid = {'n_neighbors': range(1, 100)}

Ks = 100
f1_gridsearch = np.zeros((Ks-1))

knn = KNeighborsClassifier()

grid_search = GridSearchCV(knn, param_grid, cv=5, scoring='f1_weighted')
grid_search.fit(X_train_norm_1, y)

for n in range(1, Ks):
    f1_gridsearch[n-1] = grid_search.cv_results_[f'mean_test_score'][n-1]

print("Best k:", grid_search.best_params_['n_neighbors'])
```

Figure 7: code

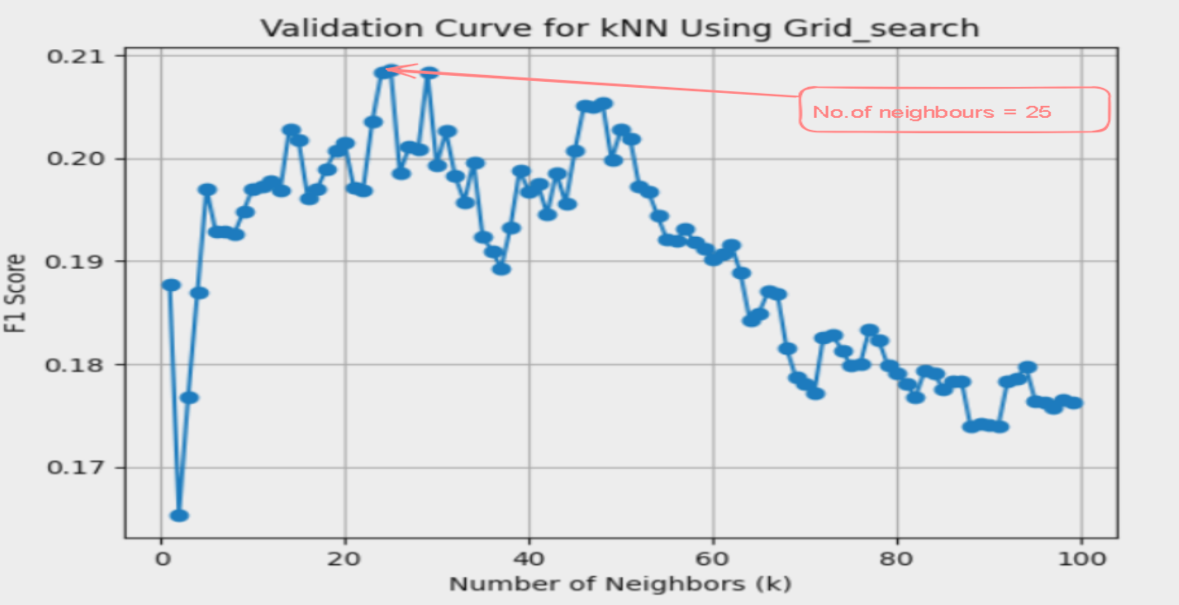
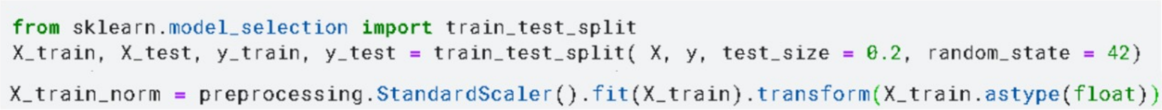


Figure 8: Graph

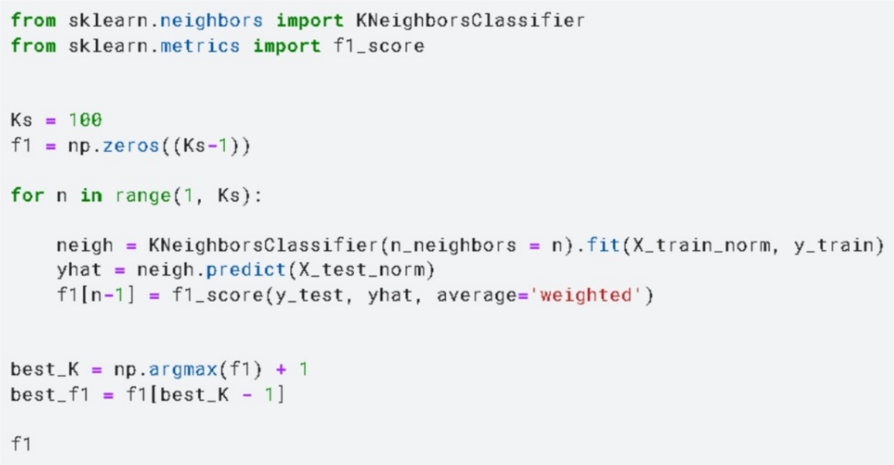
- **Public f1$_{\textbf{score}}$:** 0.24172

- **Private $\mathbf{f1_{score}}$:** 0.21126

2.0.2 Using TestSize



```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split( X, y, test_size = 0.2, random_state = 42)

X_train_norm = preprocessing.StandardScaler().fit(X_train).transform(X_train.astype(float))
```



```
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import f1_score


Ks = 100
f1 = np.zeros((Ks-1))

for n in range(1, Ks):

    neigh = KNeighborsClassifier(n_neighbors = n).fit(X_train_norm, y_train)
    yhat = neigh.predict(X_test_norm)
    f1[n-1] = f1_score(y_test, yhat, average='weighted')


best_K = np.argmax(f1) + 1
best_f1 = f1[best_K - 1]

f1
```

Figure 9: code

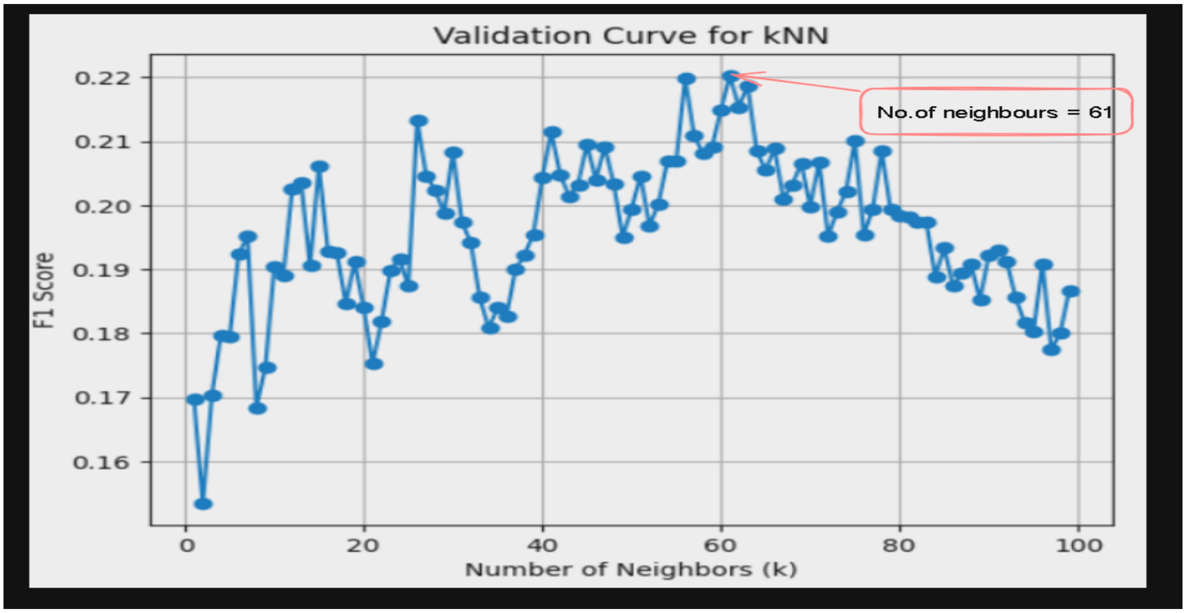


Figure 10: Graph

- **Public $\mathbf{f1_{score}}$:** 0.23010
- **Private $\mathbf{f1_{score}}$:** 0.22958

In the end, I utilized GridSearchCV with 25 neighbors as the parameter, and found that this model yielded the best results.

2.1 Criminal Cases

2.1.1 By Party

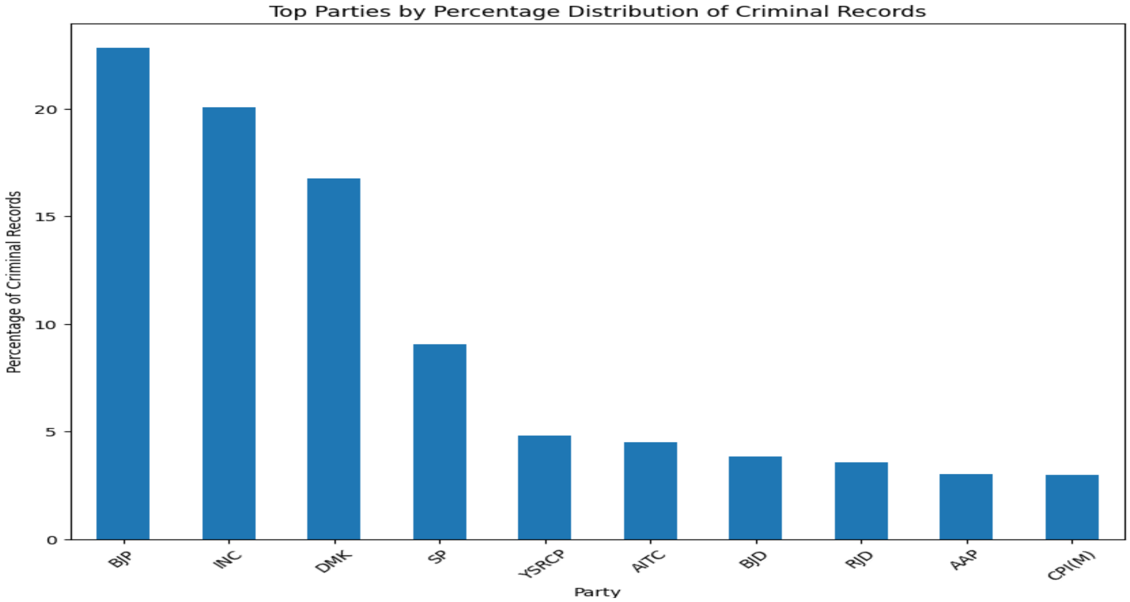


Figure 11: Graph

From this plot, we infer that the candidates of the BJP party have more criminal cases than any other party.

2.2 Wealthy Candidates

2.2.1 By Party

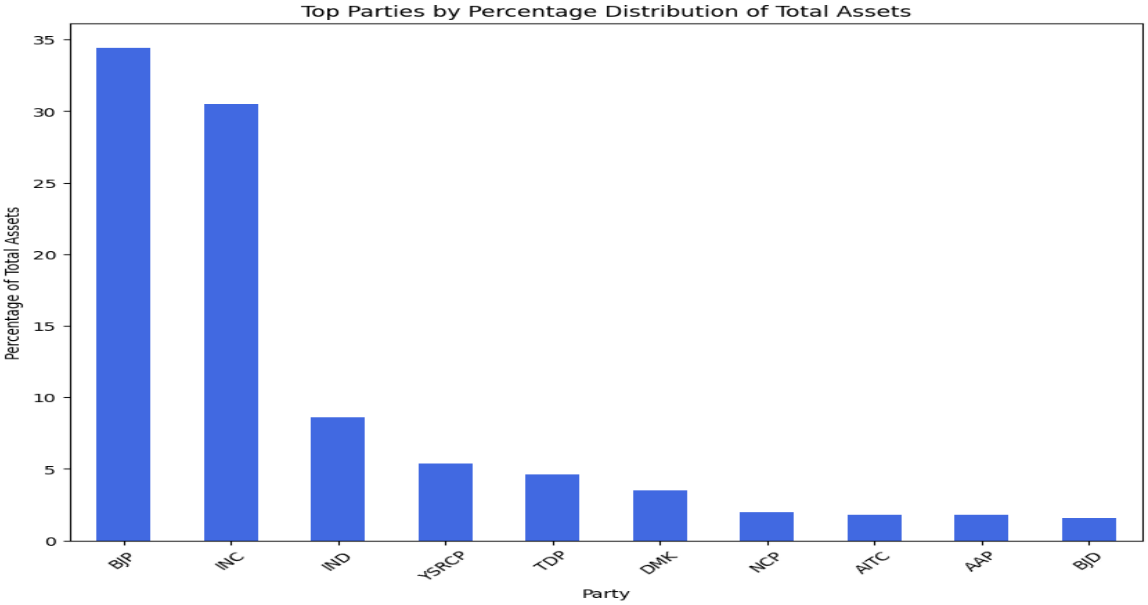


Figure 12: Graph

Analysis from the plots

From the above plots, we infer the following details:

- The "Average Criminal Cases by Party" plot revealed significant disparities in the average number of criminal cases among different political parties. Specifically, it highlighted that candidates affiliated with the BJP party tended to have higher average criminal cases compared to other parties. This insight prompted us to consider incorporating party affiliation as an important feature in our models to better capture this variation.
- From the wealth graphs we can easily conclude that the candidates from north eastern states are more wealthy as candidates from other states.

3 Results

- **Public $\mathbf{f1_{score}}$:** 0.24172
- **Private $\mathbf{f1_{score}}$:** 0.22958
- **Public Leaderboard Rank:** 96
- **Private Leaderboard Rank:** 126

Code Link: https://github.com/BhanuPrakash-123/CS253-Assignment-3

4 References

Various sources were used to build and fine tune the model. Relevant citations are given in the report. The sources can be accessed at:

[1]: https://scikit-learn.org/stable/modules/preprocessing.html

[2]: https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.KNeighborsClassifier.html

[3]: https://www.kdnuggets.com/2021/05/deal-with-categorical-data-machine-learning.html

[4]: https://towardsdatascience.com/gridsearchcv-for-beginners-db48a90114ee