randomforest

June 17, 2025

```
[ ]: #%%

     import pandas as pd
     from sklearn.ensemble import RandomForestClassifier
     from sklearn.metrics import classification_report
     from sklearn.model_selection import train_test_split

     X_train_data = pd.read_csv("law_data.csv")
     y_train_data = X_train_data.pop("first_pf")

     X_train_data = pd.get_dummies(data=X_train_data)
     X_train, X_test, y_train, y_test = train_test_split(X_train_data, y_train_data,␣
      ↪test_size=0.3, random_state=42)

     model = RandomForestClassifier(n_estimators=100, random_state=42)
     model.fit(X_train, y_train)

     predictions = model.predict(X_test)
     print(pd.DataFrame(classification_report(y_test, predictions,␣
      ↪output_dict=True)).T.to_markdown())
```

```
|              |   precision |   recall |   f1-score |     support |
|:-------------|------------:|---------:|-----------:|------------:|
| 0.0          |    0.538217 | 0.246715 |   0.338338 |  685        |
| 1.0          |    0.917095 | 0.975226 |   0.945268 | 5853        |
| accuracy     |    0.898899 | 0.898899 |   0.898899 |    0.898899 |
| macro avg    |    0.727656 | 0.610971 |   0.641803 | 6538        |
| weighted avg |    0.877399 | 0.898899 |   0.881679 | 6538        |
```

```
[ ]: # %%

     # Prediction per sex
     sex = X_test.groupby("sex")
     for name, groups in sex:
         pred = model.predict(groups)
         print("\n", name, groups.shape[0])
         print(pd.DataFrame(classification_report(y_test.loc[groups.index], pred,␣
      ↪output_dict=True)).T.to_markdown())
```

```
1 2894
|              |   precision |   recall |   f1-score |     support |
|:-------------|------------:|---------:|-----------:|------------:|
| 0.0          |    0.516854 | 0.282209 |   0.365079 |  326        |
| 1.0          |    0.913844 | 0.966511 |   0.93944  | 2568        |
| accuracy     |    0.889426 | 0.889426 |   0.889426 |    0.889426 |
| macro avg    |    0.715349 | 0.62436  |   0.65226  | 2894        |
| weighted avg |    0.869124 | 0.889426 |   0.87474  | 2894        |

2 3644
|              |   precision |   recall |   f1-score |     support |
|:-------------|------------:|---------:|-----------:|------------:|
| 0.0          |    0.566176 | 0.214485 |   0.311111 |  359        |
| 1.0          |    0.919612 | 0.98204  |   0.949801 | 3285        |
| accuracy     |    0.906422 | 0.906422 |   0.906422 |    0.906422 |
| macro avg    |    0.742894 | 0.598262 |   0.630456 | 3644        |
| weighted avg |    0.884792 | 0.906422 |   0.886879 | 3644        |
```

[ ]:
```
# %%

# Prediction per ethnicity
ethnicities = ["Amerindian", "Asian", "Black", "Hispanic", "Mexican", "Other",␣
 ↪"Puertorican", "White"]
# print(X_test)
for ethnicity in ethnicities:
    group = X_test.groupby("race_"+ethnicity)
    for name, groups in group:
        if name == True:
            pred = model.predict(groups)
            print("\n", ethnicity, groups.shape[0])
            print(pd.DataFrame(classification_report(y_test.loc[groups.index],␣
 ↪pred, output_dict=True)).T.to_markdown())
```

```
Amerindian 28
|              |   precision |   recall |   f1-score |   support |
|:-------------|------------:|---------:|-----------:|----------:|
| 0.0          |    0.7      | 0.7      |   0.7      | 10        |
| 1.0          |    0.833333 | 0.833333 |   0.833333 | 18        |
| accuracy     |    0.785714 | 0.785714 |   0.785714 |  0.785714 |
| macro avg    |    0.766667 | 0.766667 |   0.766667 | 28        |
| weighted avg |    0.785714 | 0.785714 |   0.785714 | 28        |

Asian 261
|              |   precision |   recall |   f1-score |    support |
|:-------------|------------:|---------:|-----------:|-----------:|
| 0.0          |    0.583333 | 0.152174 |   0.241379 |  46        |
| 1.0          |    0.843373 | 0.976744 |   0.905172 | 215        |
| accuracy     |    0.831418 | 0.831418 |   0.831418 |   0.831418 |
| macro avg    |    0.713353 | 0.564459 |   0.573276 | 261        |
| weighted avg |    0.797543 | 0.831418 |   0.788182 | 261        |

 Black 402
|              |   precision |   recall |   f1-score |    support |
|:-------------|------------:|---------:|-----------:|-----------:|
| 0.0          |    0.534722 | 0.566176 |   0.55     | 136        |
| 1.0          |    0.771318 | 0.74812  |   0.759542 | 266        |
| accuracy     |    0.686567 | 0.686567 |   0.686567 |   0.686567 |
| macro avg    |    0.65302  | 0.657148 |   0.654771 | 402        |
| weighted avg |    0.691276 | 0.686567 |   0.688652 | 402        |

 Hispanic 118
|              |   precision |   recall |   f1-score |    support |
|:-------------|------------:|---------:|-----------:|-----------:|
| 0.0          |    0.5      | 0.4      |   0.444444 |  25        |
| 1.0          |    0.846939 | 0.892473 |   0.86911  |  93        |
| accuracy     |    0.788136 | 0.788136 |   0.788136 |   0.788136 |
| macro avg    |    0.673469 | 0.646237 |   0.656777 | 118        |
| weighted avg |    0.773435 | 0.788136 |   0.779138 | 118        |

 Mexican 120
|              |   precision |   recall |   f1-score |    support |
|:-------------|------------:|---------:|-----------:|-----------:|
| 0.0          |    0.666667 | 0.413793 |   0.510638 |  29        |
| 1.0          |    0.833333 | 0.934066 |   0.880829 |  91        |
| accuracy     |    0.808333 | 0.808333 |   0.808333 |   0.808333 |
| macro avg    |    0.75     | 0.67393  |   0.695734 | 120        |
| weighted avg |    0.793056 | 0.808333 |   0.791366 | 120        |

 Other 90
|              |   precision |   recall |   f1-score |   support |
|:-------------|------------:|---------:|-----------:|----------:|
| 0.0          |    0.727273 | 0.615385 |   0.666667 | 13        |
| 1.0          |    0.936709 | 0.961039 |   0.948718 | 77        |
| accuracy     |    0.911111 | 0.911111 |   0.911111 |  0.911111 |
| macro avg    |    0.831991 | 0.788212 |   0.807692 | 90        |
| weighted avg |    0.906457 | 0.911111 |   0.907977 | 90        |

 Puertorican 37
|              |   precision |   recall |   f1-score |   support |
|:-------------|------------:|---------:|-----------:|----------:|
| 0.0          |    0.555556 | 0.416667 |   0.47619  | 12        |
| 1.0          |    0.75     | 0.84     |   0.792453 | 25        |
| accuracy     |    0.702703 | 0.702703 |   0.702703 |  0.702703 |
| macro avg    |    0.652778 | 0.628333 |   0.634322 | 37        |
| weighted avg |    0.686937 | 0.702703 |   0.689881 | 37         |

 White 5482
|              |   precision |   recall |   f1-score |    support |
|:-------------|------------:|---------:|-----------:|-----------:|
| 0.0          |    0.477778 | 0.103865 |   0.170635 |  414       |
| 1.0          |    0.931194 | 0.990726 |   0.960038 | 5068       |
| accuracy     |    0.92375  | 0.92375  |   0.92375  |    0.92375 |
| macro avg    |    0.704486 | 0.547295 |   0.565337 | 5482       |
| weighted avg |    0.896952 | 0.92375  |   0.900423 | 5482       |
```

[ ]:
```python
# %%

# Prediction per region
regions = ["FW","GL","MS","MW","Mt","NE","NG","NW","PO","SC","SE"]
for region in regions:
    group = X_test.groupby("region_first_"+region)
    for name, groups in group:
        if name == True:
            pred = model.predict(groups)
            print("\n", region, groups.shape[0])
            print(pd.DataFrame(classification_report(y_test.loc[groups.index],␣
 ↪pred, output_dict=True)).T.to_markdown())
```

```
 FW 905
|              |   precision |   recall |   f1-score |    support |
|:-------------|------------:|---------:|-----------:|-----------:|
| 0.0          |    0.574074 | 0.219858 |   0.317949 | 141        |
| 1.0          |    0.87074  | 0.969895 |   0.917647 | 764        |
| accuracy     |    0.853039 | 0.853039 |   0.853039 |   0.853039 |
| macro avg    |    0.722407 | 0.594877 |   0.617798 | 905        |
| weighted avg |    0.824519 | 0.853039 |   0.824213 | 905        |

 GL 1131
|              |   precision |   recall |   f1-score |     support |
|:-------------|------------:|---------:|-----------:|------------:|
| 0.0          |    0.789474 | 0.189873 |   0.306122 |   79        |
| 1.0          |    0.942446 | 0.996198 |   0.968577 | 1052        |
| accuracy     |    0.939876 | 0.939876 |   0.939876 |    0.939876 |
| macro avg    |    0.86596  | 0.593036 |   0.63735  | 1131        |
| weighted avg |    0.931761 | 0.939876 |   0.922304 | 1131        |

 MS 701
|              |   precision |   recall |   f1-score |    support |
|:-------------|------------:|---------:|-----------:|-----------:|
| 0.0          |    0.583333 | 0.3125   |   0.406977 | 112        |
| 1.0          |    0.879875 | 0.957555 |   0.917073 | 589        |
| accuracy     |    0.854494 | 0.854494 |   0.854494 |   0.854494 |
| macro avg    |    0.731604 | 0.635028 |   0.662025 | 701        |
| weighted avg |    0.832496 | 0.854494 |   0.835574 | 701        |
```

```
 MW 298
|              |   precision |   recall |   f1-score |    support |
|:-------------|------------:|---------:|-----------:|-----------:|
| 0.0          |    0.428571 | 0.130435 |   0.2      |  23        |
| 1.0          |    0.931271 | 0.985455 |   0.957597 | 275        |
| accuracy     |    0.919463 | 0.919463 |   0.919463 |   0.919463 |
| macro avg    |    0.679921 | 0.557945 |   0.578799 | 298        |
| weighted avg |    0.892472 | 0.919463 |   0.899125 | 298        |
```

```
 Mt 367
|              |   precision |   recall |   f1-score |    support |
|:-------------|------------:|---------:|-----------:|-----------:|
| 0.0          |    0.722222 | 0.254902 |   0.376812 |  51        |
| 1.0          |    0.891117 | 0.984177 |   0.935338 | 316        |
| accuracy     |    0.882834 | 0.882834 |   0.882834 |   0.882834 |
| macro avg    |    0.80667  | 0.61954  |   0.656075 | 367        |
| weighted avg |    0.867647 | 0.882834 |   0.857723 | 367        |
```

```
 NE 1300
|              |   precision |   recall |   f1-score |     support |
|:-------------|------------:|---------:|-----------:|------------:|
| 0.0          |    0.42623  | 0.245283 |   0.311377 |  106        |
| 1.0          |    0.935432 | 0.970687 |   0.952733 | 1194        |
| accuracy     |    0.911538 | 0.911538 |   0.911538 |    0.911538 |
| macro avg    |    0.680831 | 0.607985 |   0.632055 | 1300        |
| weighted avg |    0.893912 | 0.911538 |   0.900438 | 1300        |
```

```
 NG 365
|              |   precision |   recall |   f1-score |    support |
|:-------------|------------:|---------:|-----------:|-----------:|
| 0.0          |    0.538462 | 0.205882 |   0.297872 |  34        |
| 1.0          |    0.923295 | 0.981873 |   0.951684 | 331        |
| accuracy     |    0.909589 | 0.909589 |   0.909589 |   0.909589 |
| macro avg    |    0.730878 | 0.593878 |   0.624778 | 365        |
| weighted avg |    0.887448 | 0.909589 |   0.890781 | 365        |
```

```
 NW 45
|              |   precision |   recall |   f1-score |   support |
|:-------------|------------:|---------:|-----------:|----------:|
| 0.0          |    1        | 0.125    |   0.222222 |  8        |
| 1.0          |    0.840909 | 1        |   0.91358  | 37        |
| accuracy     |    0.844444 | 0.844444 |   0.844444 |  0.844444 |
| macro avg    |    0.920455 | 0.5625   |   0.567901 | 45        |
| weighted avg |    0.869192 | 0.844444 |   0.790672 | 45        |
SC 642
|              |   precision |   recall |   f1-score |    support |
|:-------------|------------:|---------:|-----------:|-----------:|
| 0.0          |    0.489796 | 0.324324 |   0.390244 |  74        |
| 1.0          |    0.915683 | 0.955986 |   0.935401 | 568        |
| accuracy     |    0.883178 | 0.883178 |   0.883178 |   0.883178 |
| macro avg    |    0.702739 | 0.640155 |   0.662822 | 642        |
| weighted avg |    0.866593 | 0.883178 |   0.872563 | 642        |

SE 784
|              |   precision |   recall |   f1-score |    support |
|:-------------|------------:|---------:|-----------:|-----------:|
| 0.0          |    0.4375   | 0.245614 |   0.314607 |  57        |
| 1.0          |    0.942819 | 0.975241 |   0.958756 | 727        |
| accuracy     |    0.922194 | 0.922194 |   0.922194 |   0.922194 |
| macro avg    |    0.69016  | 0.610427 |   0.636681 | 784        |
| weighted avg |    0.90608  | 0.922194 |   0.911924 | 784        |
```