

PREDICTING CRIME LEVELS WITH NYPD

CS 513 - B, GROUP 22

OUR TEAM



Anna Hauk
CWID: 20014598



Catherine DeMario
CWID: 10445359



Kieran Corson
CWID: 20010254



Jorge Ponce
CWID: 20014830

TOPICS TO COVER...

01

PROJECT
MOTIVATION

02

EXPLORATORY
DATA ANALYSIS

03

MODELS

04

CONCLUSIONS

PROJECT MOTIVATION

Project Motivation

- From NYC open Data for NYPD arrests in 2024
- New York City is the most populous city in the US
 - Brooklyn, Queens, Manhattan, Bronx, Staten Island (order of population)
- Crime statistics can be used for law enforcement
- Crime has increased dramatically in recent years

Central Question

**How can we leverage patterns and
offense levels to better allocate
resources and prevent crime?**

Other Questions

How does incorporating demographic data affect the accuracy of our model?

Are some areas more susceptible to arrests than others?

What's the trend of the top offenses over the year?

NYPD Arrest Data 2024

Felony

**Highest
Severity**

Misdemeanor

**Lower
Severity**

Violation

**Lowest Severity
(ex: traffic
violation)**

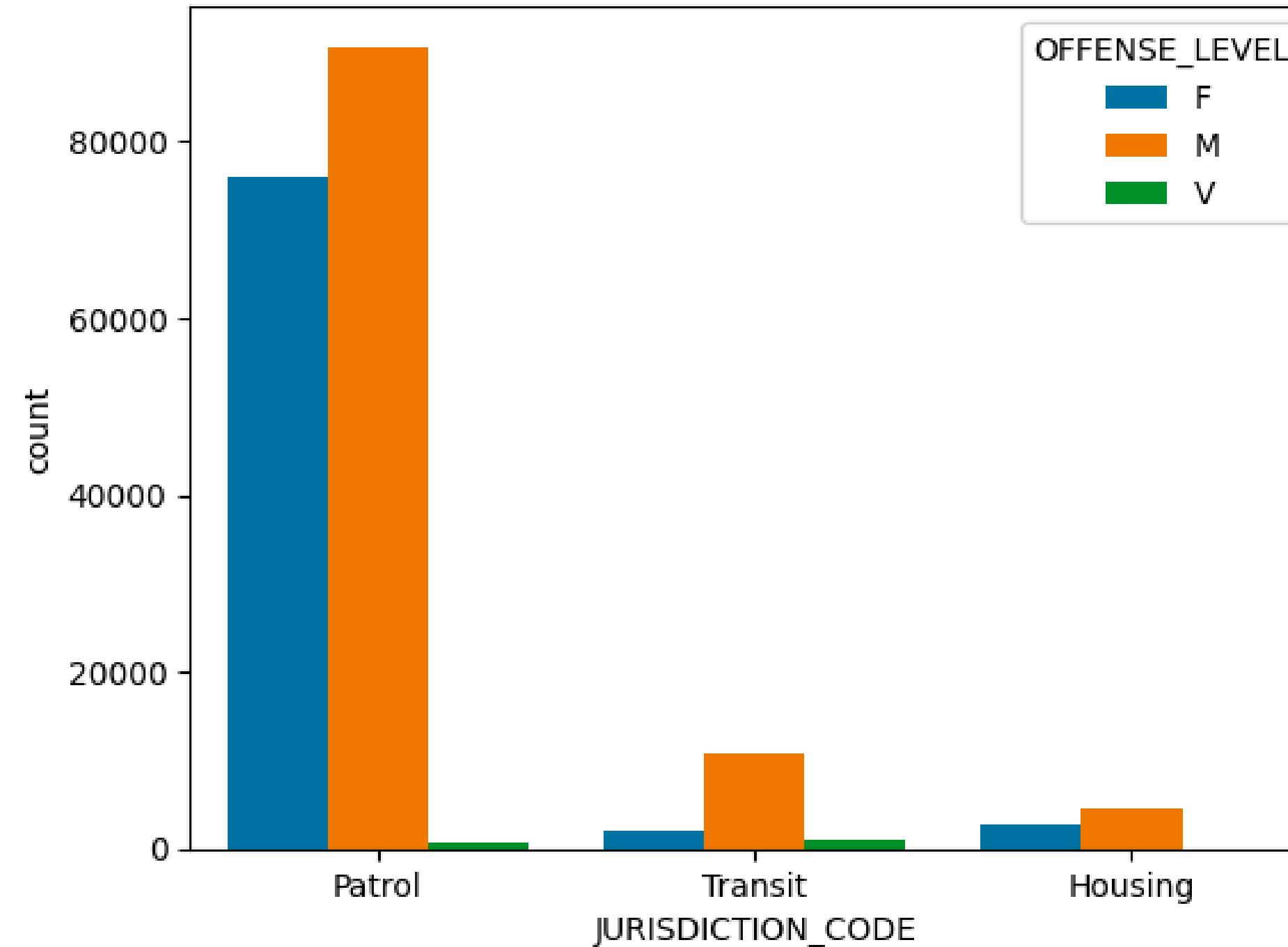
EXPLORATORY DATA ANALYSIS

The Data

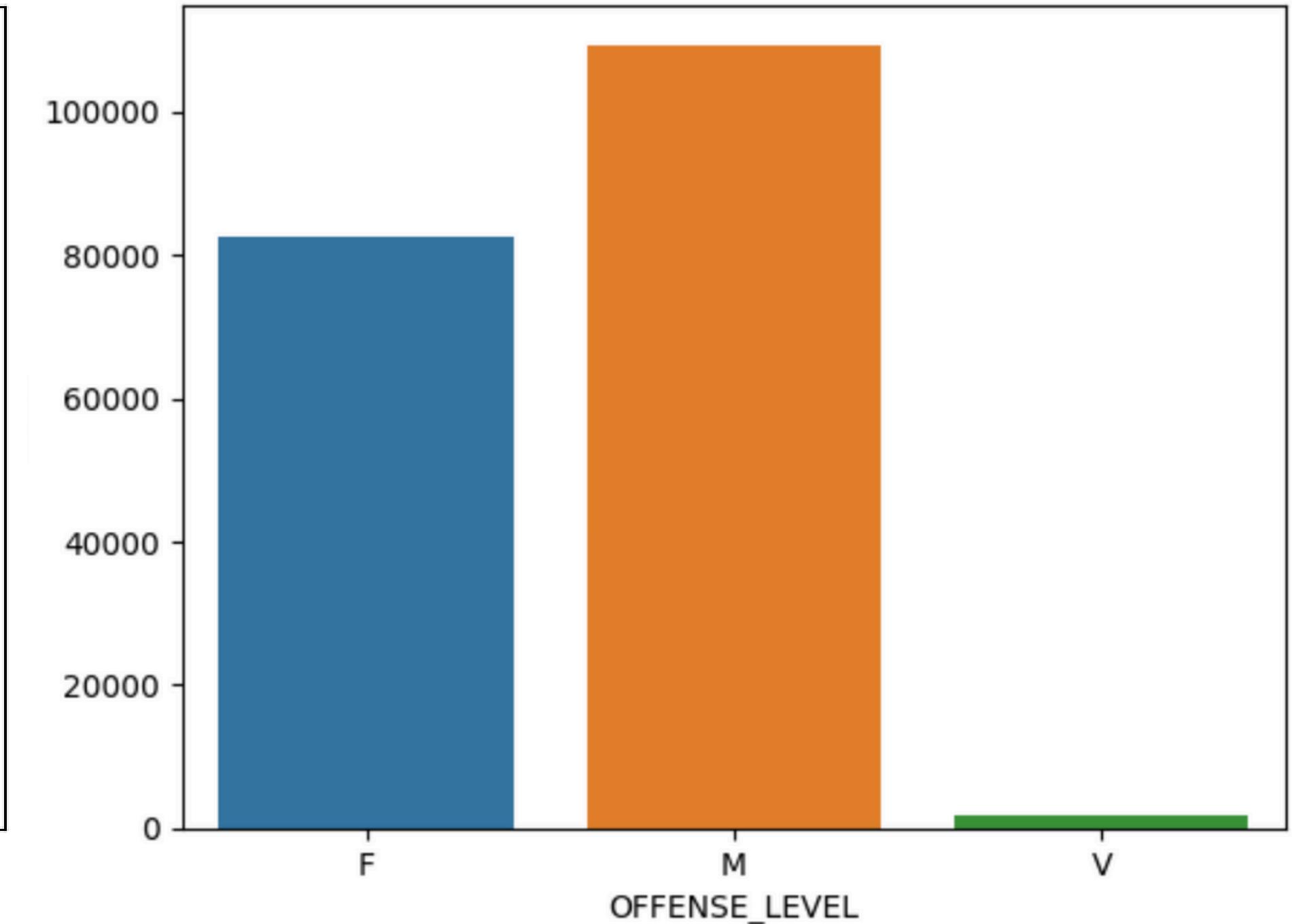
- Data from 2024, updated quarterly (from NYC Open Data)
- 195,447 rows × 19 columns --> 193,610 rows after cleaning
- Features like...
 - **ID's, Arrest Date**
 - **PD_CD, KY_CD**: Codes for the offenses (KY more general)
 - **PD_DESC, OFNS_DESC**: Description of crime
 - **Precinct, Jurisdiction, Neighborhood, Latitude/Longitude**
 - **Demographics**
 - Age, Gender, Race
 - **Offense Level (What we're predicting!!!)**

EDA

Jurisdiction Codes vs OFFENSE_LEVEL

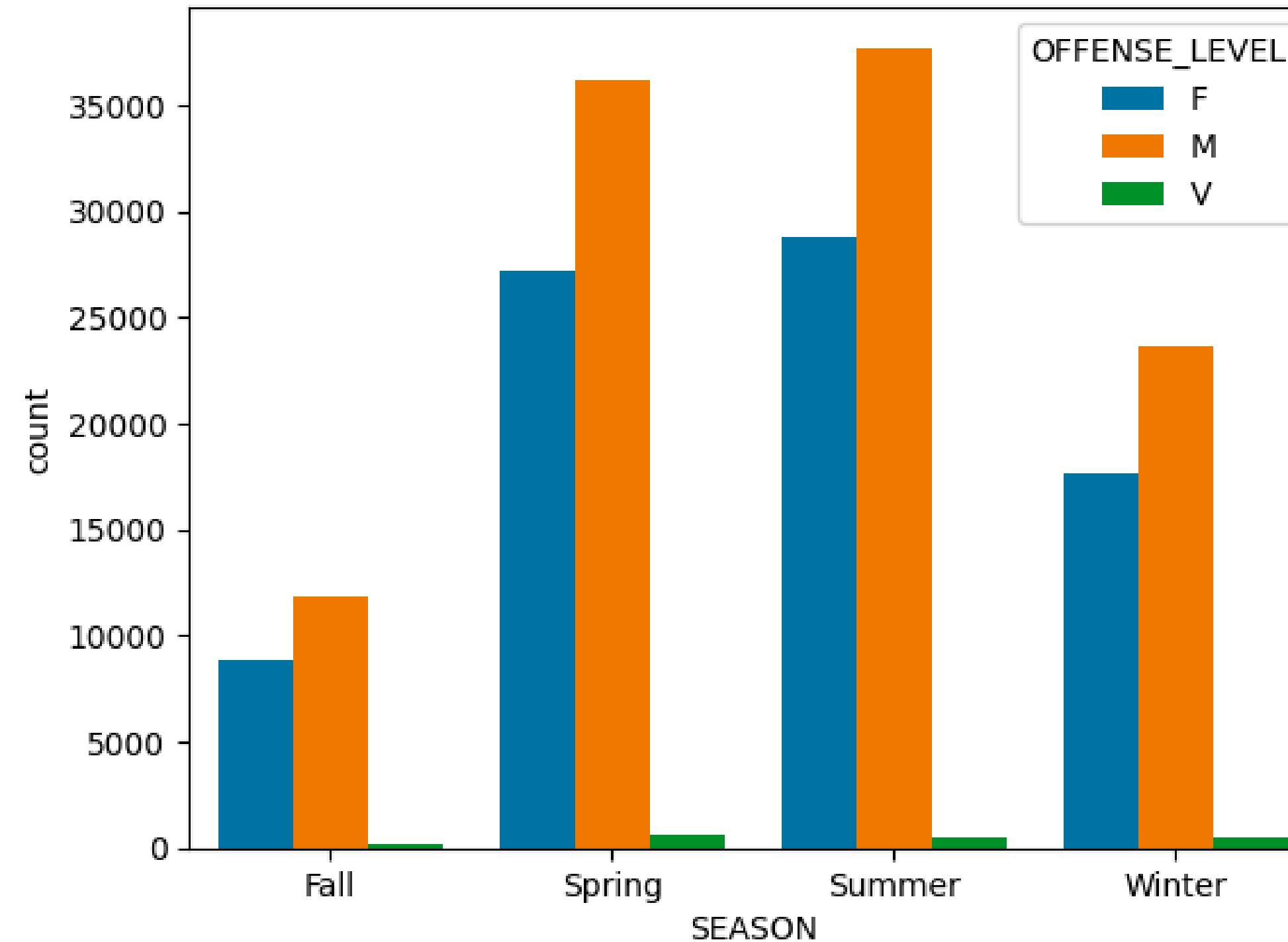


Frequency of each Offense Level

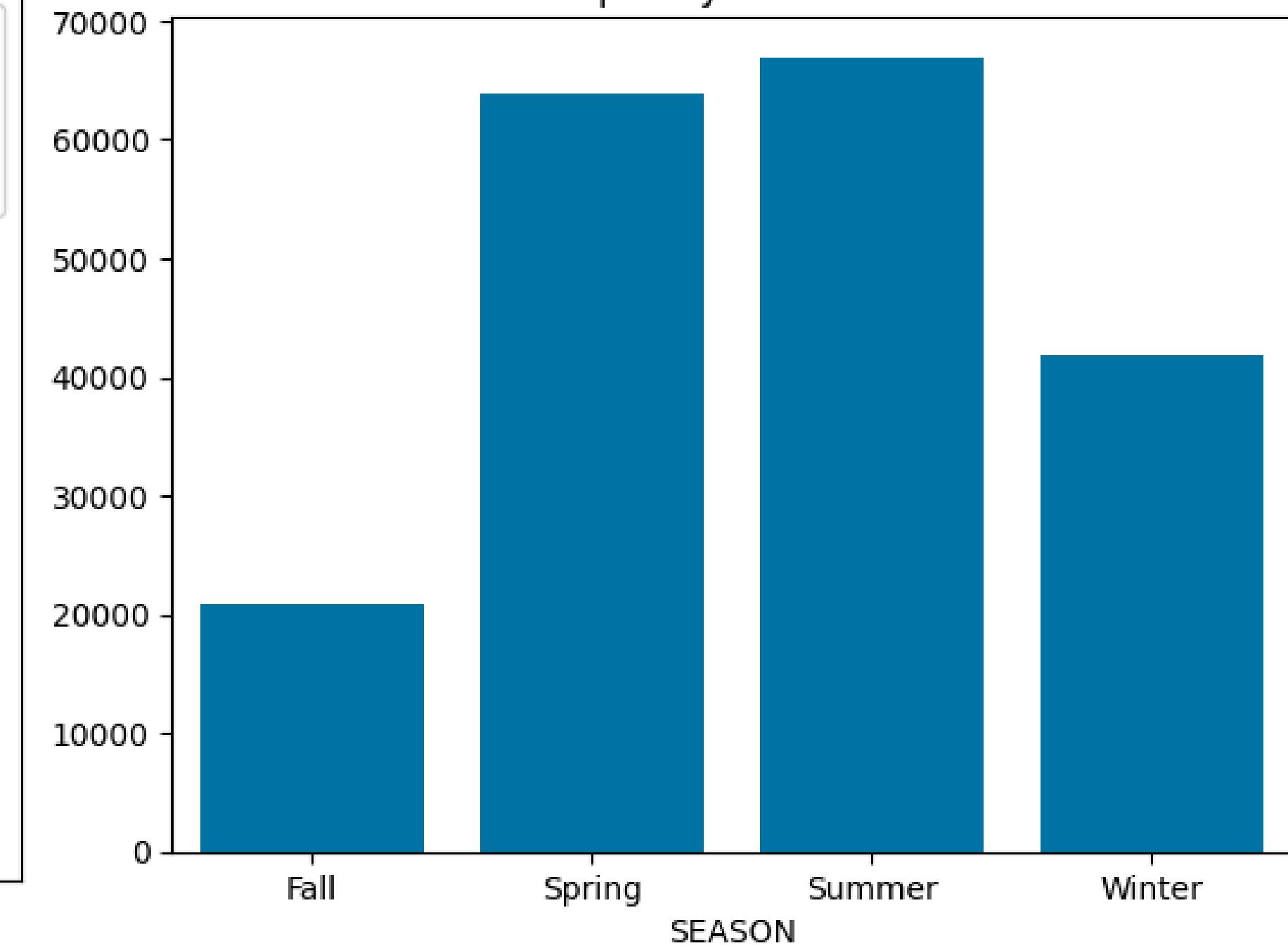


EDA

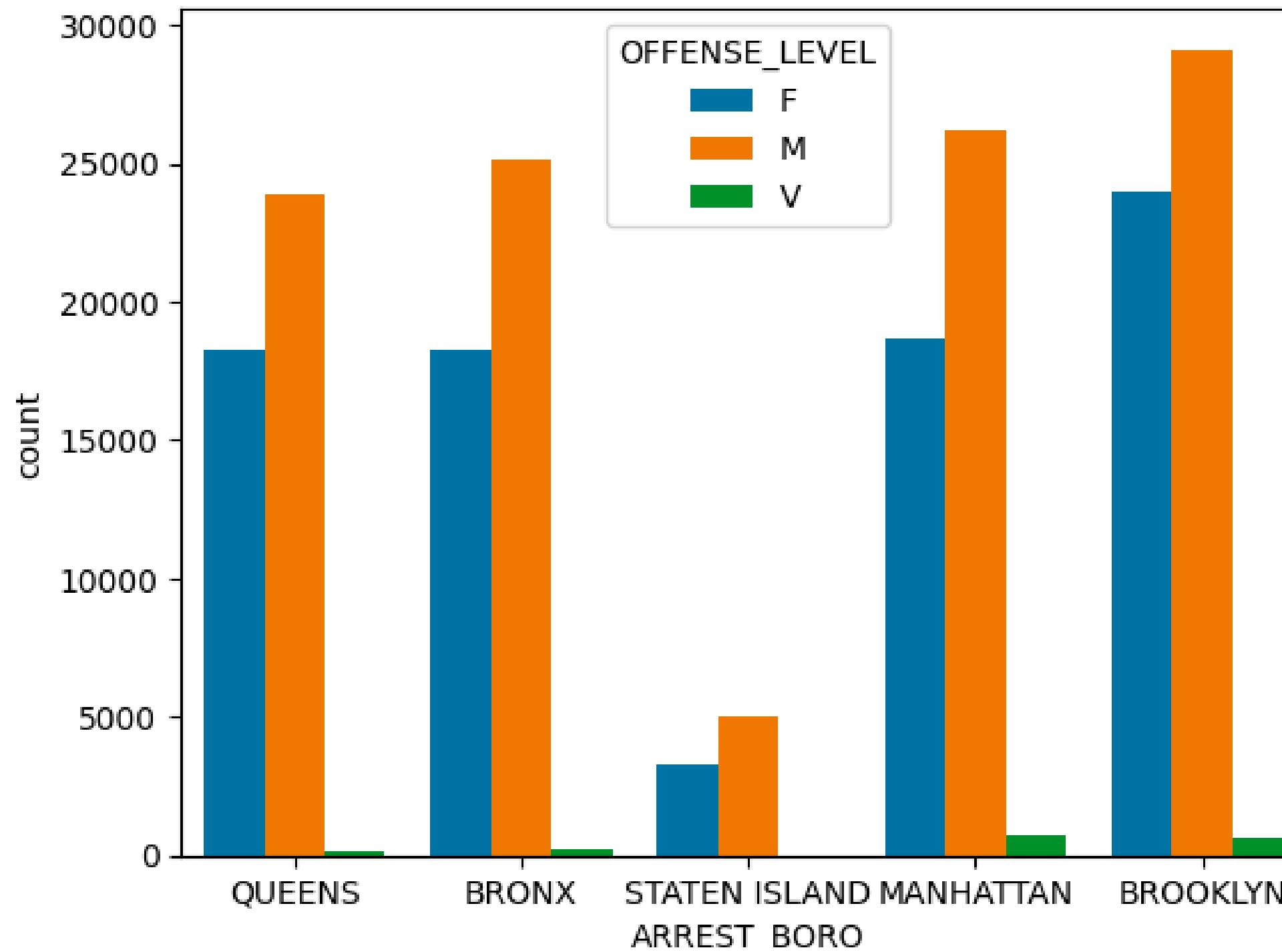
Season vs Offense Level



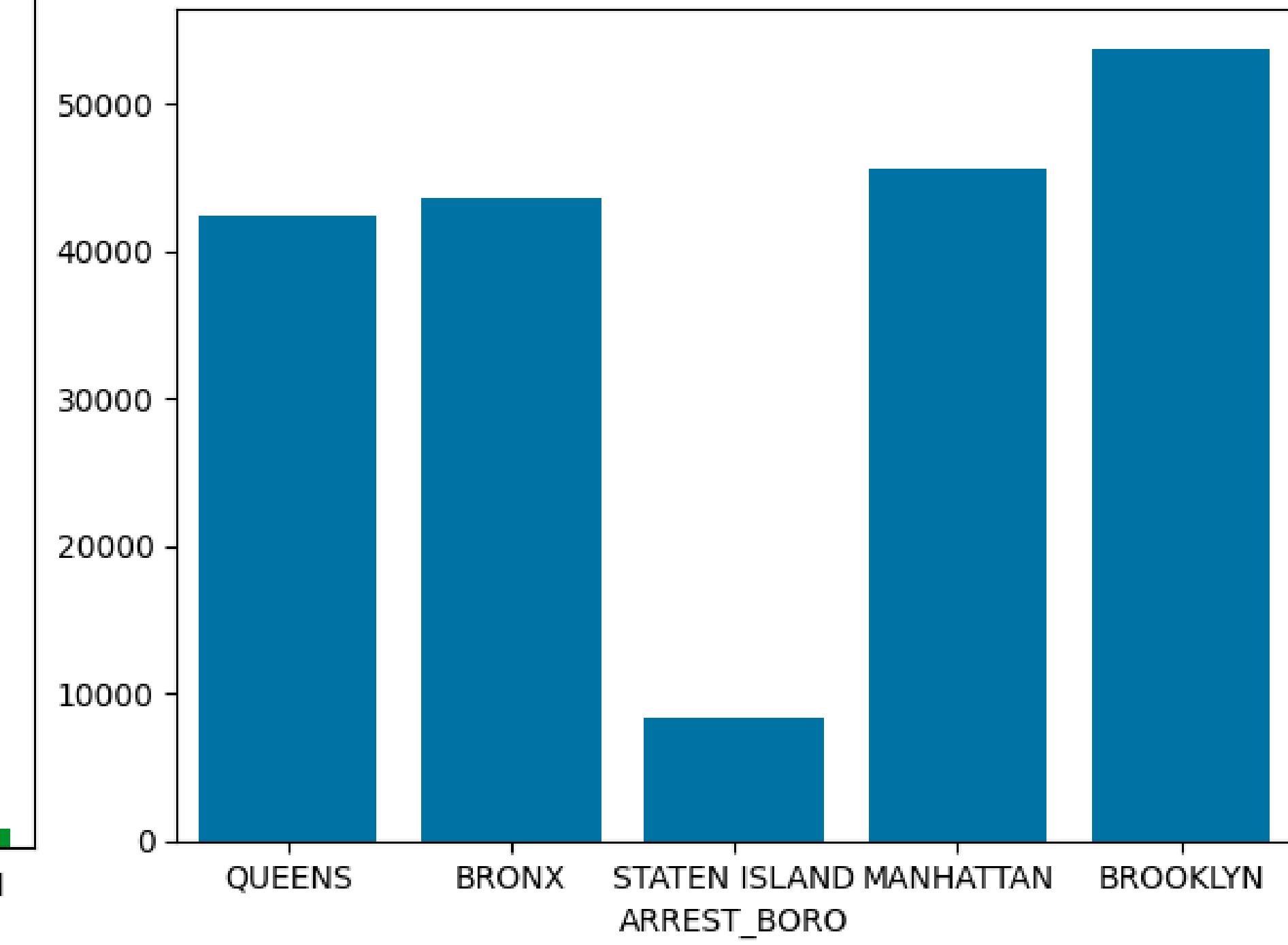
Frequency of Seasons



Boroughs vs Offense Level

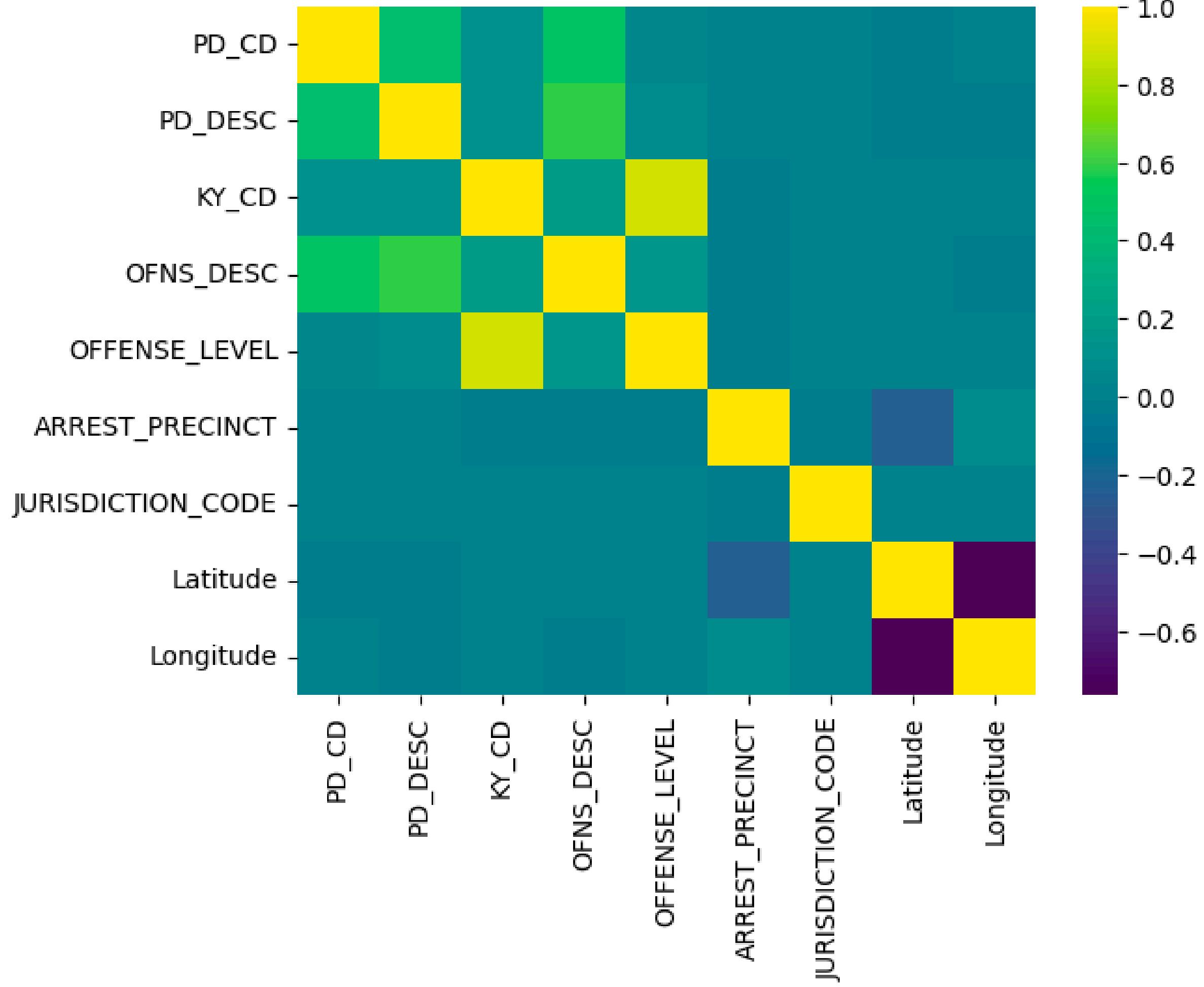


Frequency of Boroughs



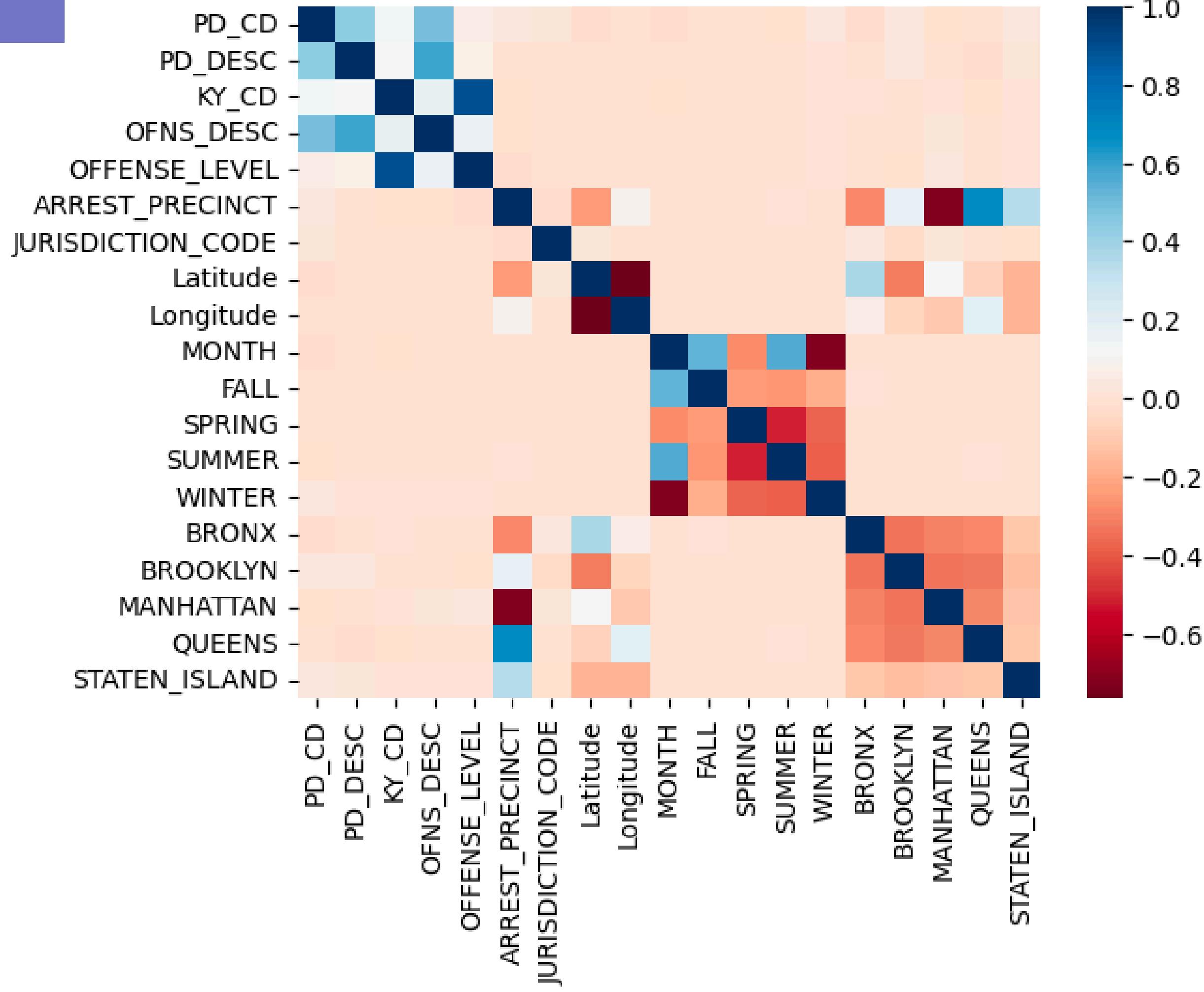
EDA

Column Correlation Heatmap



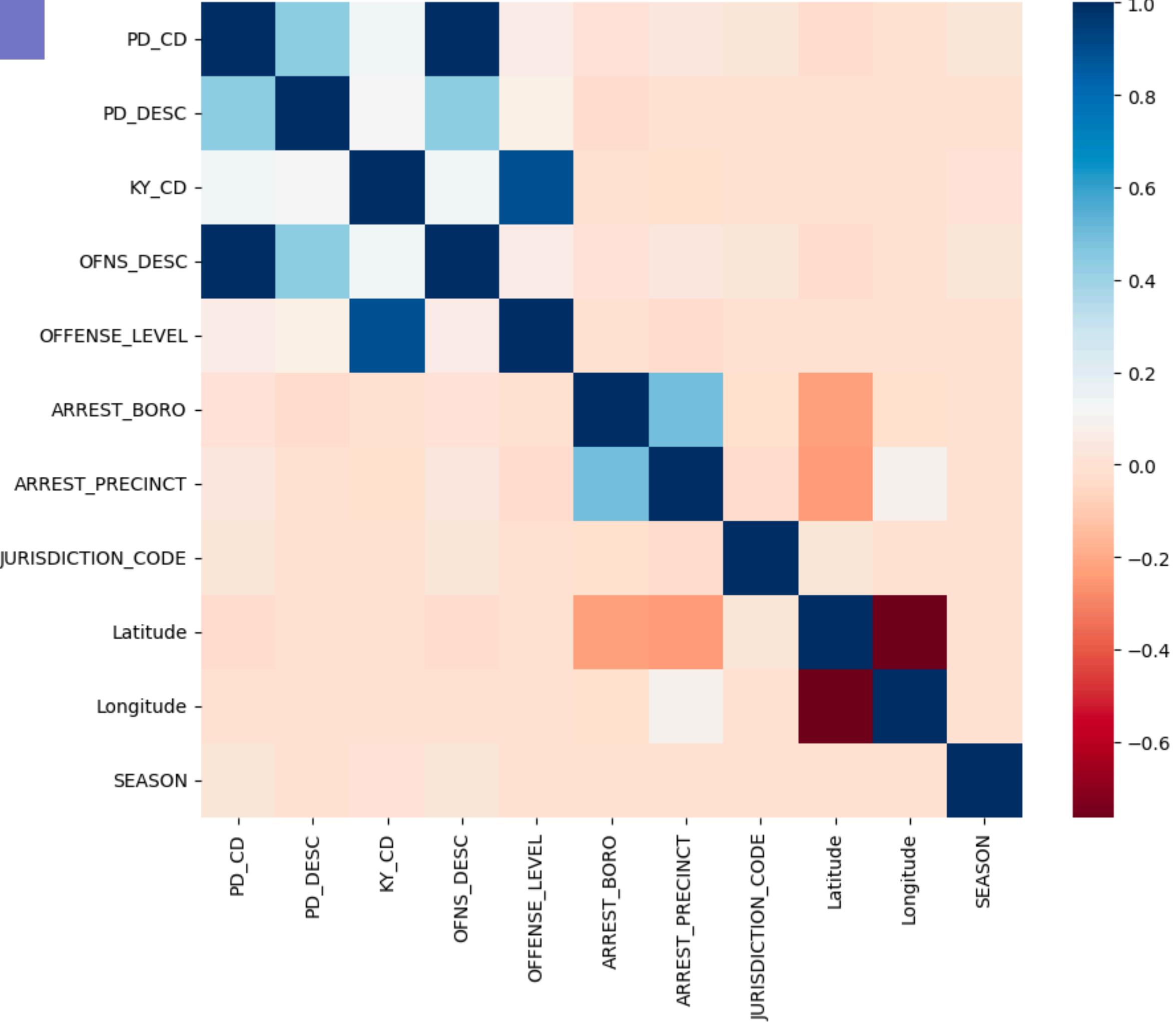
EDA

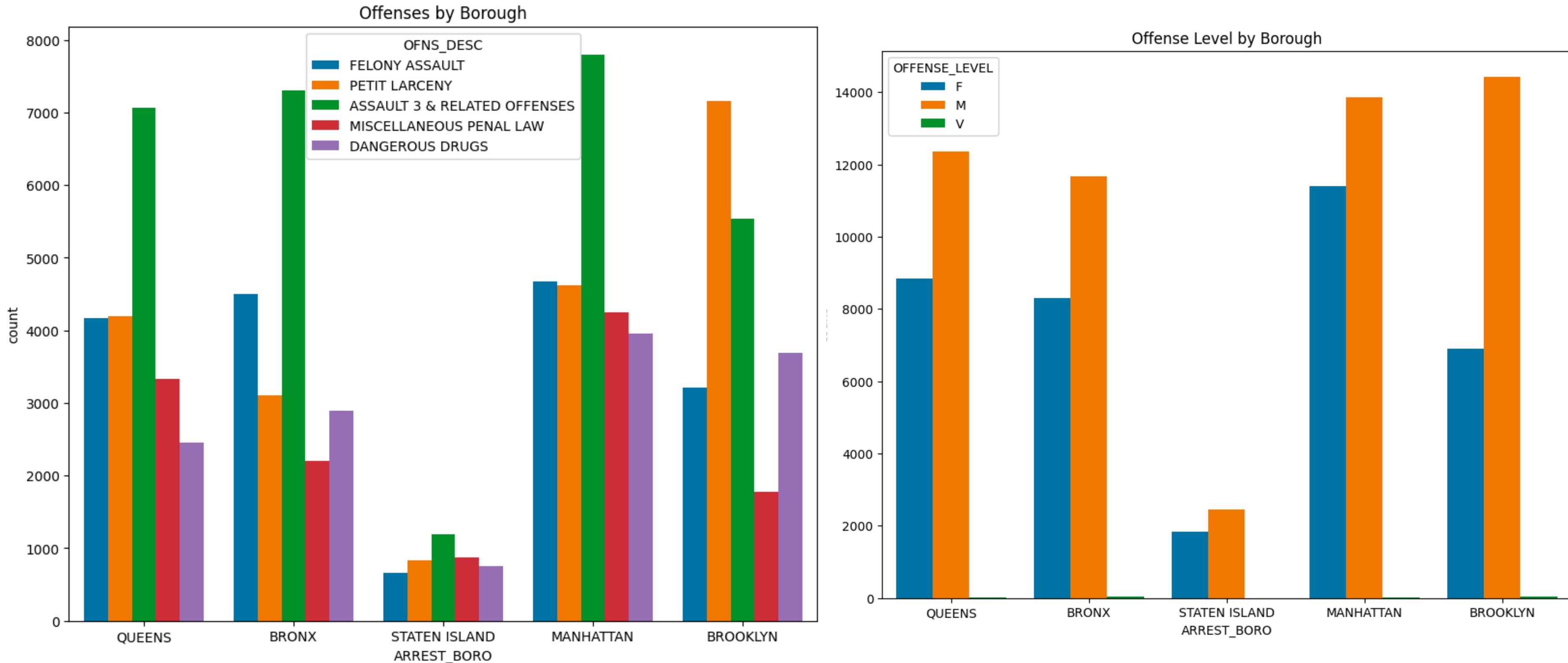
Column Correlation Heatmap



EDA

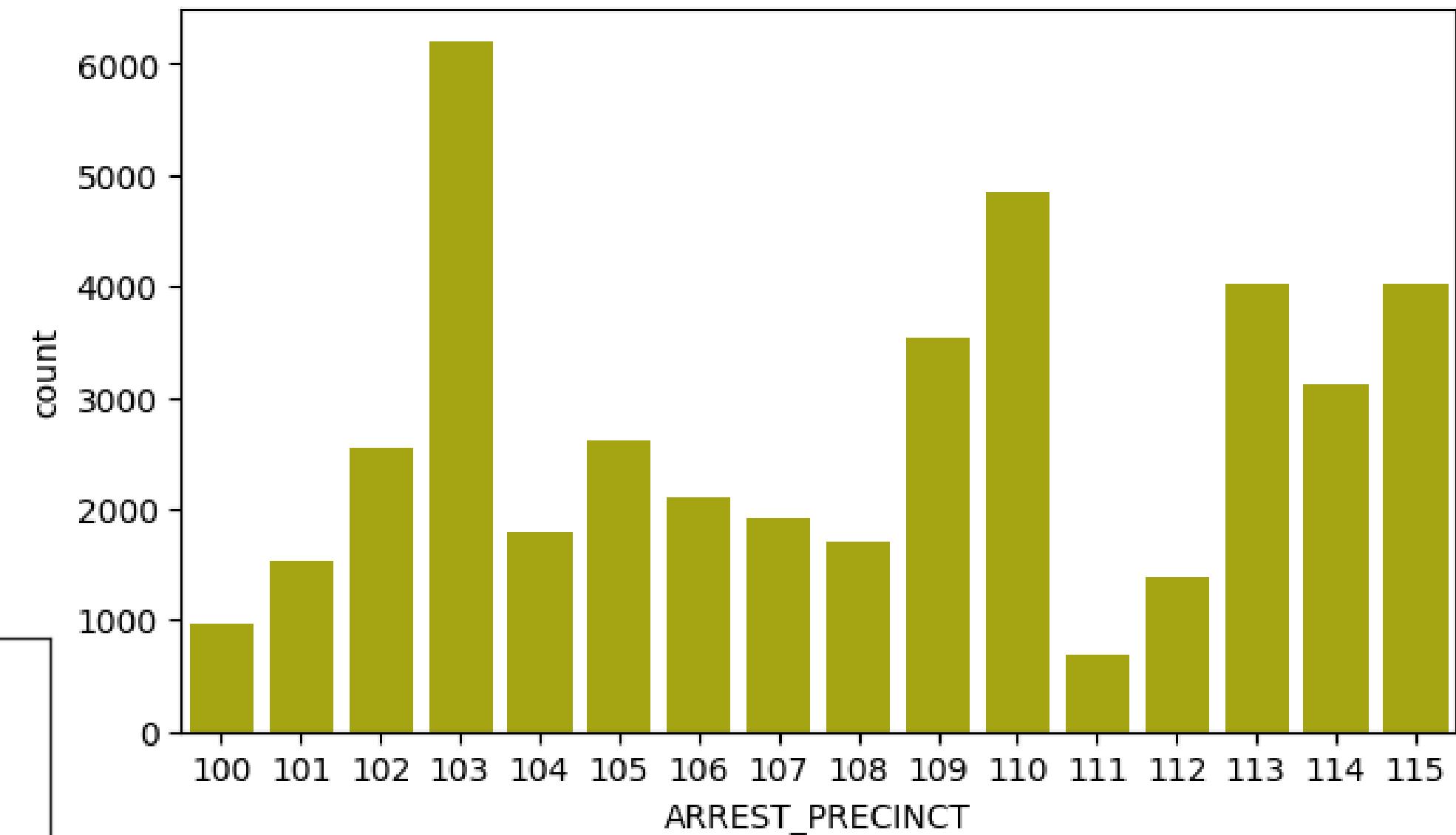
Heatmap for codes & locations



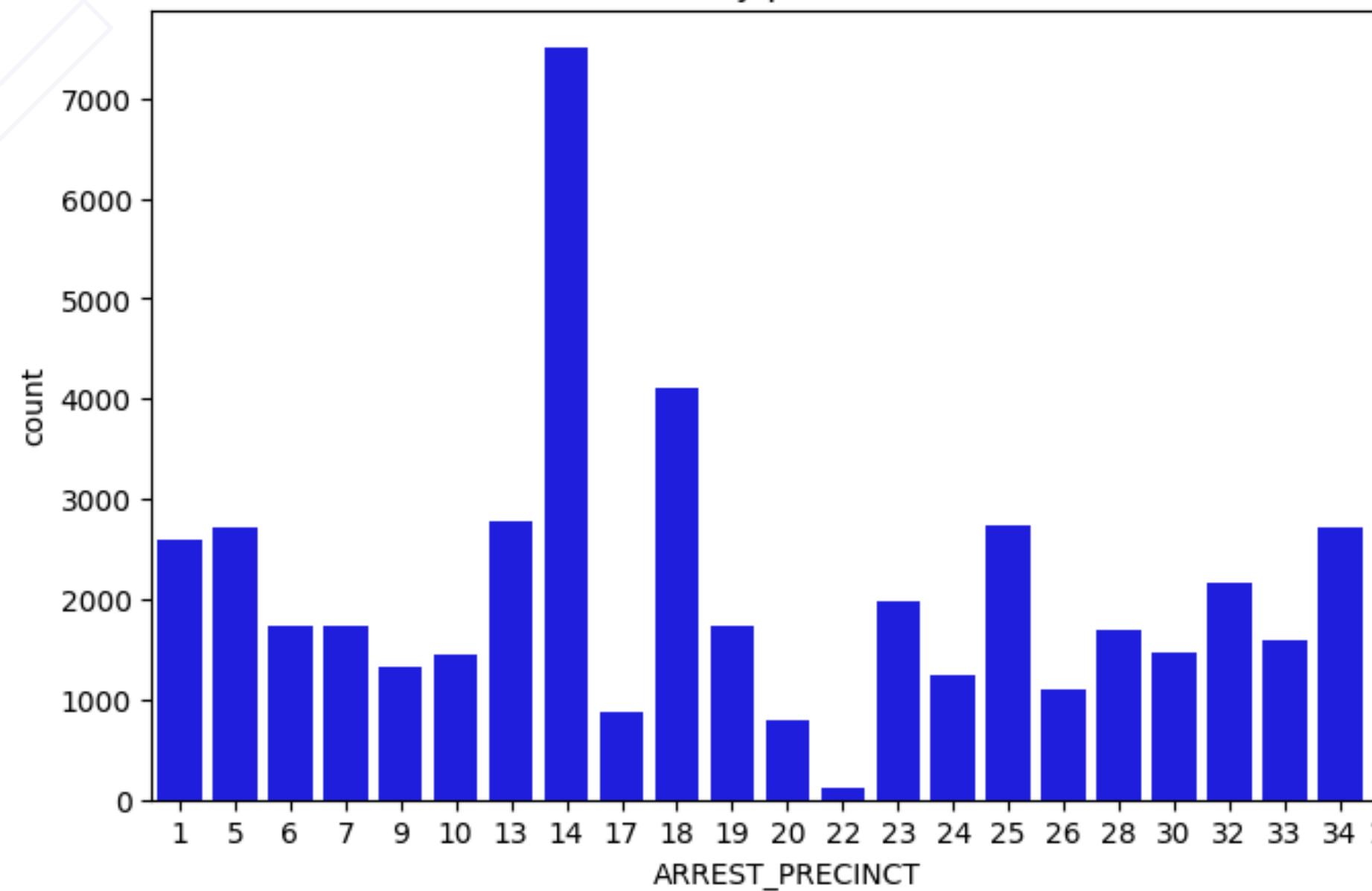


EDA

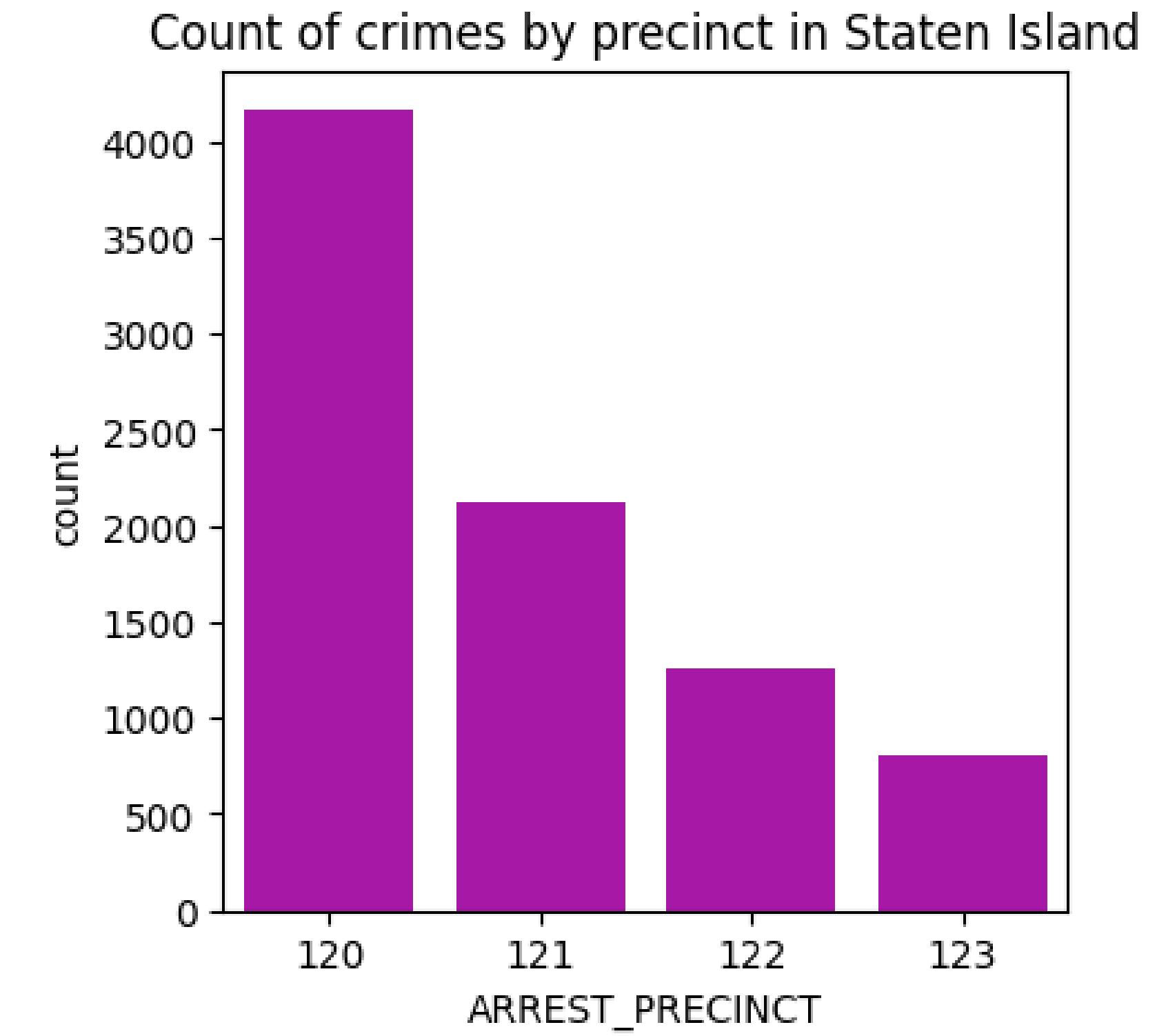
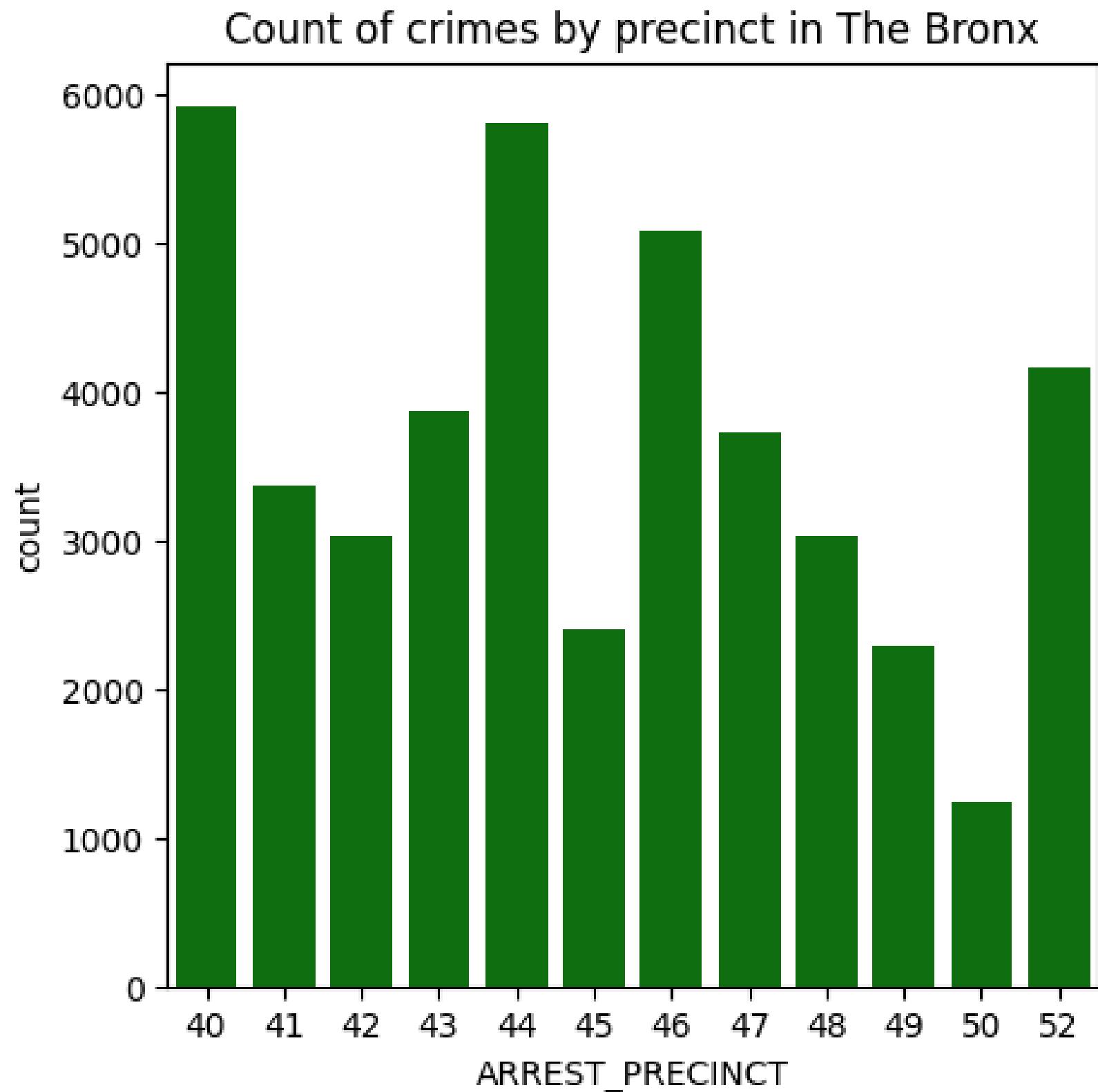
Count of crimes by precinct in Queens



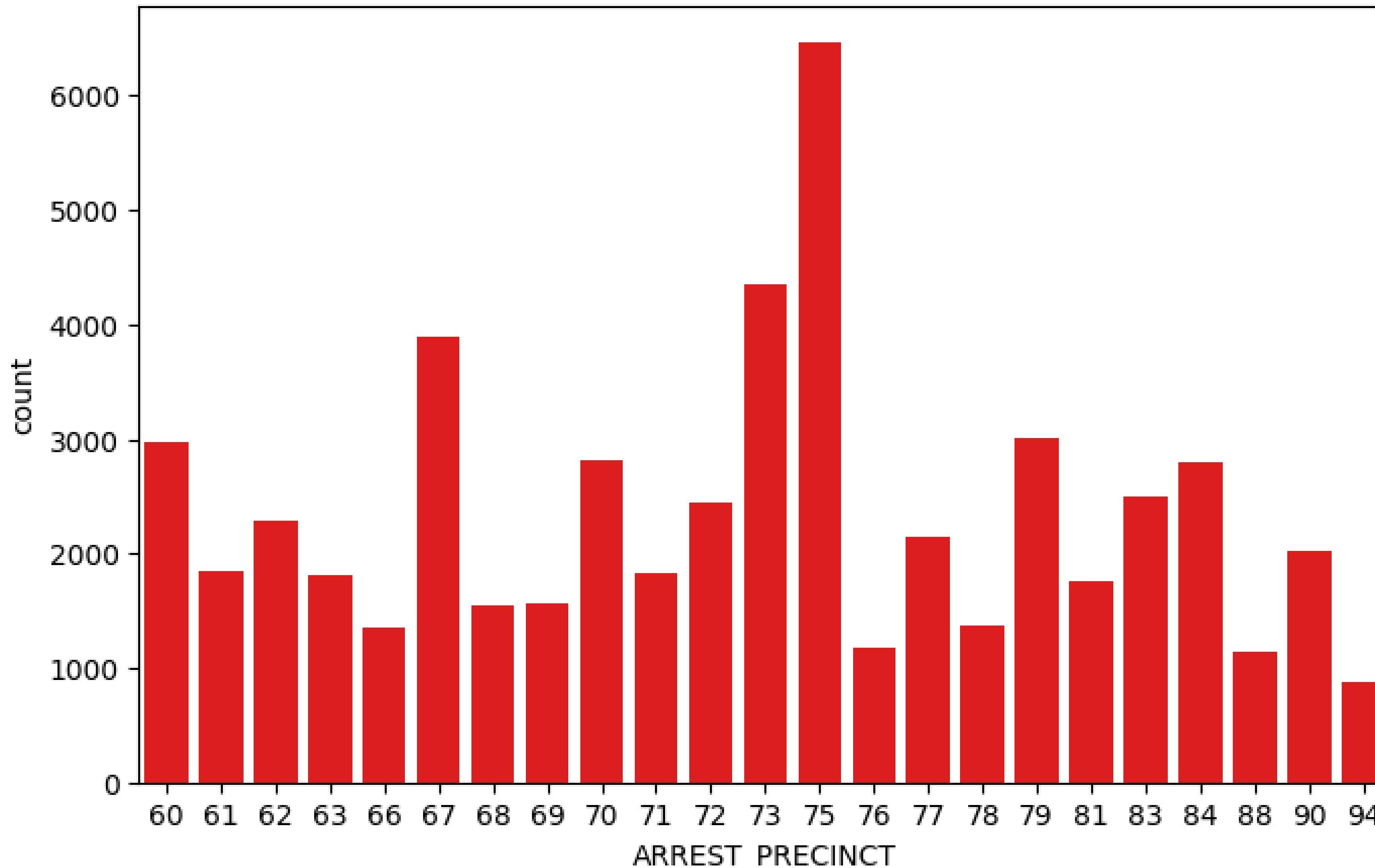
Count of crimes by precinct in Manhattan



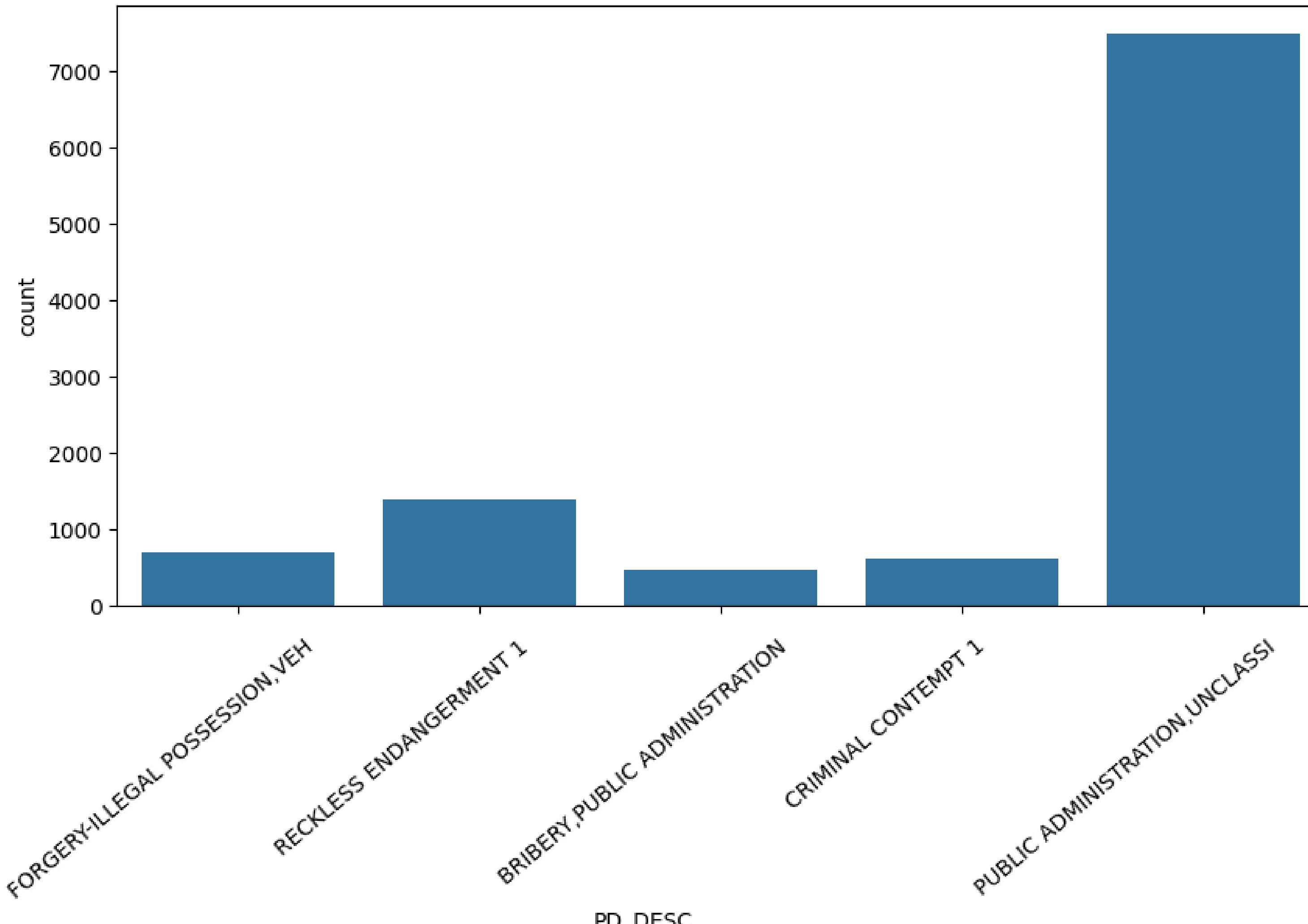
EDA



Count of crimes by precinct in Brooklyn

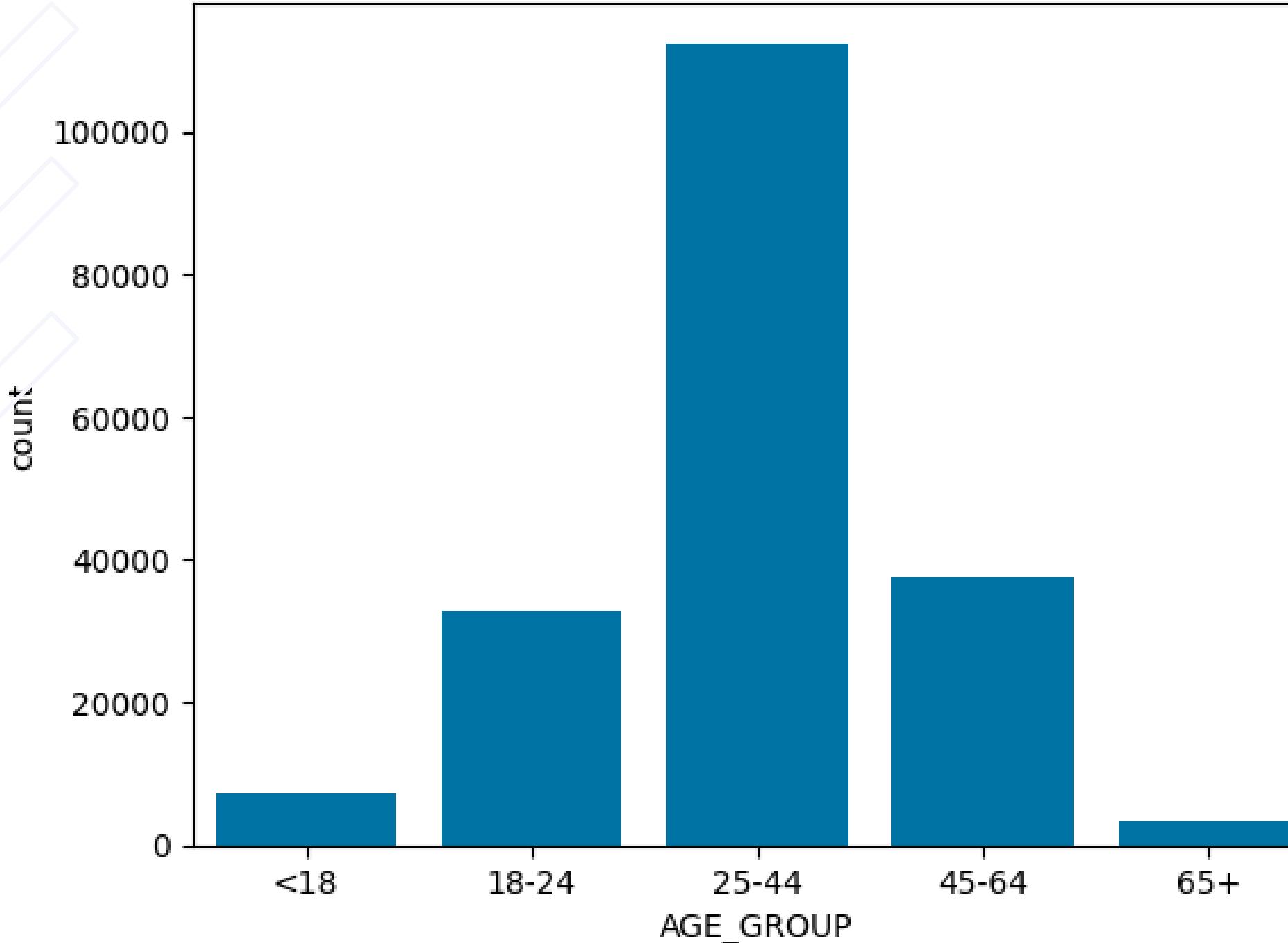


Count of top five crimes by their PD description that fall under the MISCELLANEOUS PENAL LAW category

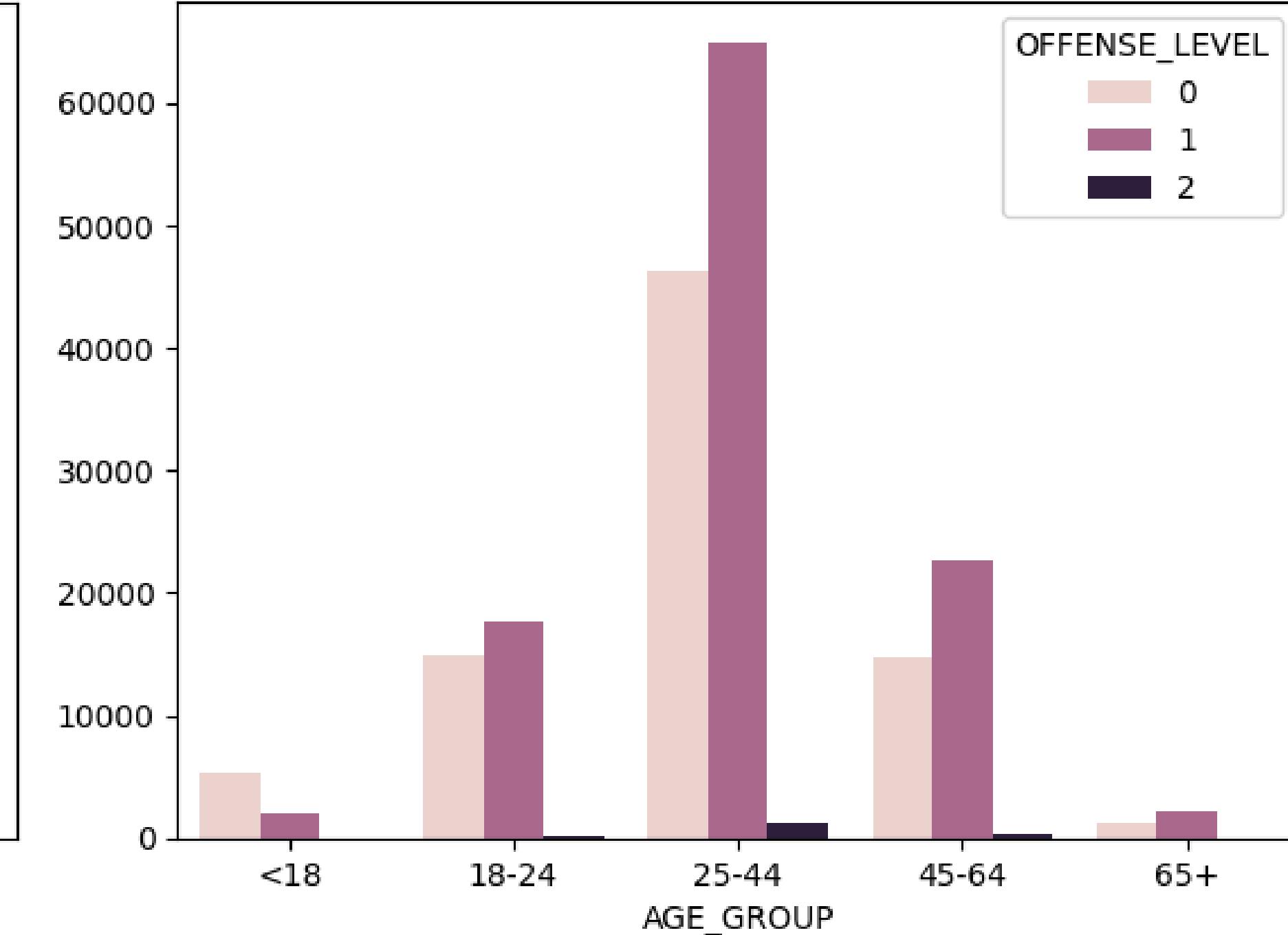


EDA

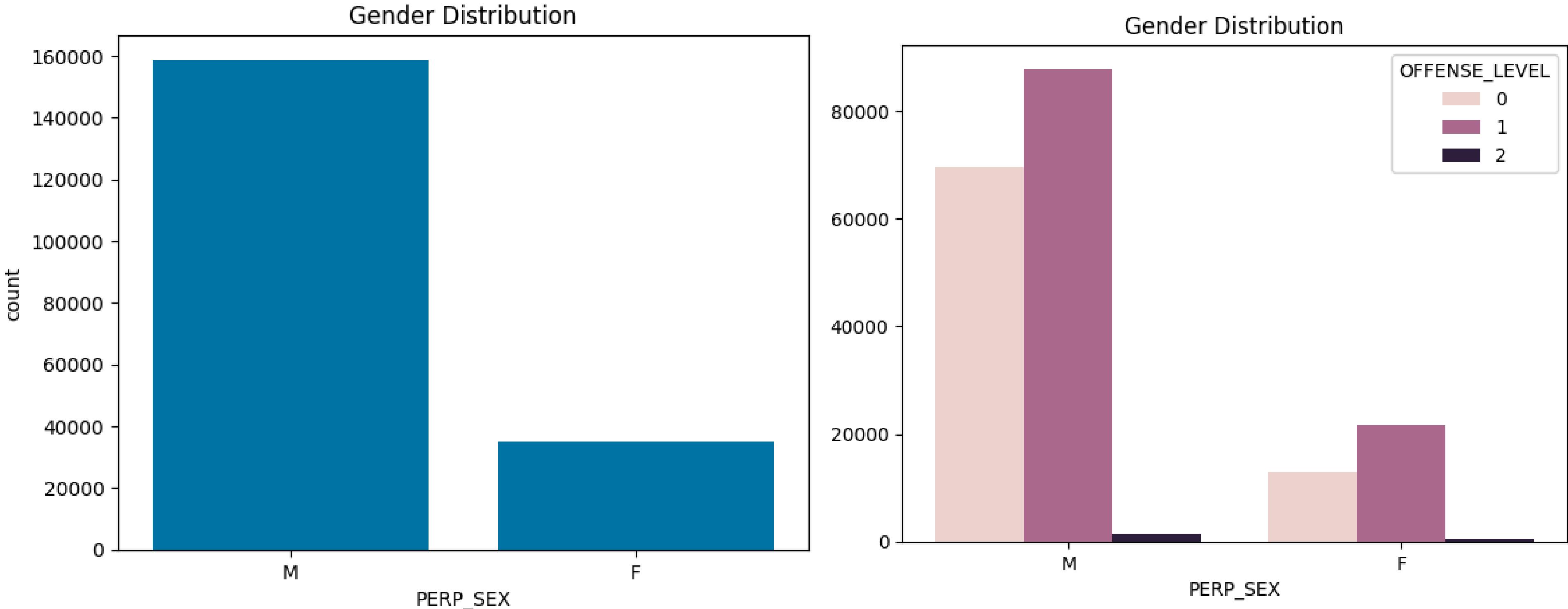
Age Group Distribution



Age Group Distribution

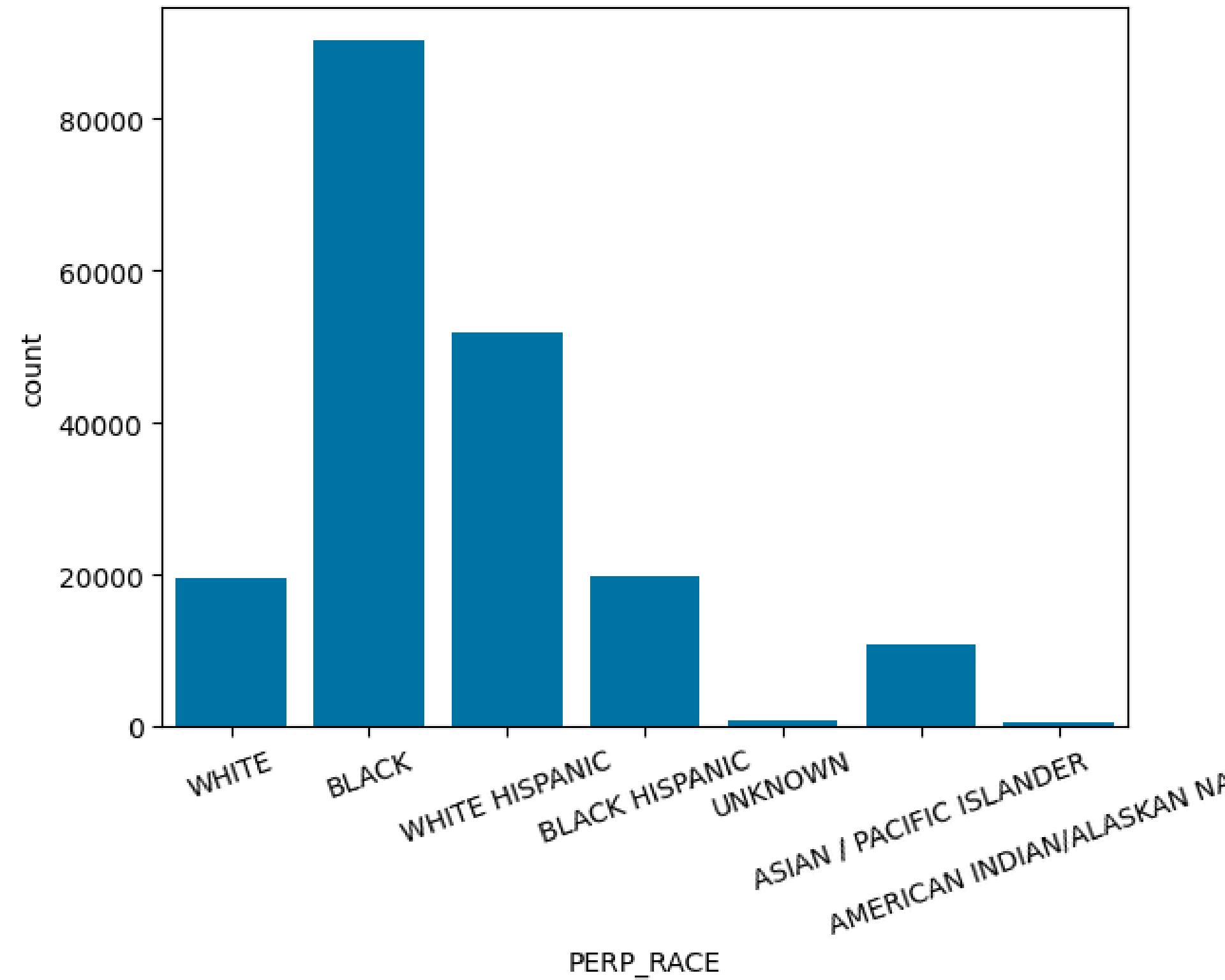


EDA

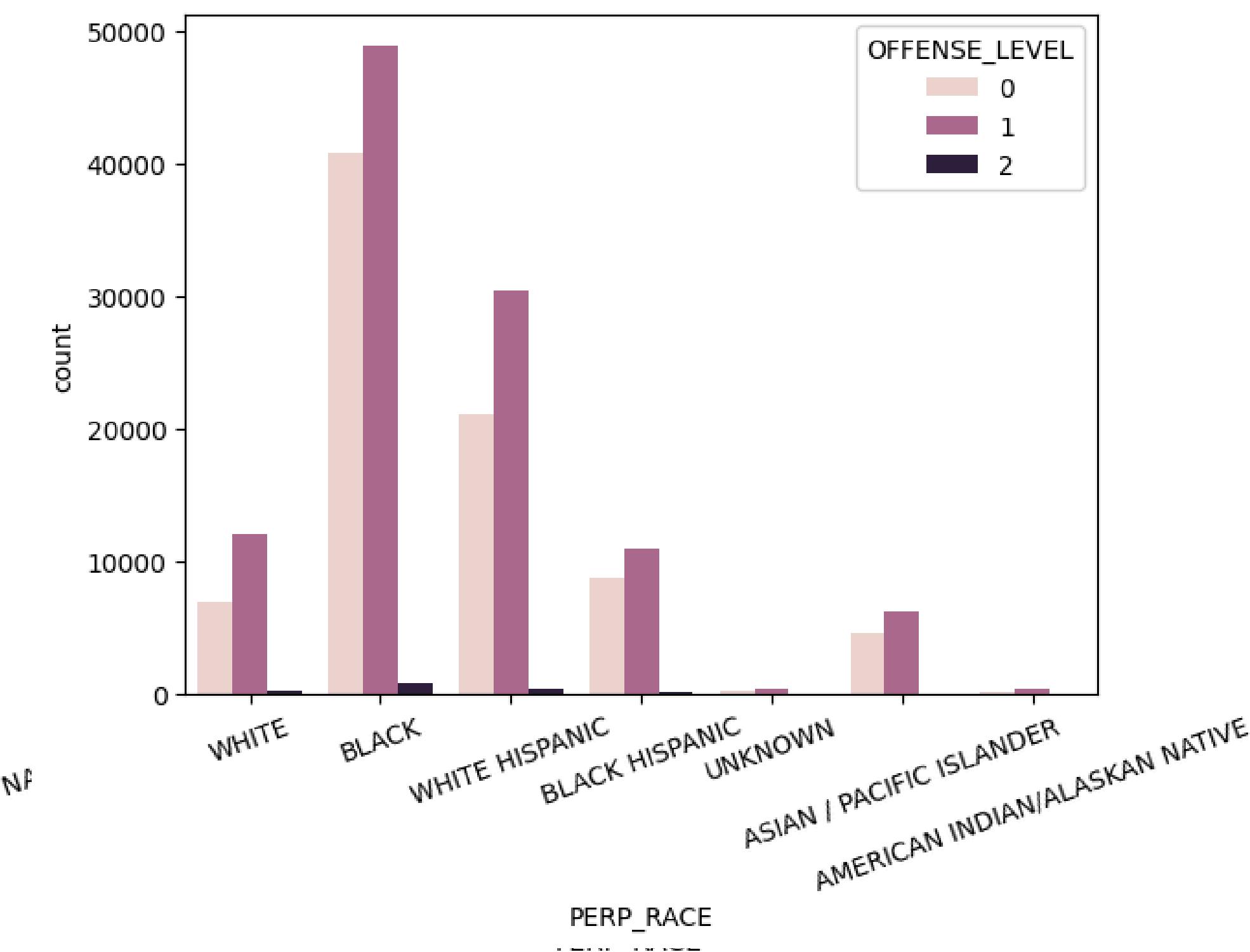


EDA

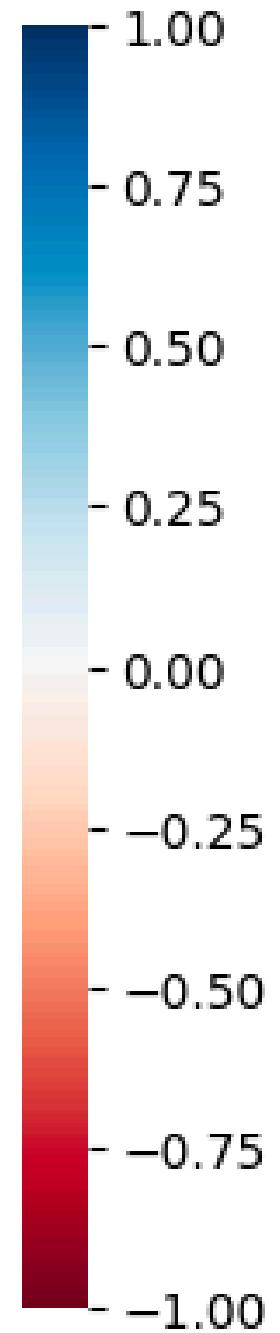
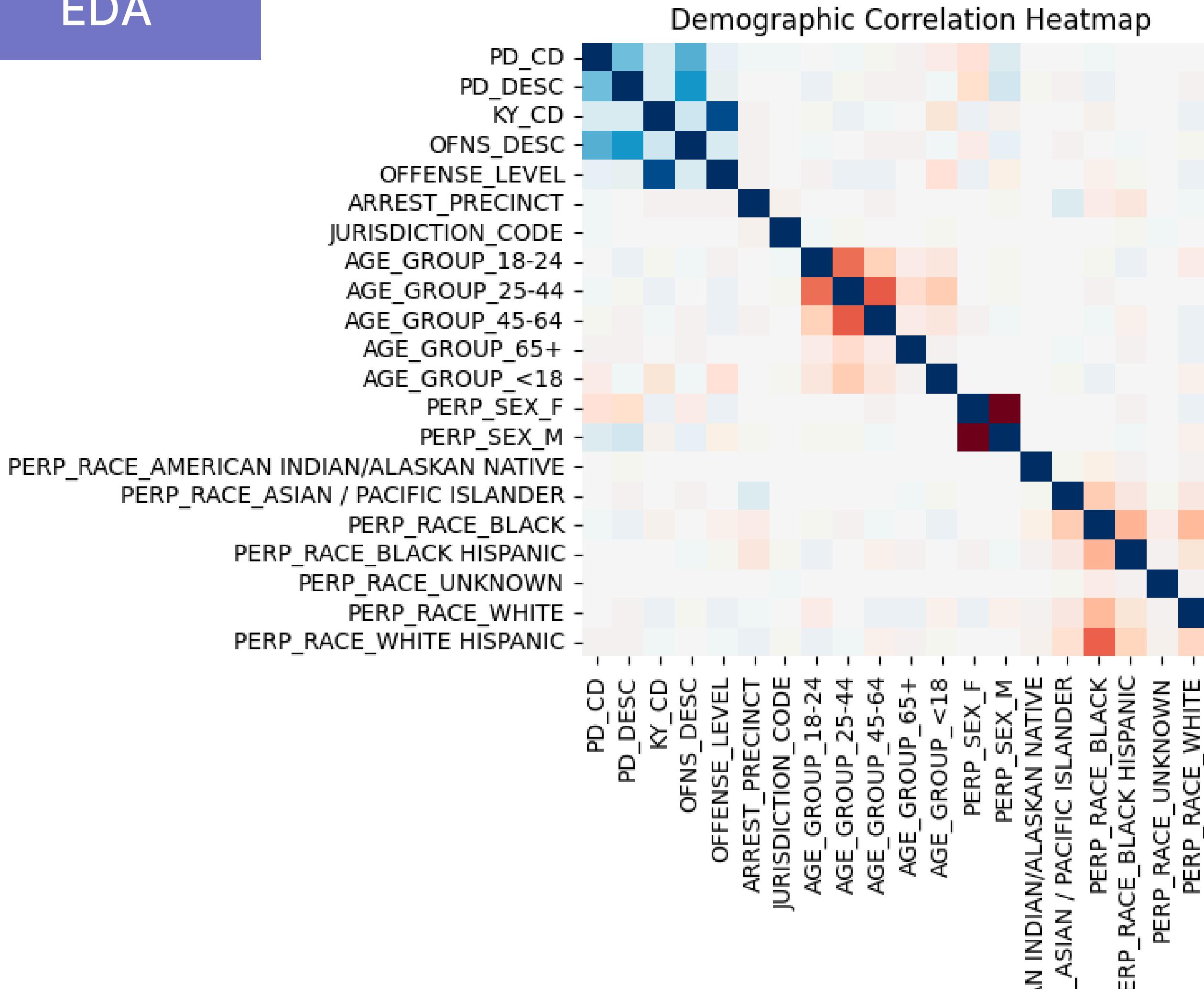
Race Distribution



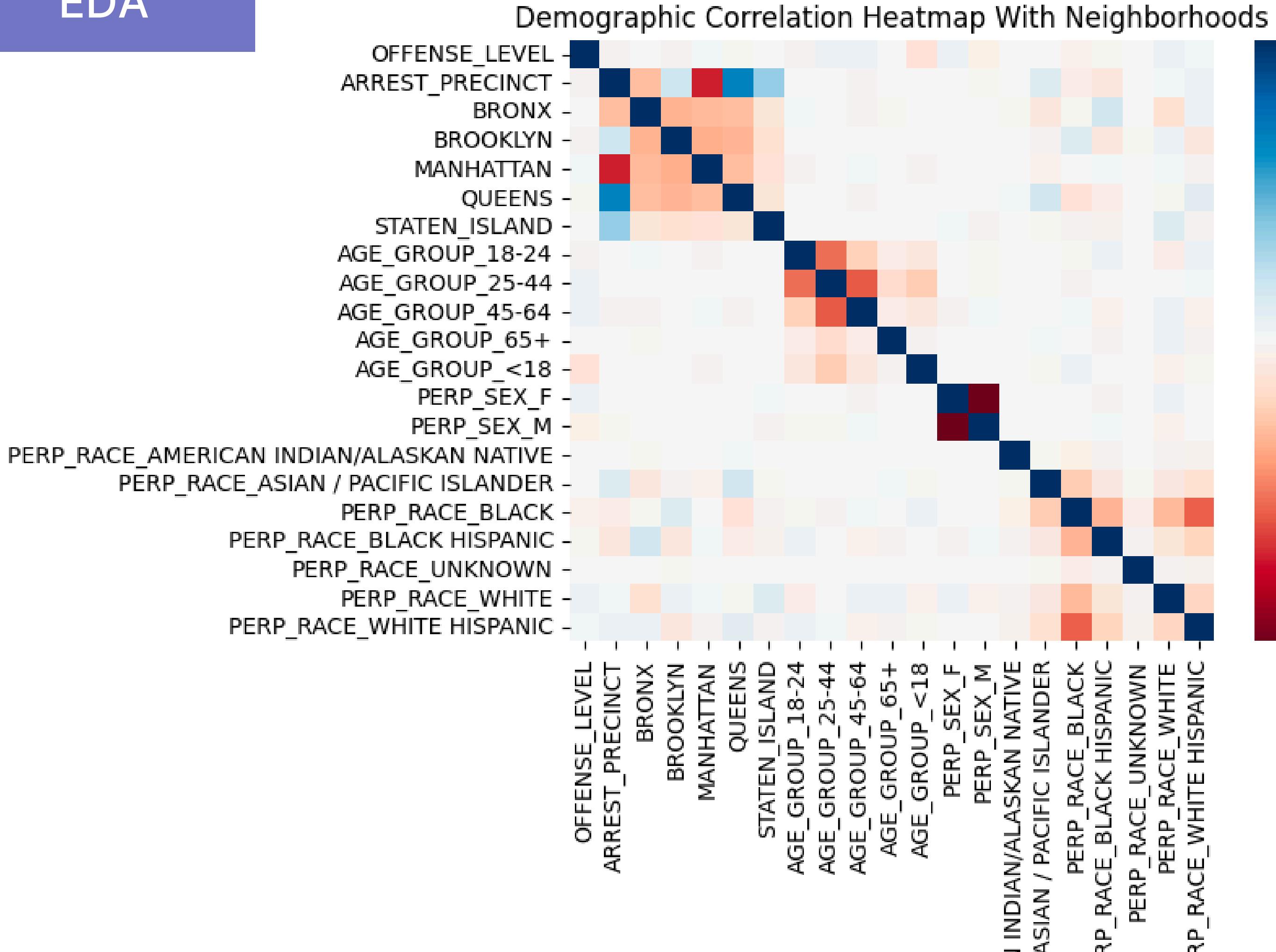
Race Distribution



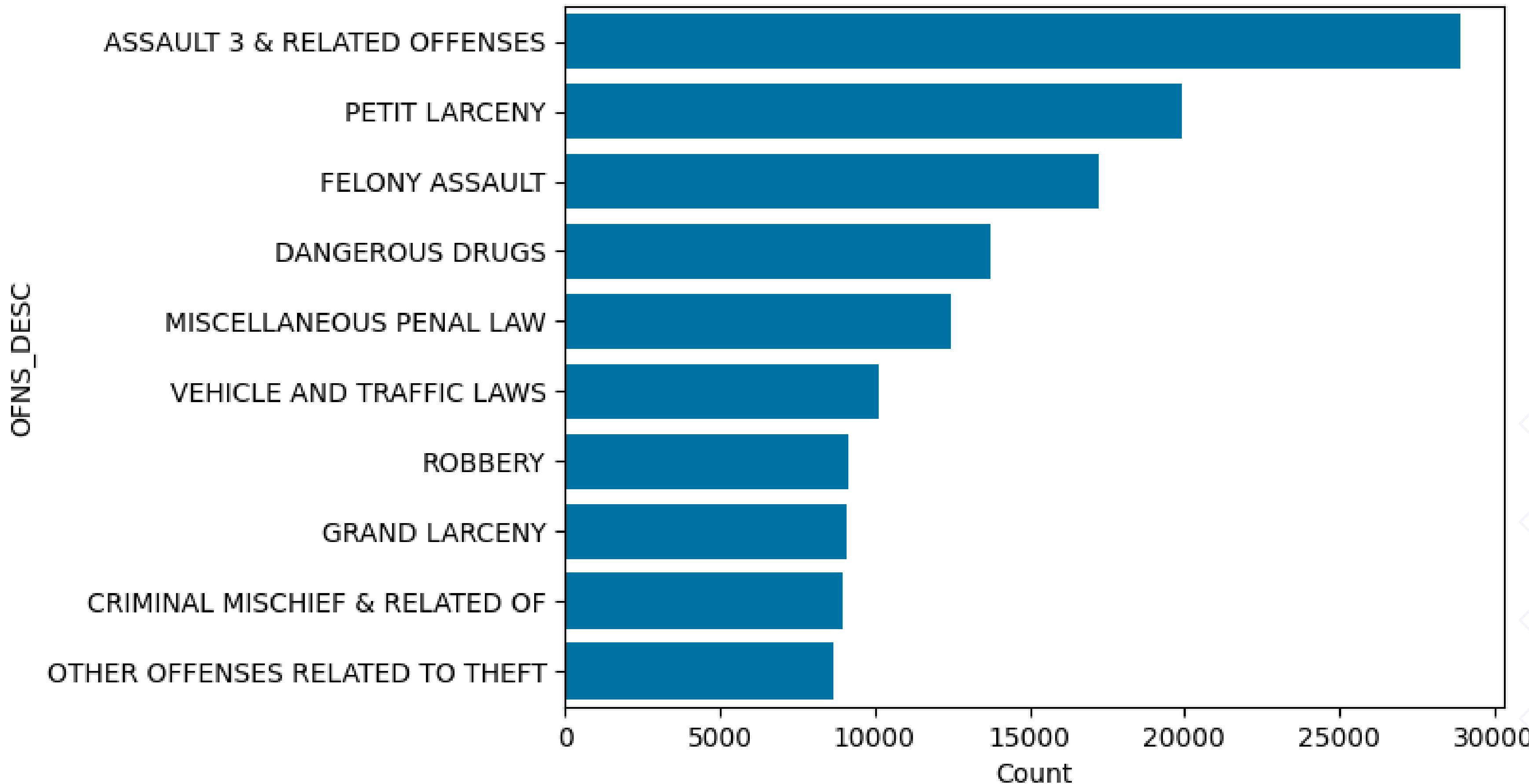
EDA



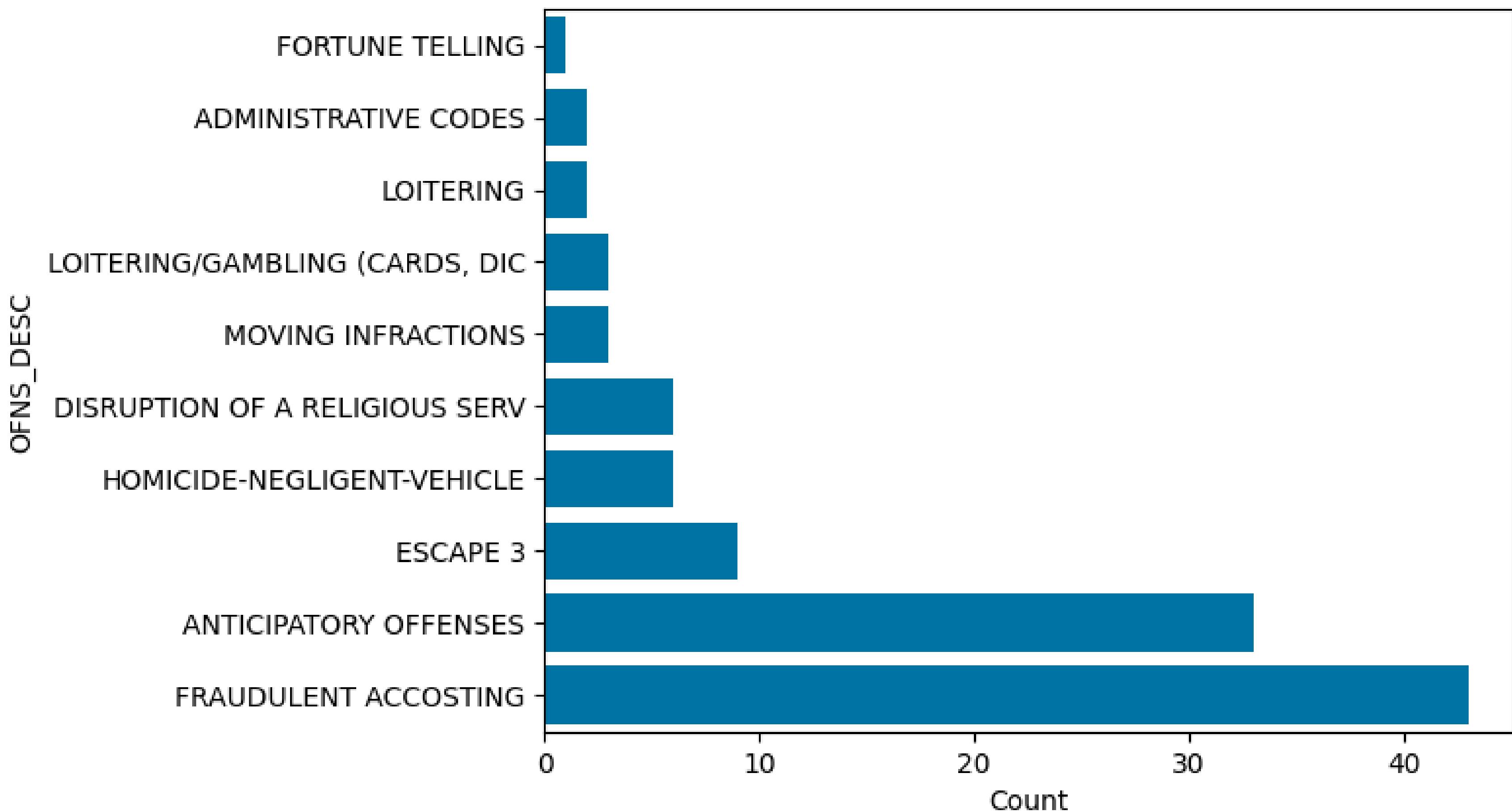
EDA



Top 10 Most Common Offenses

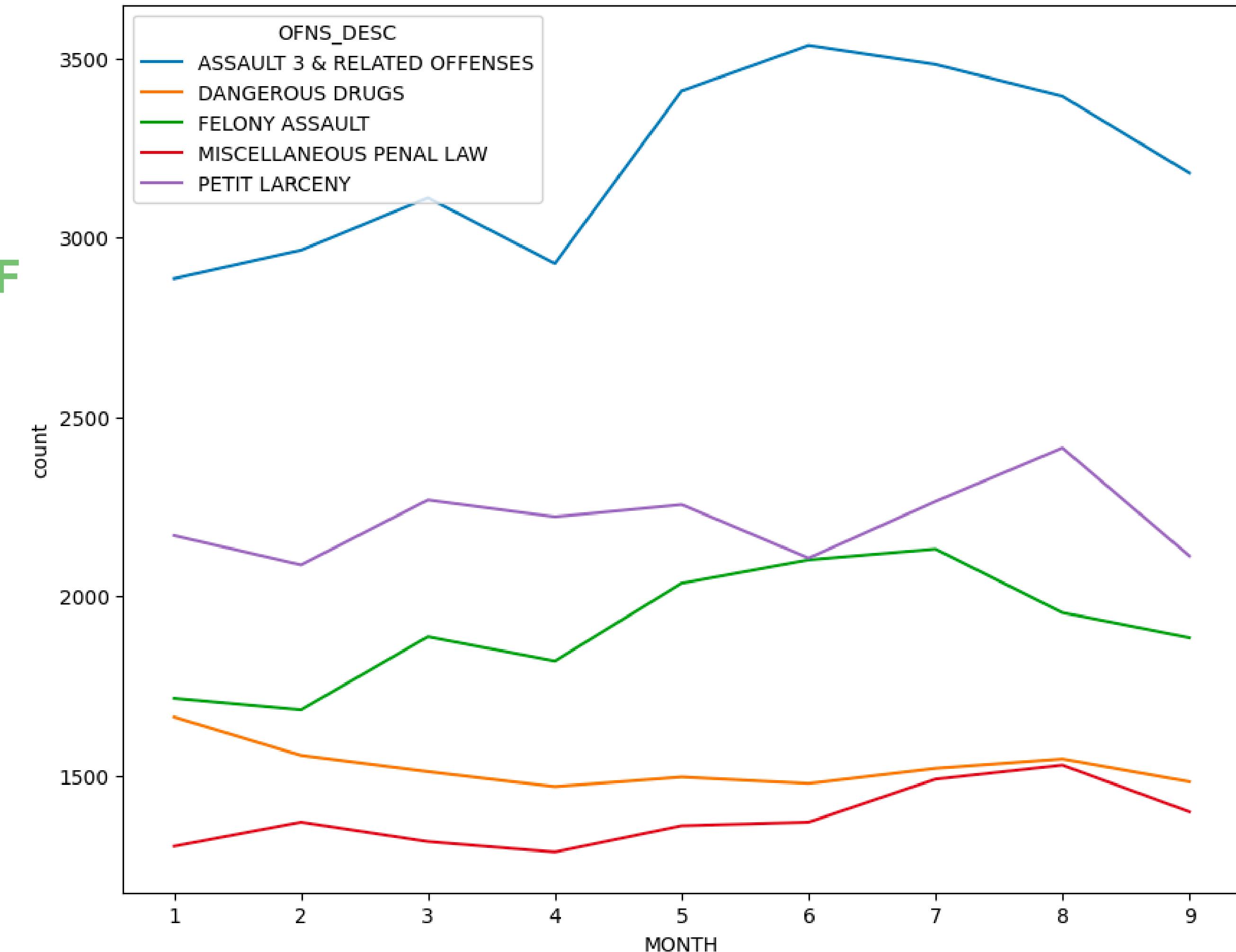


Least 10 Most Common Offenses

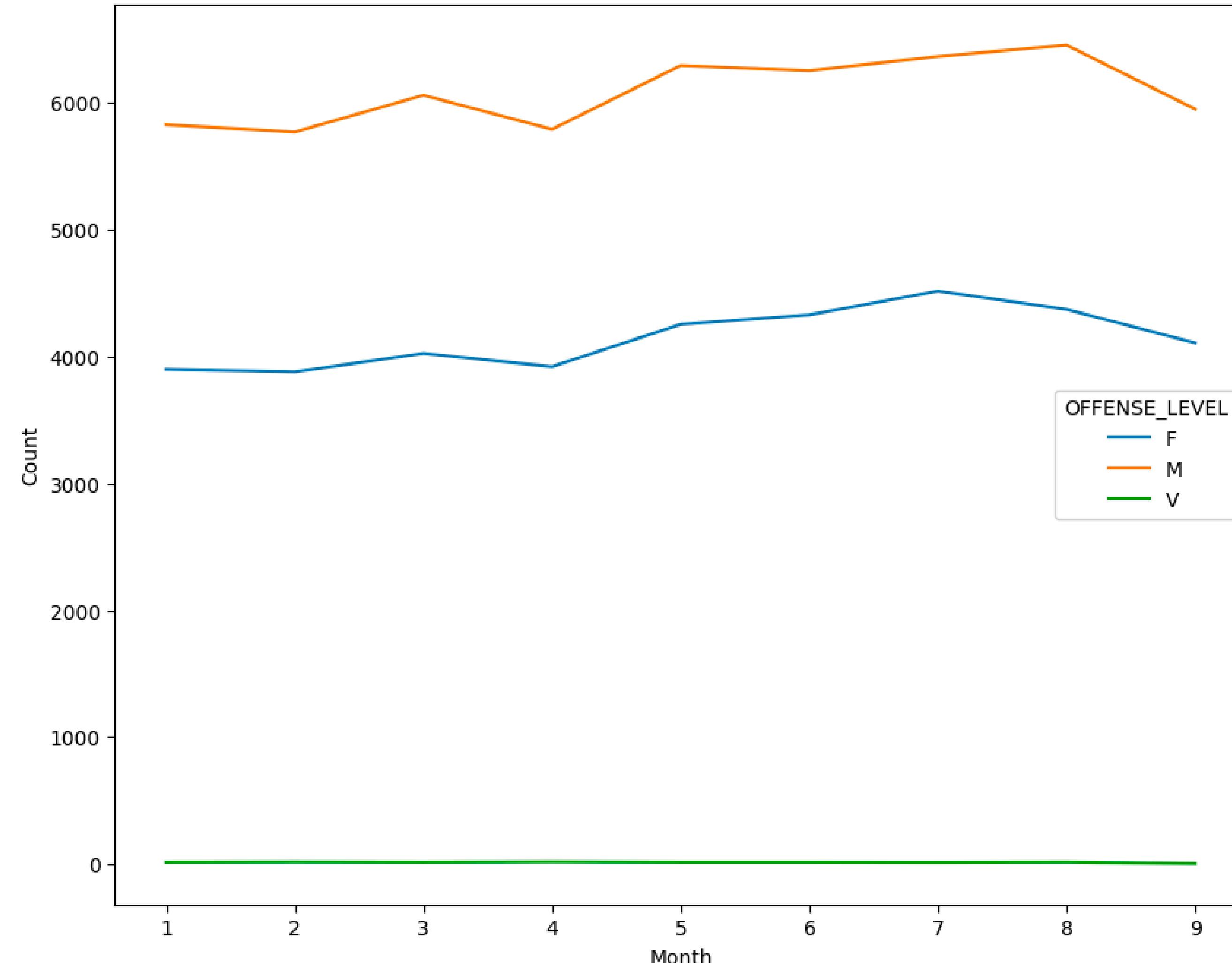


Occurrence of Top 5 Offenses by Month

FELONY ASSAULT - F
DRUGS - F
ASSAULT - M
LARCENY - M
MISC - F

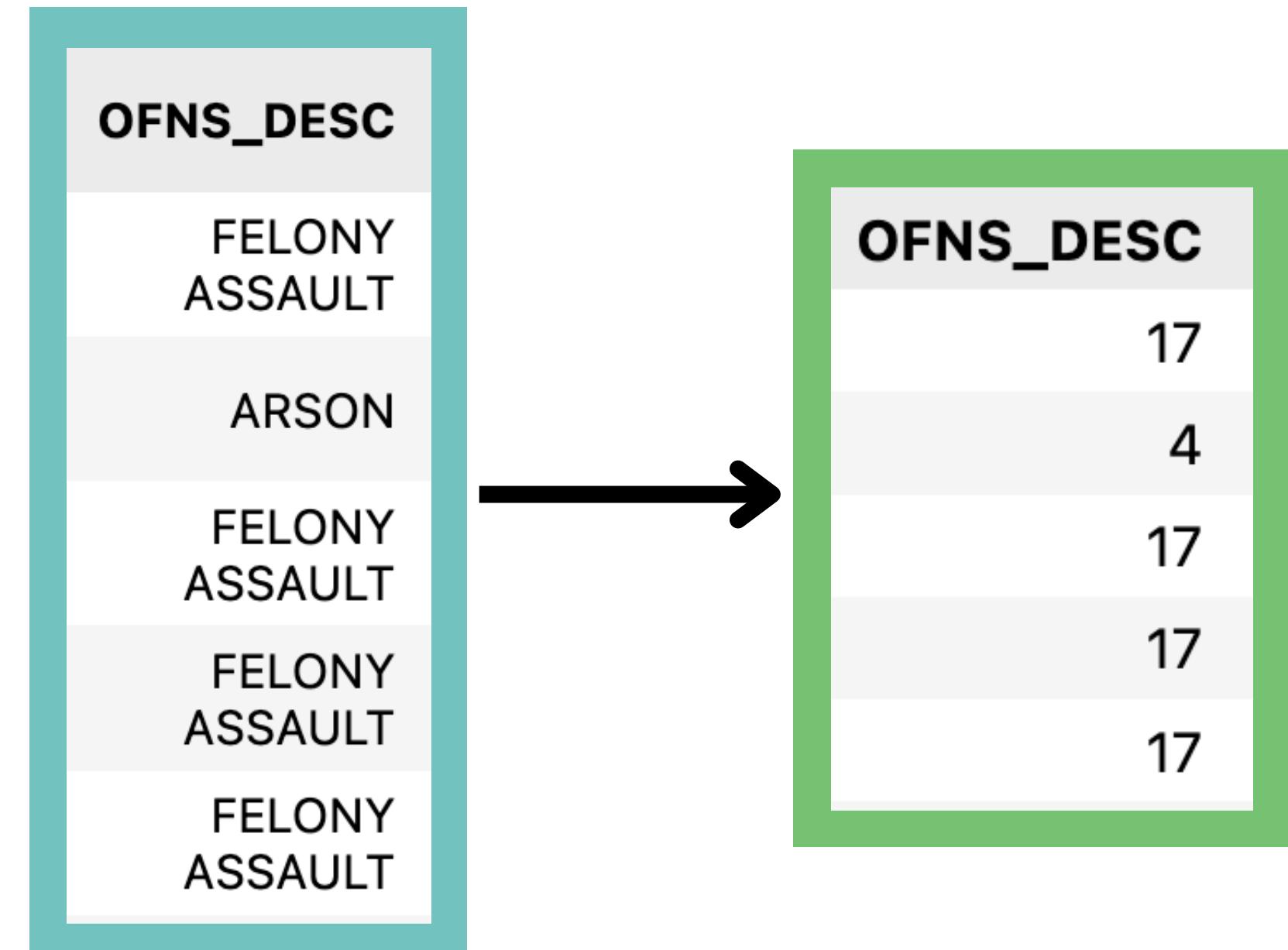


Occurrence of Offense Levels by Month



Data Cleaning

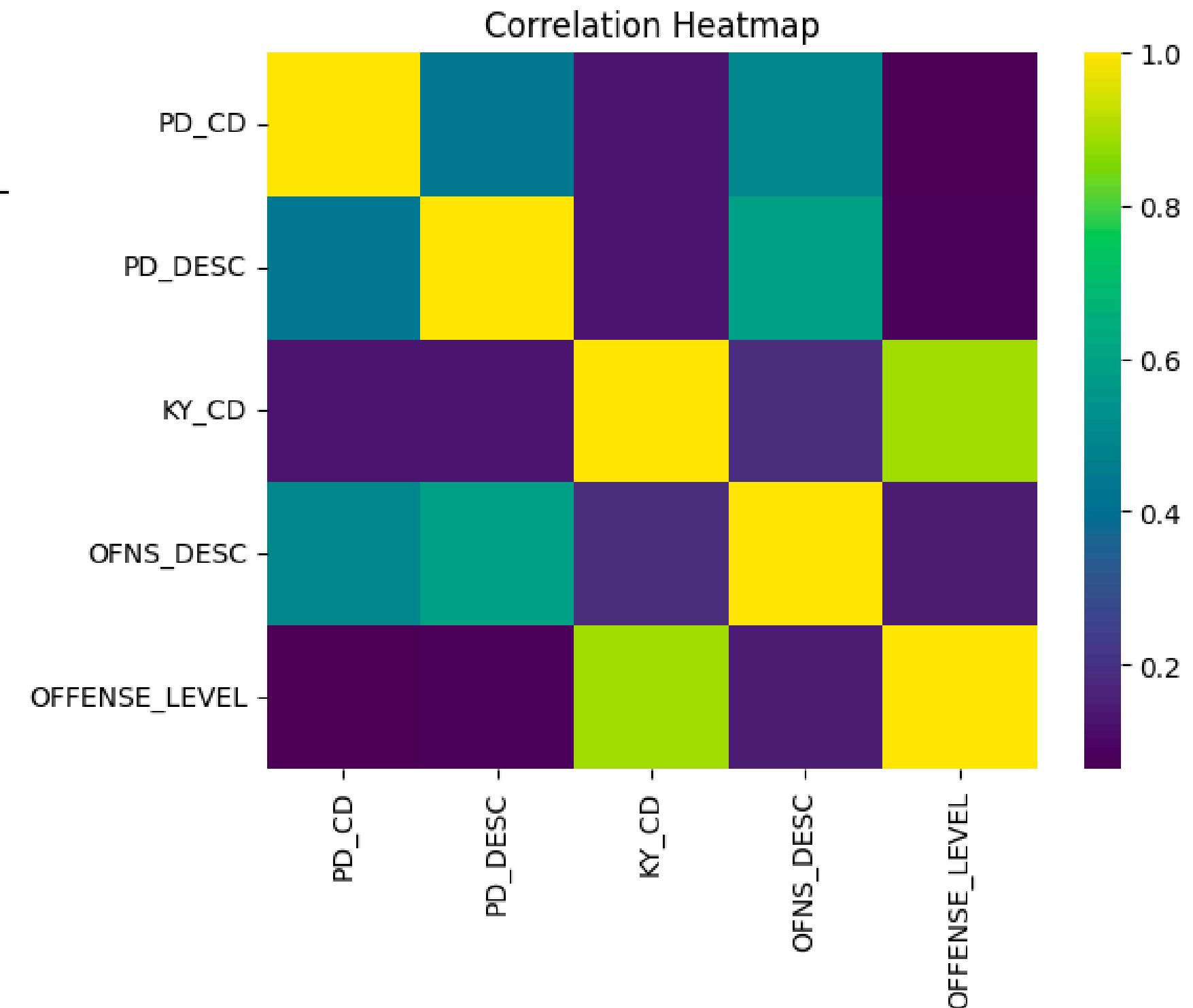
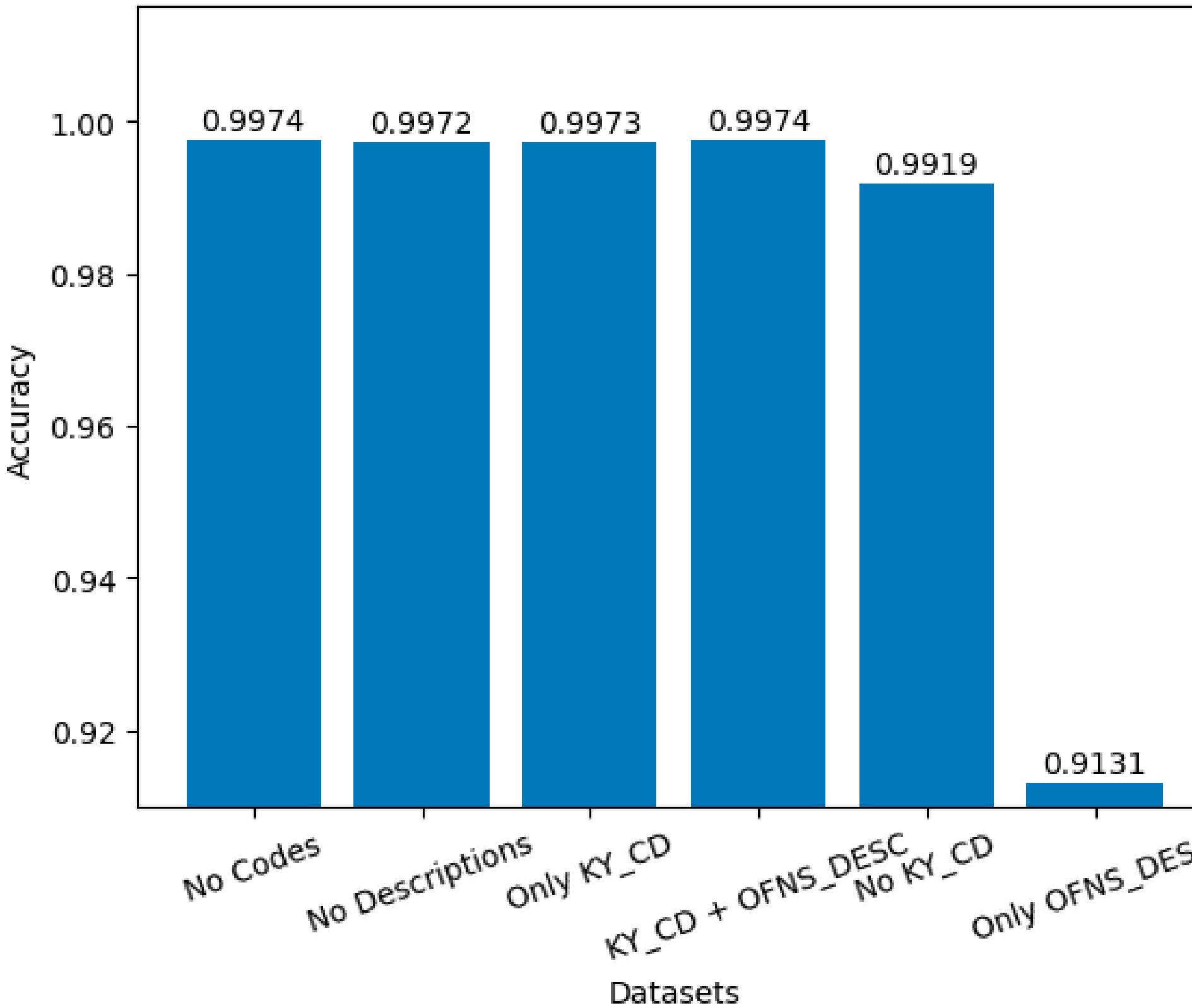
- Dropped null values
- Dropped non-important columns
- Made sure OFFENSE_LEVEL was only M, F, V
- Derived season and month column from date
- Converted code + defense columns to categorical
- Label encode the object columns
 - Label encoding converts categorical data into numerical data
 - might cause priority -- leading to misleading patterns or biases during training



MACHINE LEARNING MODELS

MODELS

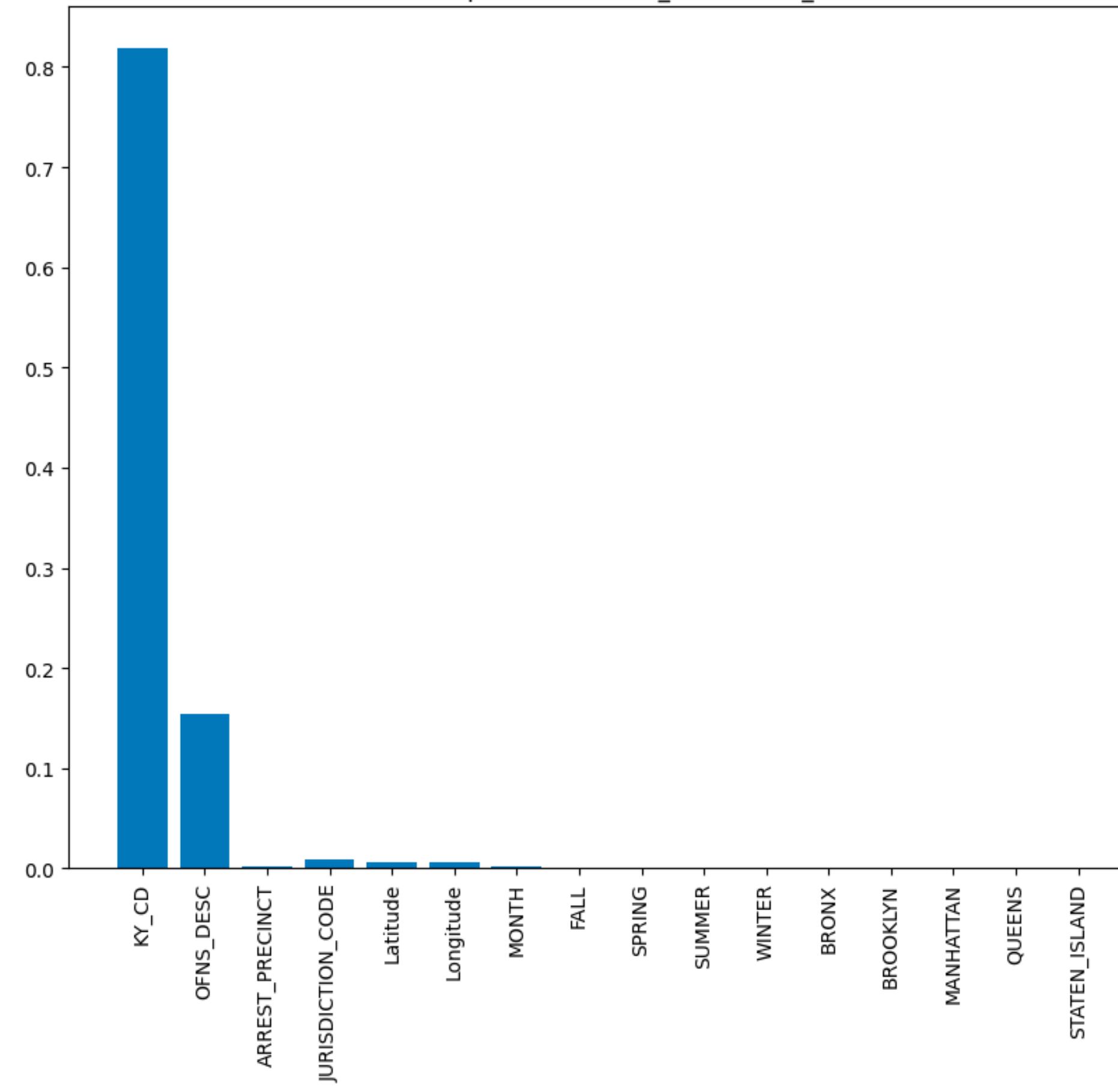
Accuracy of KNN Model on Different Datasets



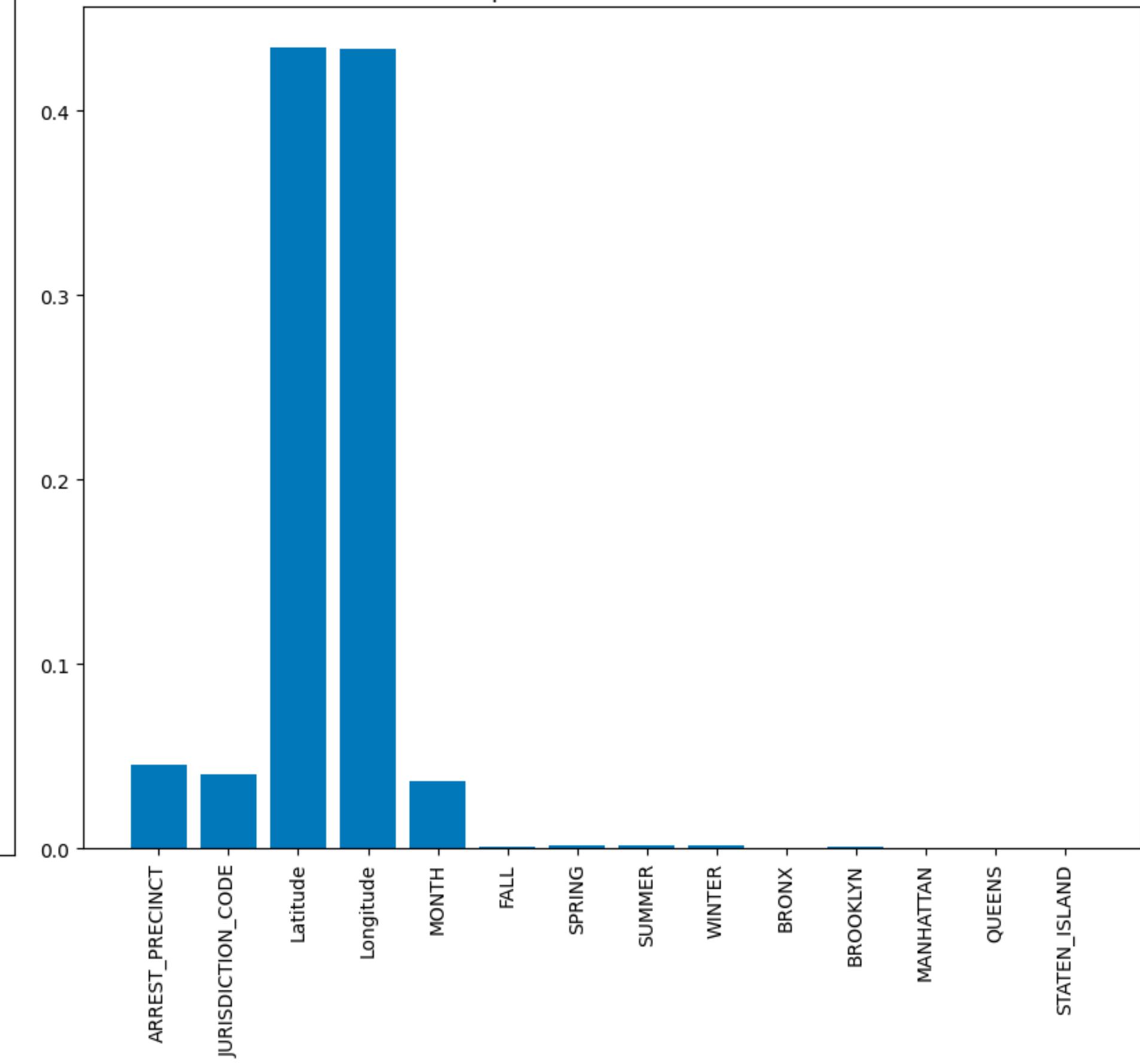
No Codes: 0.9756727441764371
No Descriptions: 0.9972367129797015
No PD_CD/DESC: 0.99728836320438
No KY_CD: 0.9918909147254791
KY_CD + OFNS_DESC: 0.9974174887660762
Only OFNS_DESC: 0.9130726718661226
No Codes/Descriptions: 0.6178399876039461

MODELS

Feature Importances for KY_CD + OFNS_DESC



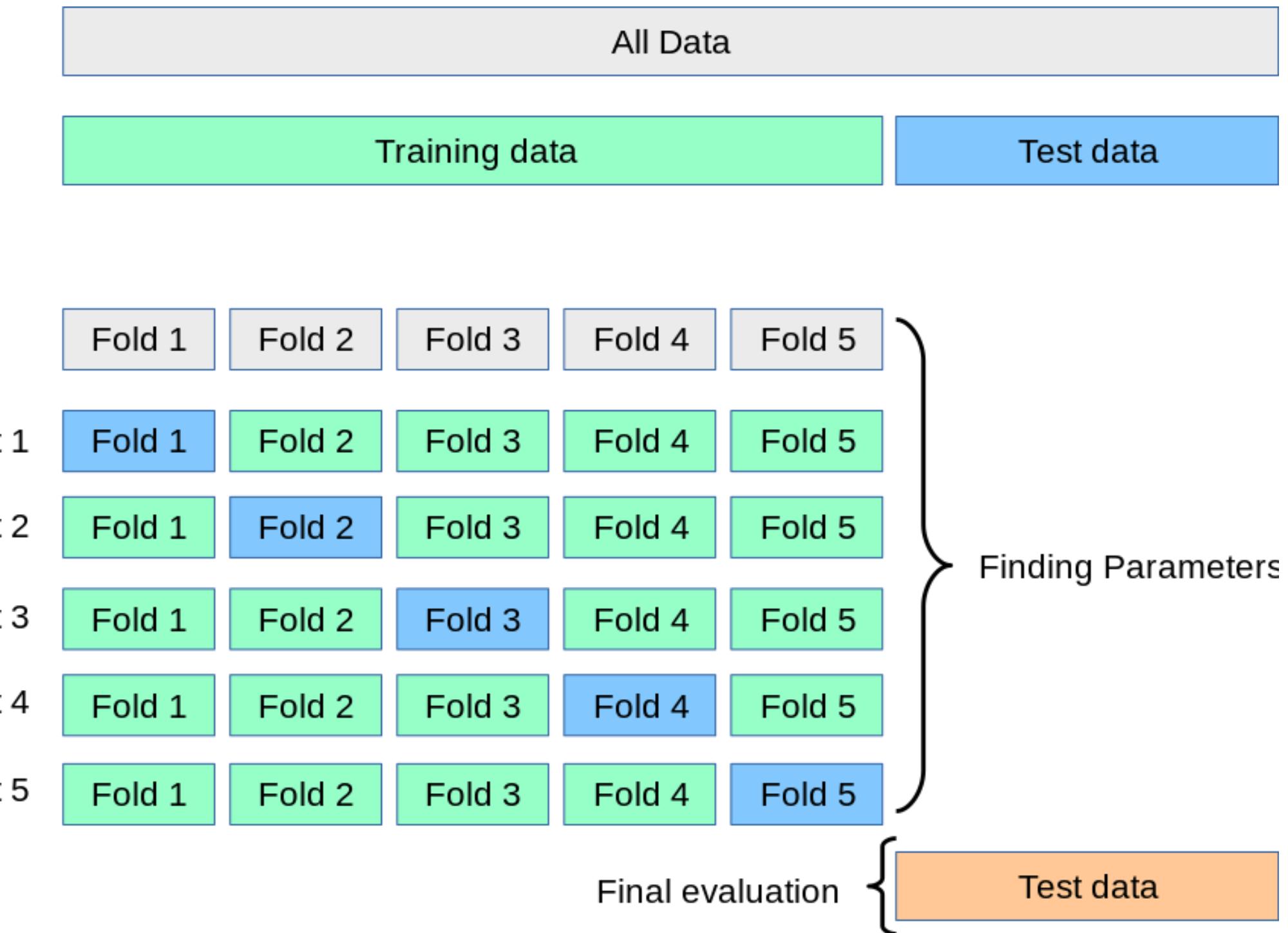
Feature Importances For no Desc or Codes



Cross Validation:

Model Evaluation Technique

- Resamples to train model on different iterations
- Compare models
- Avoid Overfitting
- Assess generalization to unseen data
- splitting your dataset into K subsets
- Not training the same model



Grid Search:

Hyperparameter Tuning Technique

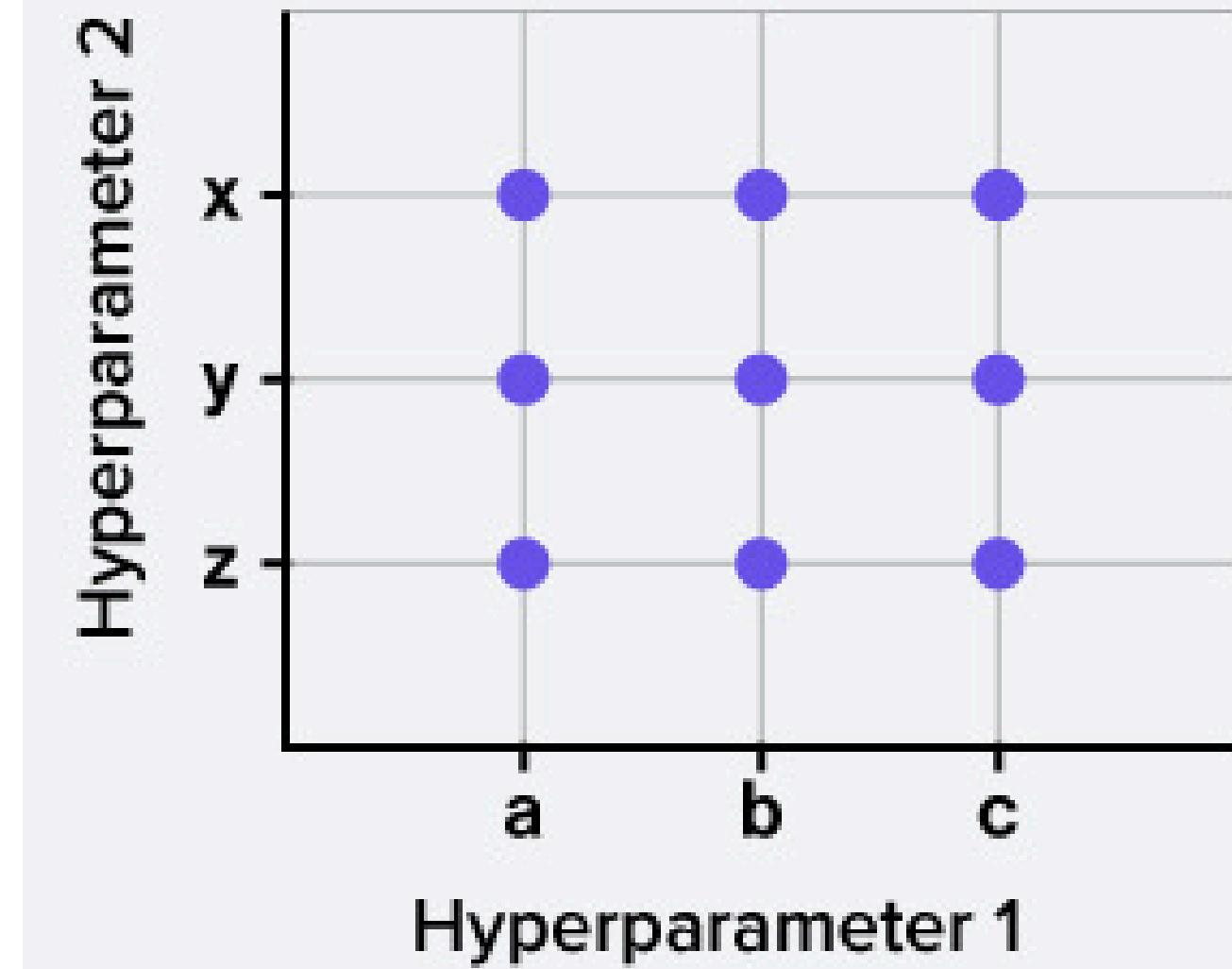
- find the best combo of hyperparameters for a given model
- GridSearchCV()
 - FIRST optimize parameters with Grid Search, SECOND evaluate performance of that model with cross val

Grid Search

Hyperparameter_One = [a, b, c]

Hyperparameter_Two = [x, y, z]

Hyperparameter_X = [i, j, k]



KNN

	precision	recall	f1-score	support
0	0.902744	0.917254	0.909941	24823
1	0.935837	0.927600	0.931700	32721
2	0.848131	0.673469	0.750776	539
accuracy	0.920820	0.920820	0.920820	0
macro avg	0.895570	0.839441	0.864139	58083
weighted avg	0.920880	0.920820	0.920722	58083

Without Demographics

Best KNN Model: {'metric': 'manhattan',
'n_neighbors': 7, 'weights': 'distance'}

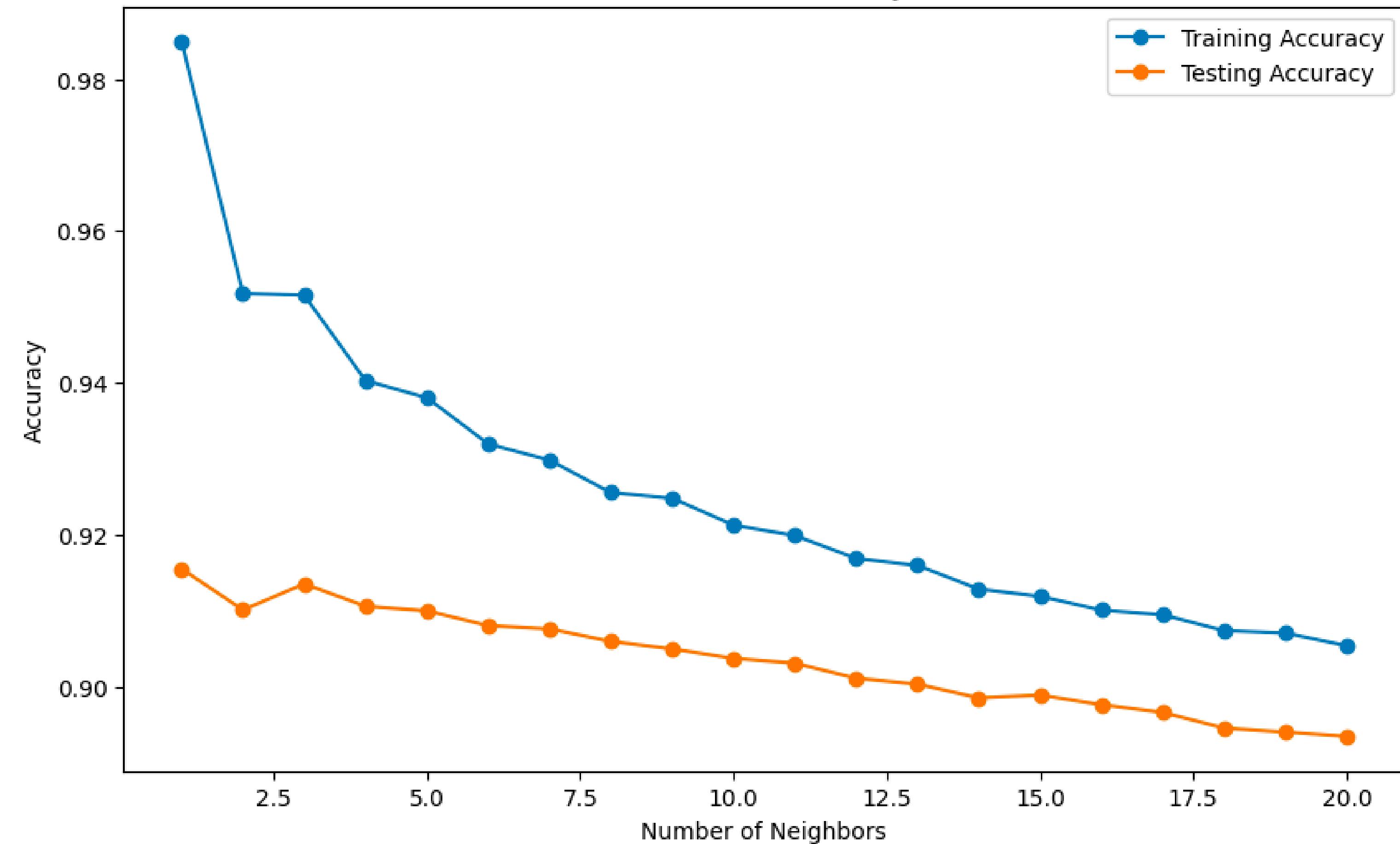
	precision	recall	f1-score	support
0	0.888982	0.894533	0.891749	24823
1	0.915780	0.918187	0.916982	32721
2	0.865772	0.478664	0.616487	539
accuracy	0.903999	0.903999	0.903999	0
macro avg	0.890178	0.763795	0.808406	58083
weighted avg	0.903863	0.903999	0.903410	58083

With Demographics

Best KNN Model: {'metric': 'manhattan',
'n_neighbors': 7, 'weights': 'distance'}

KNN

KNN Model Accuracy



DECISION TREES (CART)

	precision	recall	f1-score	support
0	0.893831	0.945575	0.918975	24823
1	0.957464	0.916995	0.936793	32721
2	0.944330	0.849722	0.894531	539
accuracy	0.928585	0.928585	0.928585	0
macro avg	0.931875	0.904097	0.916766	58083
weighted avg	0.930147	0.928585	0.928786	58083

Without Demographics

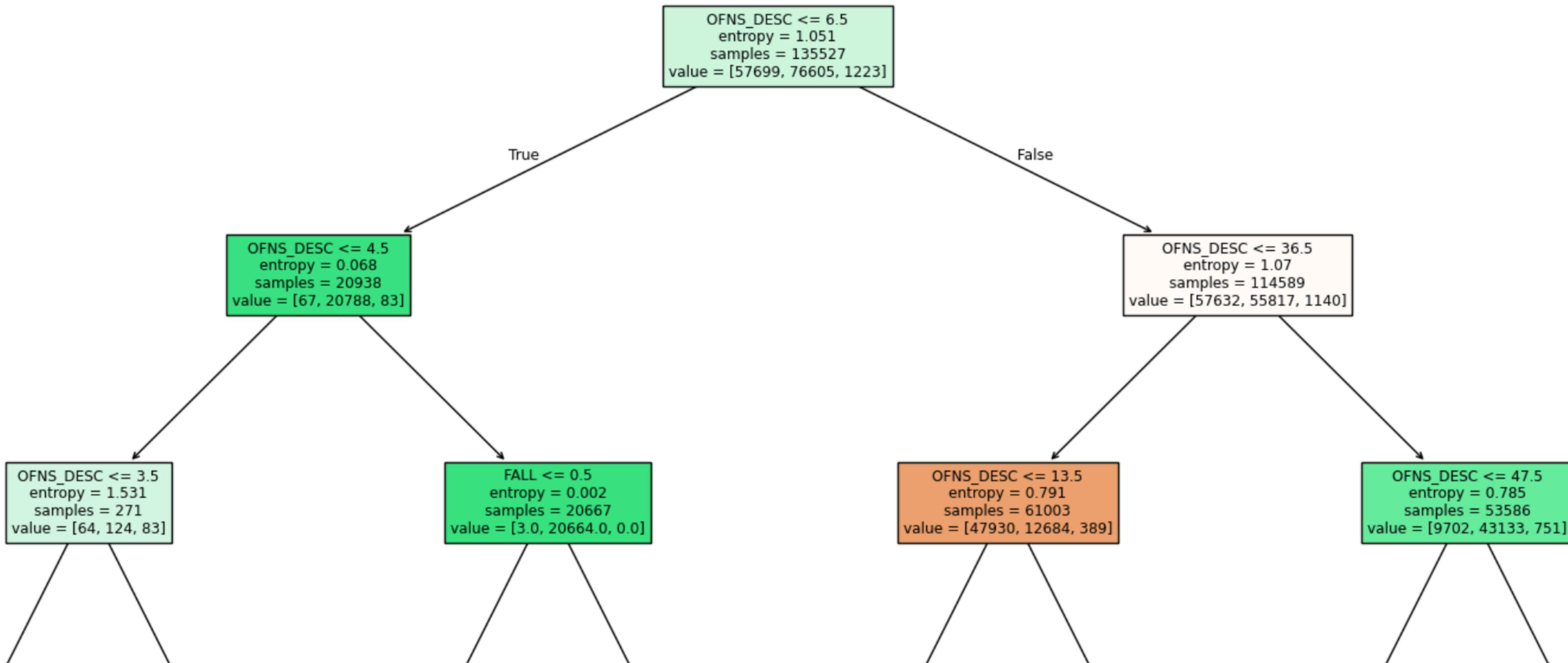
Best Params: {'criterion': 'entropy',
'max_depth': 15, 'min_samples_leaf': 4,
'min_samples_split': 10}

	precision	recall	f1-score	support
0	0.896273	0.938807	0.917047	24823
1	0.952556	0.919776	0.935879	32721
2	0.944559	0.853432	0.896686	539
accuracy	0.927294	0.927294	0.927294	0
macro avg	0.931129	0.904005	0.916537	58083
weighted avg	0.928428	0.927294	0.927467	58083

With Demographics

Best Params: {'criterion': 'entropy',
'max_depth': 15, 'min_samples_leaf': 4,
'min_samples_split': 10}

DECISION TREES (CART)



SVM

	precision	recall	f1-score	support
0	0.679851	0.762398	0.718762	24823
1	0.793659	0.733627	0.762463	32721
2	0.000000	0.000000	0.000000	539
accuracy	0.739115	0.739115	0.739115	0
macro avg	0.491170	0.498675	0.493742	58083
weighted avg	0.737655	0.739115	0.736711	58083

Without Demographics

Default Params: {'C': 1.0, 'kernel': 'rbf',
 'gamma': 'scale'}

	precision	recall	f1-score	support
0	0.679795	0.762801	0.718910	24823
1	0.793873	0.733413	0.762446	32721
2	0.000000	0.000000	0.000000	539
accuracy	0.739166	0.739166	0.739166	0
macro avg	0.491223	0.498738	0.493785	58083
weighted avg	0.737752	0.739166	0.736765	58083

With Demographics

Default Params: {'C': 1.0, 'kernel': 'rbf',
 'gamma': 'scale'}

LINEAR SVM

	precision	recall	f1-score	support
0	0.273829	0.144664	0.189314	24823
1	0.516623	0.710003	0.598069	32721
2	0.000000	0.000000	0.000000	539
accuracy	0.461805	0.461805	0.461805	0
macro avg	0.263484	0.284889	0.262461	58083
weighted avg	0.408066	0.461805	0.417829	58083

Without Demographics

Default Params: {'C': 1.0, 'kernel': 'linear',
 'gamma': 'scale'}

	precision	recall	f1-score	support
0	0.477977	0.265802	0.341626	24823
1	0.577023	0.780844	0.663636	32721
2	0.000000	0.000000	0.000000	539
accuracy	0.553484	0.553484	0.553484	0
macro avg	0.351667	0.348882	0.335088	58083
weighted avg	0.529339	0.553484	0.519860	58083

With Demographics

Default Params: {'C': 1.0, 'kernel': 'linear',
 'gamma': 'scale'}

LOGISTIC REGRESSION

	precision	recall	f1-score	support
0	0.282136	0.150667	0.196434	24823
1	0.518415	0.710217	0.599345	32721
2	0.000000	0.000000	0.000000	539
accuracy	0.464490	0.464490	0.464490	0
macro avg	0.266851	0.286961	0.265260	58083
weighted avg	0.412626	0.464490	0.421590	58083

Without Demographics

Best Params: {'max_iter': 1000, 'solver': 'liblinear'}

	precision	recall	f1-score	support
0	0.478397	0.271200	0.346163	24823
1	0.577515	0.776779	0.662488	32721
2	0.000000	0.000000	0.000000	539
accuracy	0.553501	0.553501	0.553501	0
macro avg	0.351971	0.349327	0.336217	58083
weighted avg	0.529795	0.553501	0.521152	58083

With Demographics

Best Params: {'max_iter': 1000, 'solver': 'liblinear'}

NAIVE BAYES

	precision	recall	f1-score	support
0	0.532478	0.466301	0.497197	24823
1	0.621984	0.690871	0.654620	32721
2	0.000000	0.000000	0.000000	539
accuracy	0.588485	0.588485	0.588485	0
macro avg	0.384820	0.385724	0.383939	58083
weighted avg	0.577960	0.588485	0.581267	58083

	precision	recall	f1-score	support
0	0.572105	0.428474	0.489980	24823
1	0.628862	0.758993	0.687826	32721
2	0.000000	0.000000	0.000000	539
accuracy	0.610695	0.610695	0.610695	0
macro avg	0.400322	0.395822	0.392602	58083
weighted avg	0.598770	0.610695	0.596890	58083

Without Demographics

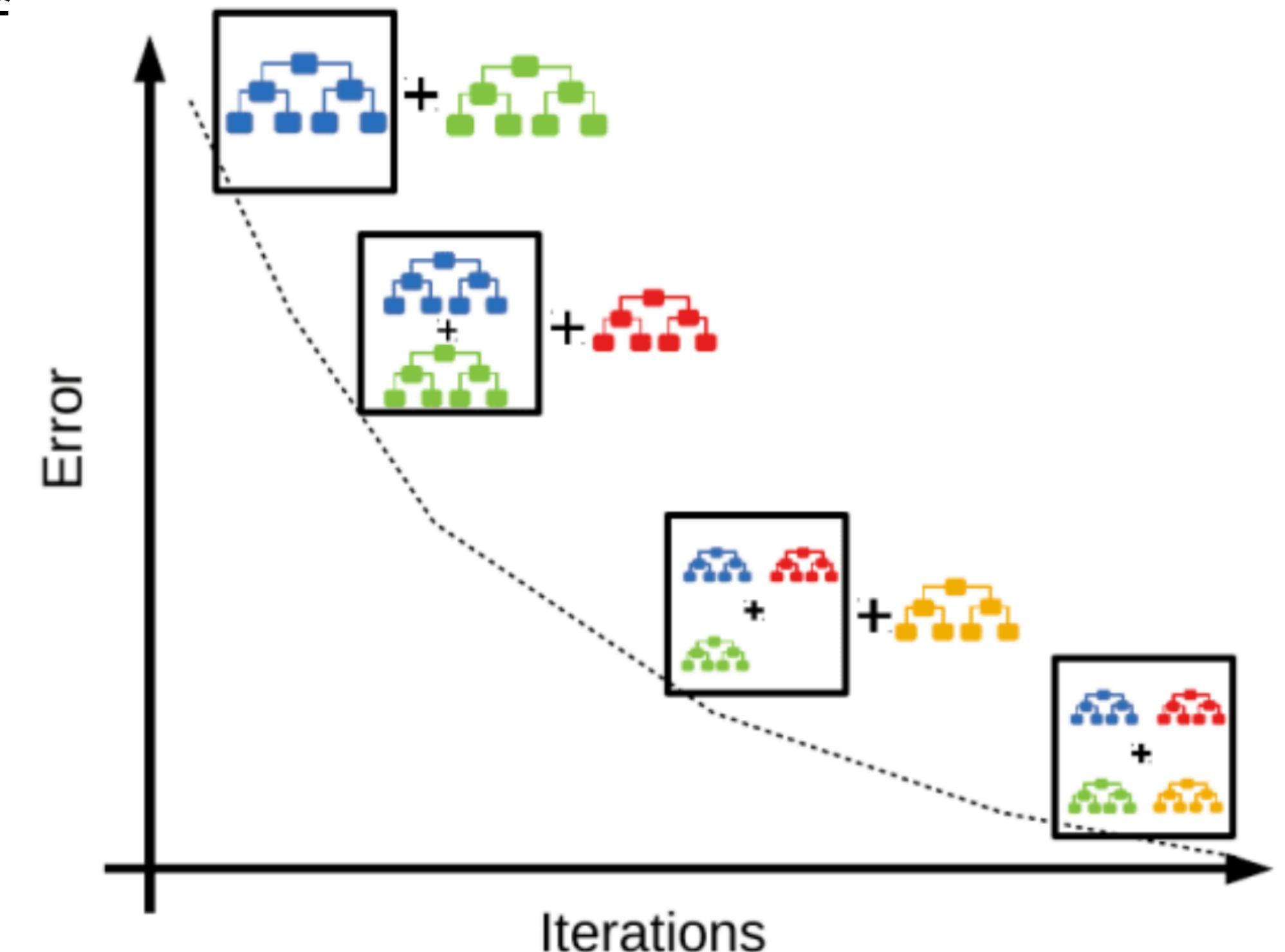
Mixed NB, For categorial/Numeric

With Demographics

Mixed NB, For categorial/Numeric

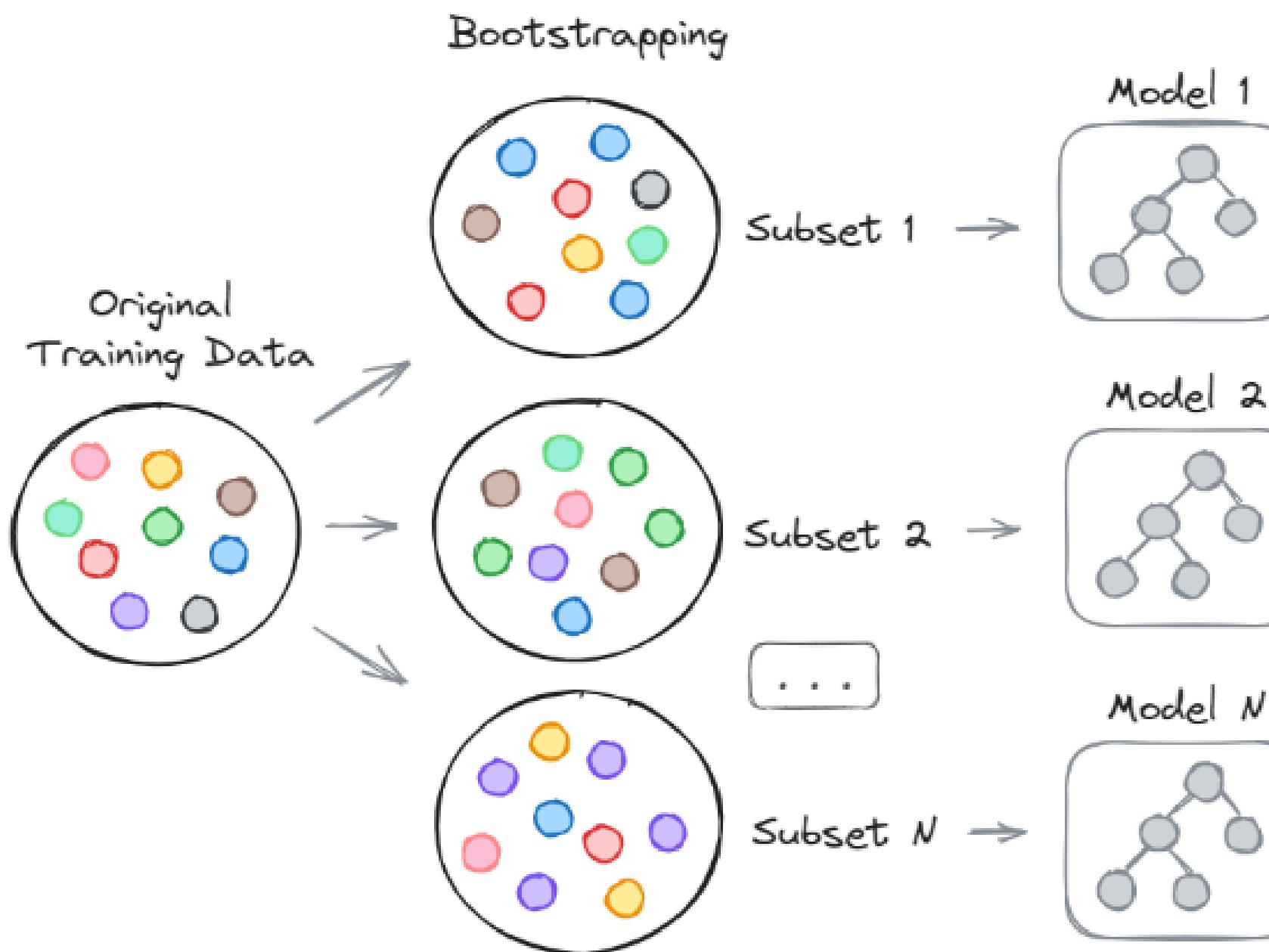
ENSEMBLE MODELS

- combine the predictions of multiple individual models (weak learners) to improve the overall performance
 - reduce overfitting
 - different errors, can cancel out
 - more robust
- Bagging
- Boosting

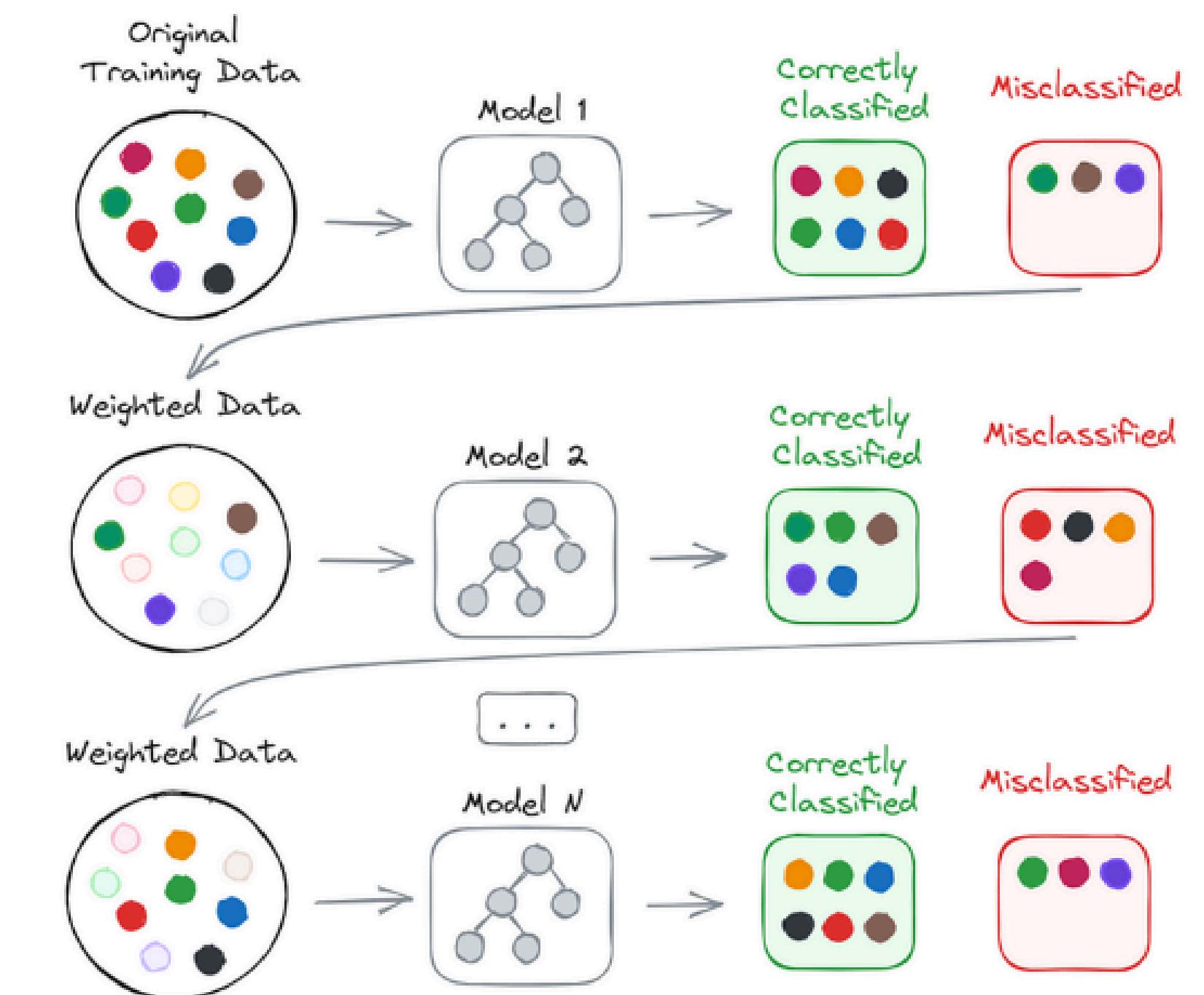


BAGGING (BOOTSTRAP) VS BOOSTING

Bagging



Boosting



RANDOM FOREST

	precision	recall	f1-score	support
0	0.889875	0.924828	0.907015	24823
1	0.940086	0.915895	0.927833	32721
2	0.970443	0.730983	0.833862	539
accuracy	0.917997	0.917997	0.917997	0
macro avg	0.933468	0.857235	0.889570	58083
weighted avg	0.918909	0.917997	0.918064	58083

	precision	recall	f1-score	support
0	0.874675	0.920799	0.897145	24823
1	0.935389	0.903029	0.918924	32721
2	0.988950	0.664193	0.794673	539
accuracy	0.908407	0.908407	0.908407	0
macro avg	0.933005	0.829340	0.870247	58083
weighted avg	0.909938	0.908407	0.908463	58083

Without Demographics

Best parameters for Random Forest:

```
{'max_depth': 30, 'min_samples_leaf': 1,
'min_samples_split': 5, 'n_estimators': 200}
```

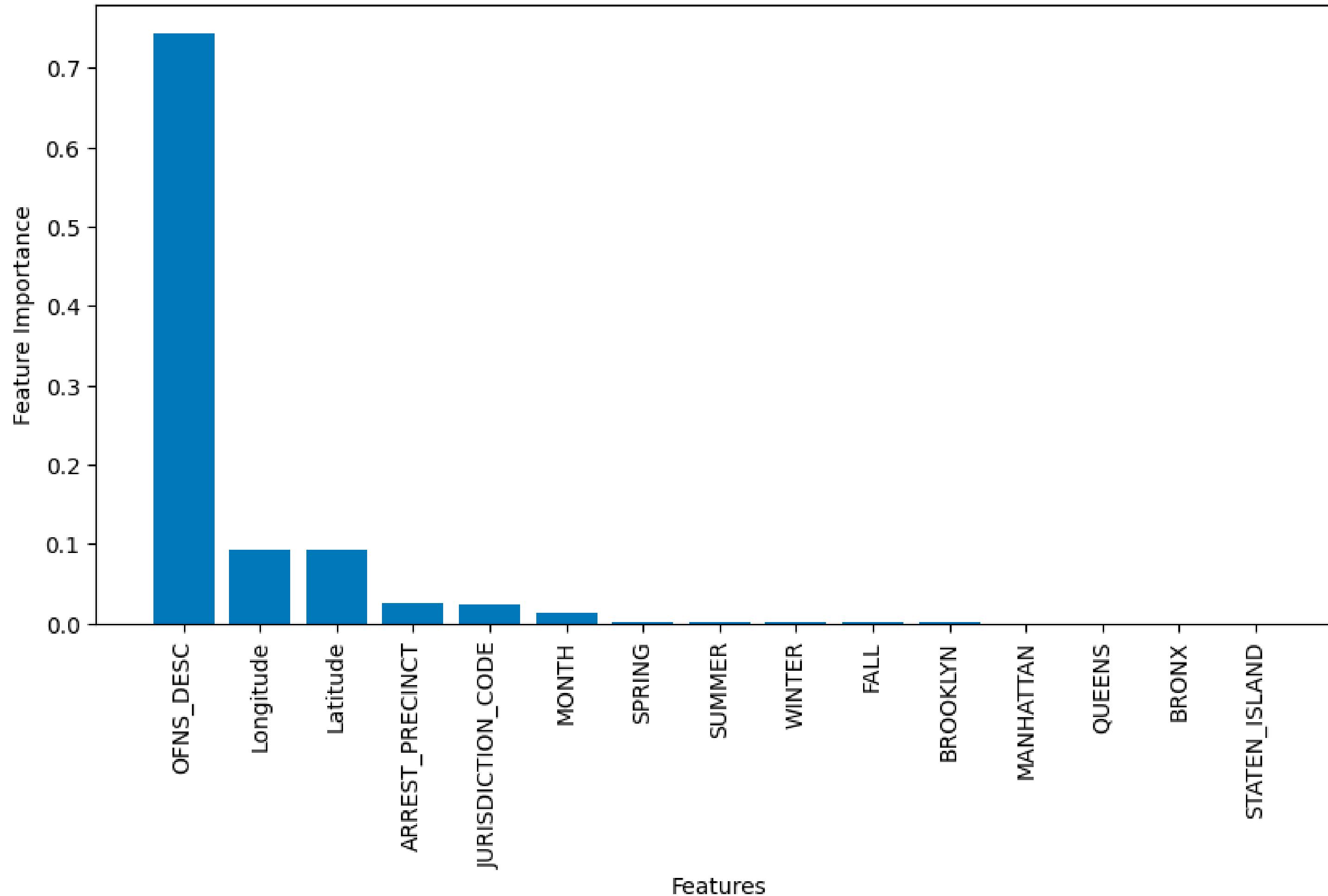
With Demographics

Best parameters for Random Forest:

```
{'max_depth': 40, 'min_samples_leaf': 1,
'min_samples_split': 5, 'n_estimators': 300}
```

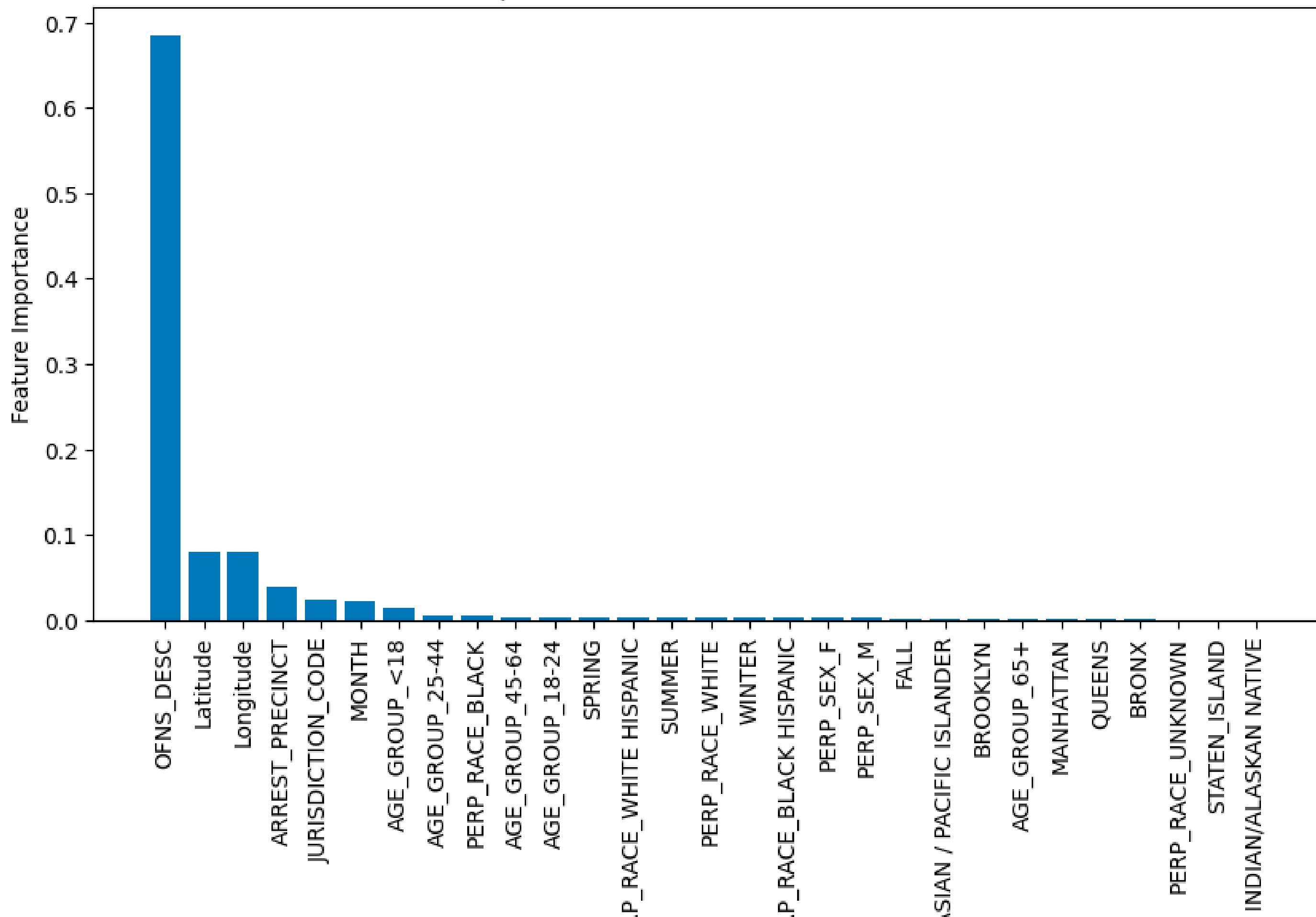
MODELS

Feature Importance of the Best Random Forest Model



MODELS

Feature Importance of the Best Random Forest Model



GRADIENT CLASSIFIER

	precision	recall	f1-score	support
0	0.894910	0.950610	0.921920	24823
1	0.961180	0.917117	0.938632	32721
2	0.943320	0.864564	0.902227	539
accuracy	0.930944	0.930944	0.930944	0
macro avg	0.933137	0.910764	0.920926	58083
weighted avg	0.932693	0.930944	0.931152	58083

Without Demographics

Best parameters for Gradient Clasifier :

```
{'learning_rate': 0.1, 'max_depth': 5,
'min_samples_leaf': 1, 'n_estimators': 300}
```

	precision	recall	f1-score	support
0	0.904030	0.947951	0.925470	24823
1	0.959361	0.925644	0.942201	32721
2	0.962733	0.862709	0.909980	539
accuracy	0.934594	0.934594	0.934594	0
macro avg	0.942041	0.912101	0.925884	58083
weighted avg	0.935746	0.934594	0.934752	58083

With Demographics

Best parameters for Gradient Classifier:

```
{'learning_rate': 0.1, 'max_depth': 5,
'min_samples_leaf': 4, 'n_estimators': 300}
```

HISTOGRAM GRADIENT

	precision	recall	f1-score	support
0	0.887117	0.958949	0.921635	24823
1	0.967552	0.909477	0.937616	32721
2	0.935091	0.855288	0.893411	539
accuracy	0.930117	0.930117	0.930117	0
macro avg	0.929920	0.907905	0.917554	58083
weighted avg	0.932875	0.930117	0.930376	58083

Without Demographics

Best parameters: {'l2_regularization': 0.1,
 'learning_rate': 0.1, 'max_depth': 7}

	precision	recall	f1-score	support
0	0.903500	0.946340	0.924424	24823
1	0.958469	0.925369	0.941628	32721
2	0.941057	0.858998	0.898157	539
accuracy	0.933716	0.933716	0.933716	0
macro avg	0.934342	0.910236	0.921403	58083
weighted avg	0.934815	0.933716	0.933872	58083

With Demographics

Best parameters: {'l2_regularization': 0.1,
 'learning_rate': 0.1, 'max_depth': 7}

BAGGING CLASSIFIER

	precision	recall	f1-score	support
0	0.913929	0.934657	0.924177	24823
1	0.950166	0.935240	0.942644	32721
2	0.951020	0.864564	0.905734	539
accuracy	0.934335	0.934335	0.934335	0
macro avg	0.938372	0.911487	0.924185	58083
weighted avg	0.934687	0.934335	0.934409	58083

Without Demographics

Best parameters: {'max_samples': 0.5,
'n_estimators': 400}

	precision	recall	f1-score	support
0	0.912496	0.936390	0.924288	24823
1	0.951232	0.934110	0.942593	32721
2	0.964435	0.855288	0.906588	539
accuracy	0.934353	0.934353	0.934353	0
macro avg	0.942721	0.908596	0.924490	58083
weighted avg	0.934800	0.934353	0.934436	58083

With Demographics

Best parameters: {'max_samples': 0.5,
'n_estimators': 400}

XGBOOST

	precision	recall	f1-score	support
0	0.899332	0.944366	0.921299	24823
1	0.956467	0.921916	0.938873	32721
2	0.964435	0.855288	0.906588	539
accuracy	0.930892	0.930892	0.930892	0
macro avg	0.940078	0.907190	0.922254	58083
weighted avg	0.932123	0.930892	0.931063	58083

Without Demographics

Best parameters: {'max_depth': 4,
'n_estimators': 300}

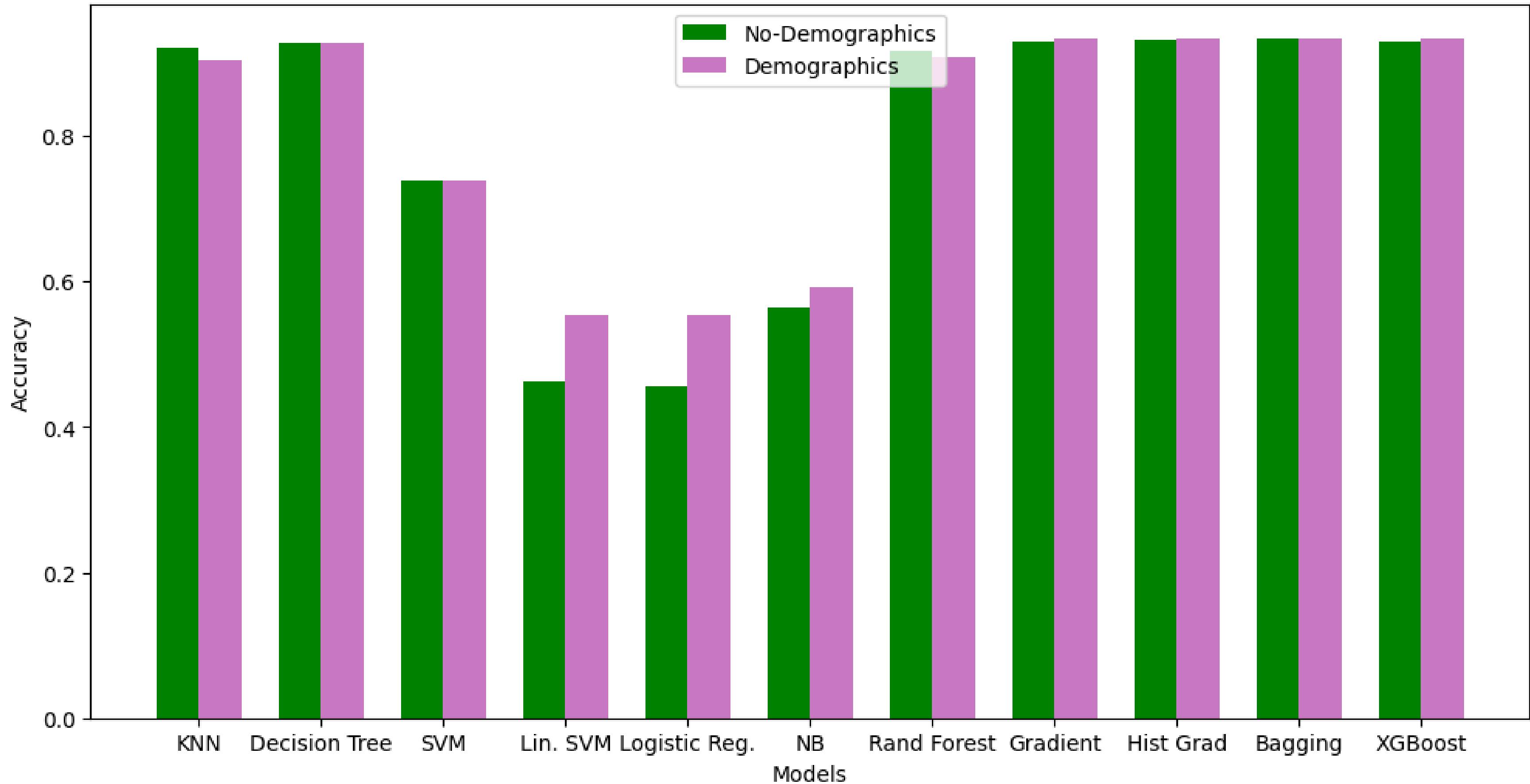
	precision	recall	f1-score	support
0	0.905841	0.944084	0.924567	24823
1	0.956514	0.927661	0.941866	32721
2	0.966527	0.857143	0.908555	539
accuracy	0.934025	0.934025	0.934025	0
macro avg	0.942960	0.909629	0.924996	58083
weighted avg	0.934950	0.934025	0.934164	58083

With Demographics

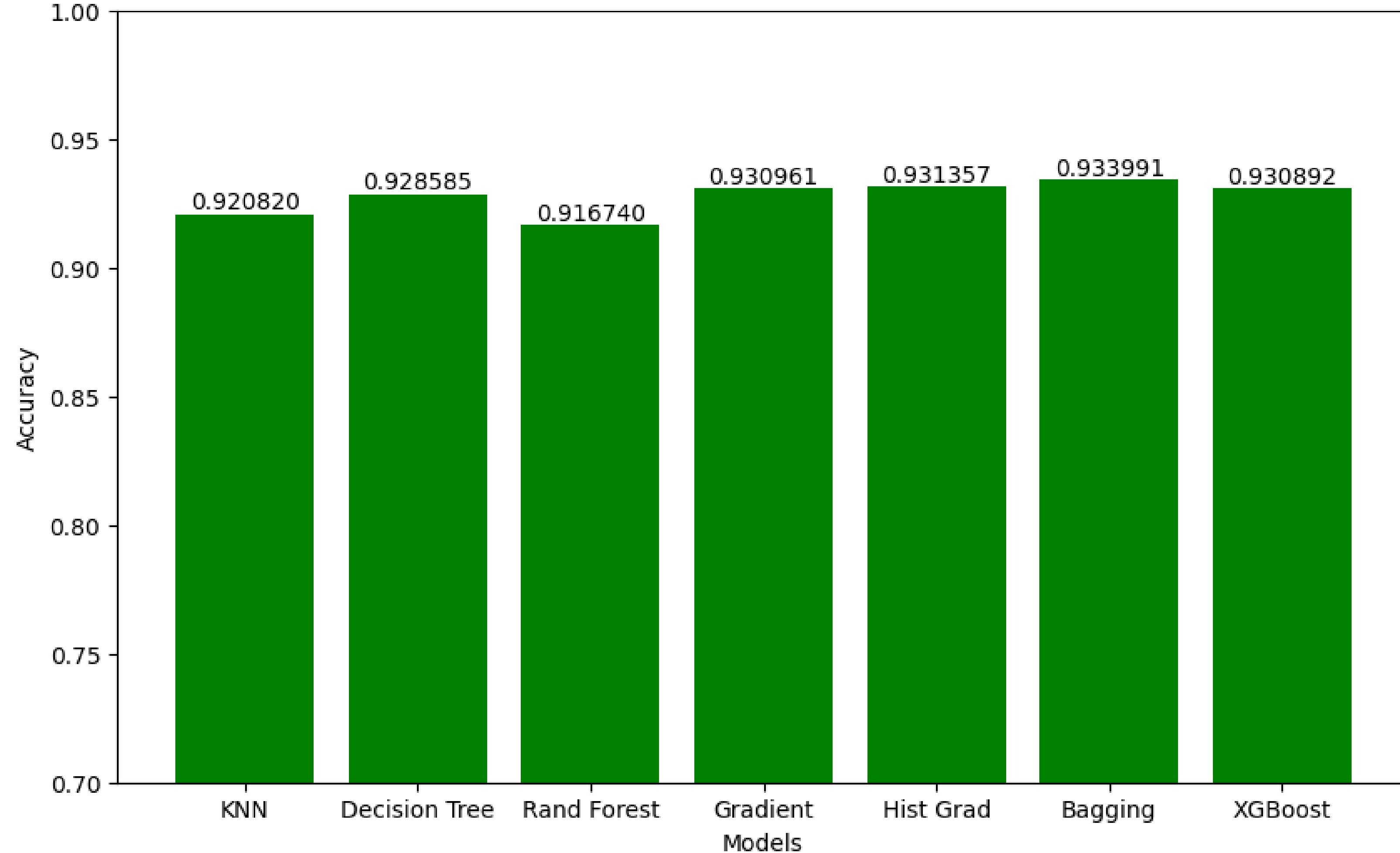
Best parameters: {'max_depth': 5,
'n_estimators': 200}

CONCLUSIONS + TAKEAWAYS

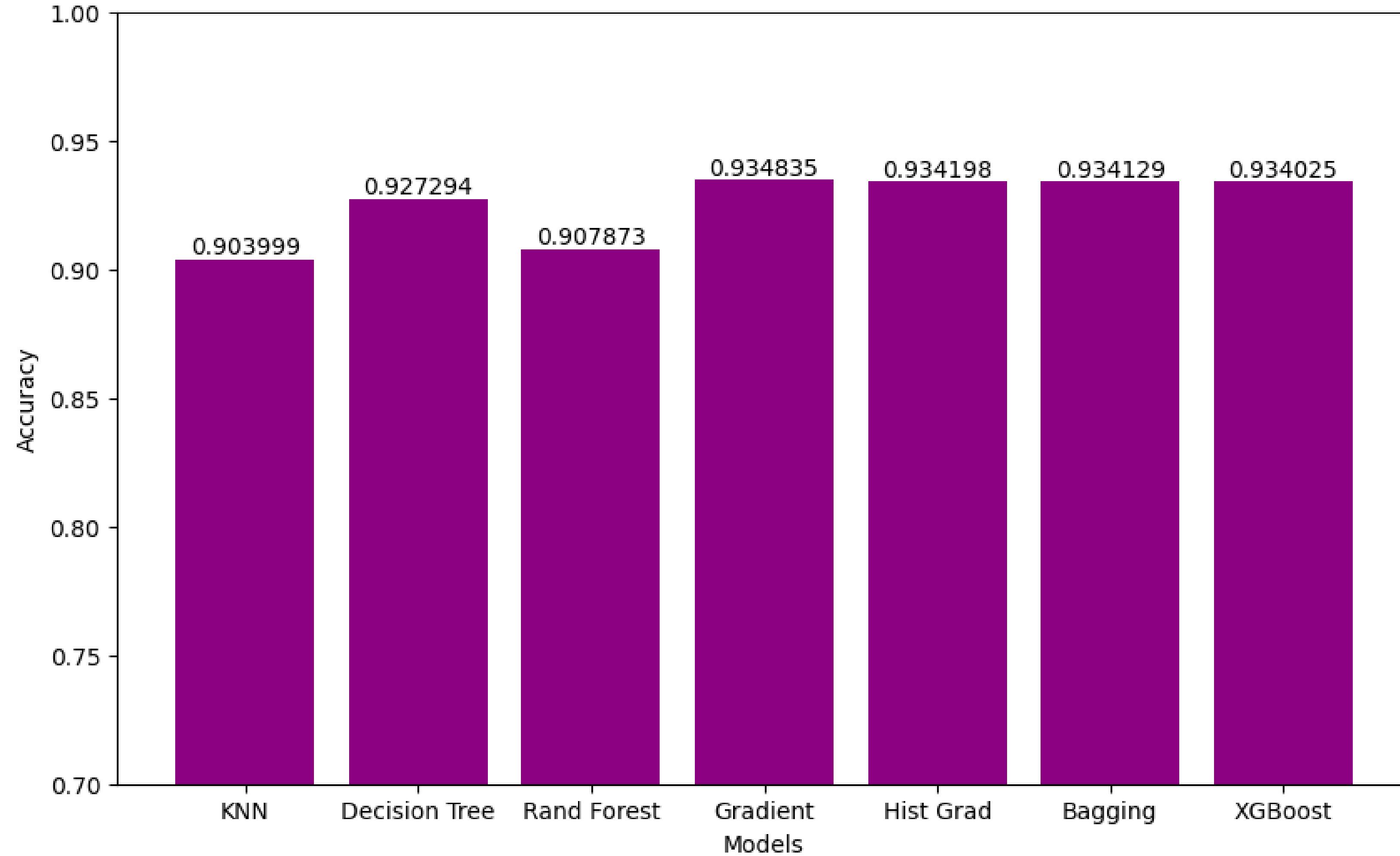
Accuracy of models for the two datasets



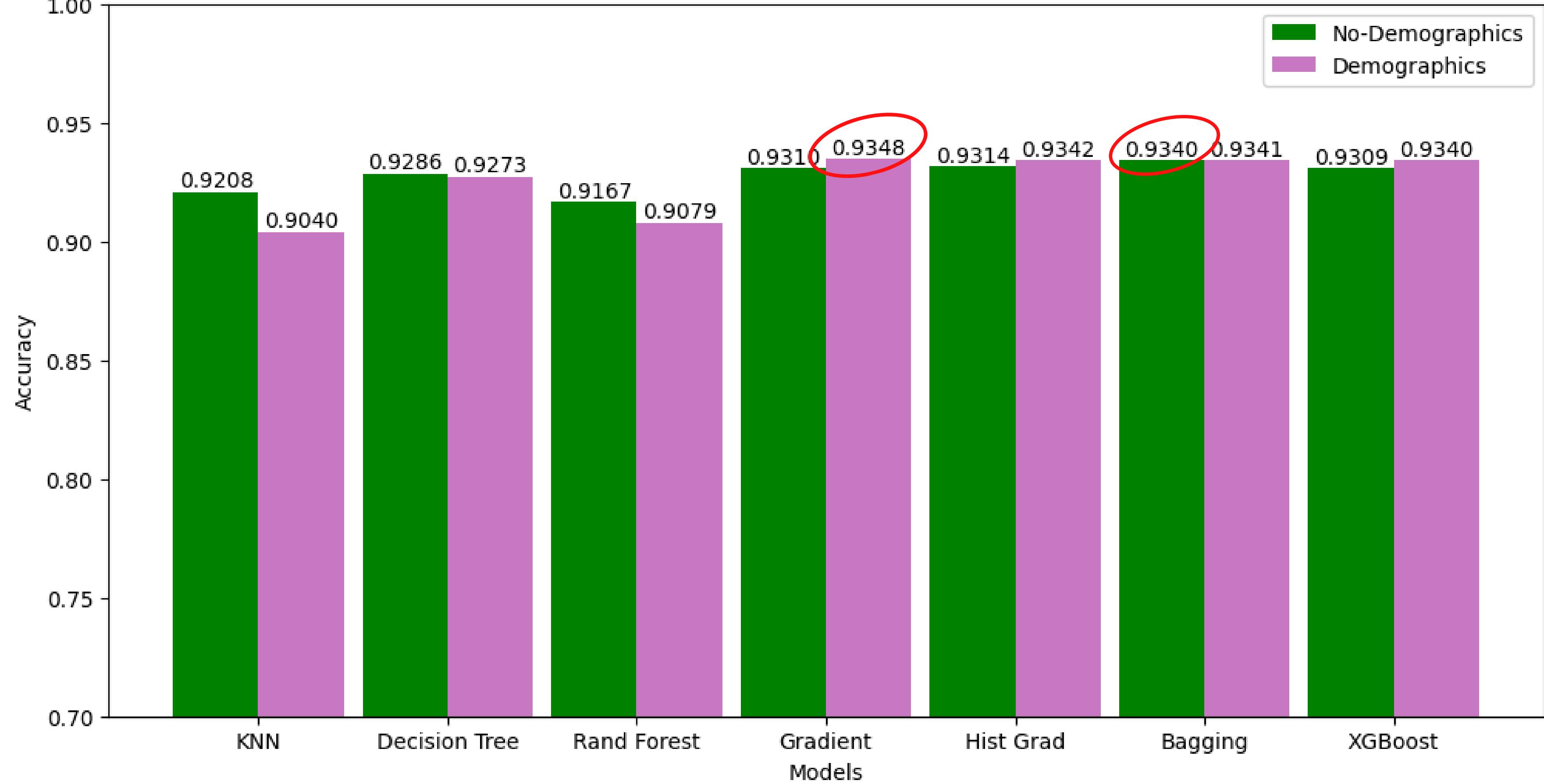
Best Accuracies of Non-Demo models



Best Accuracies of Demo models



Accuracy of models for the two datasets



CONCLUSION

- Geography matters more than time of year, although there seems to be a spike in all crimes in Spring/Summer
- Models with demographic data are generally less accurate than models without demographic data
- Higher risk areas are identified allowing for better resource allocation to combat crime
- Offenses grouped together as “miscellaneous” warrant further attention from authorities

× × × ×

THANK YOU