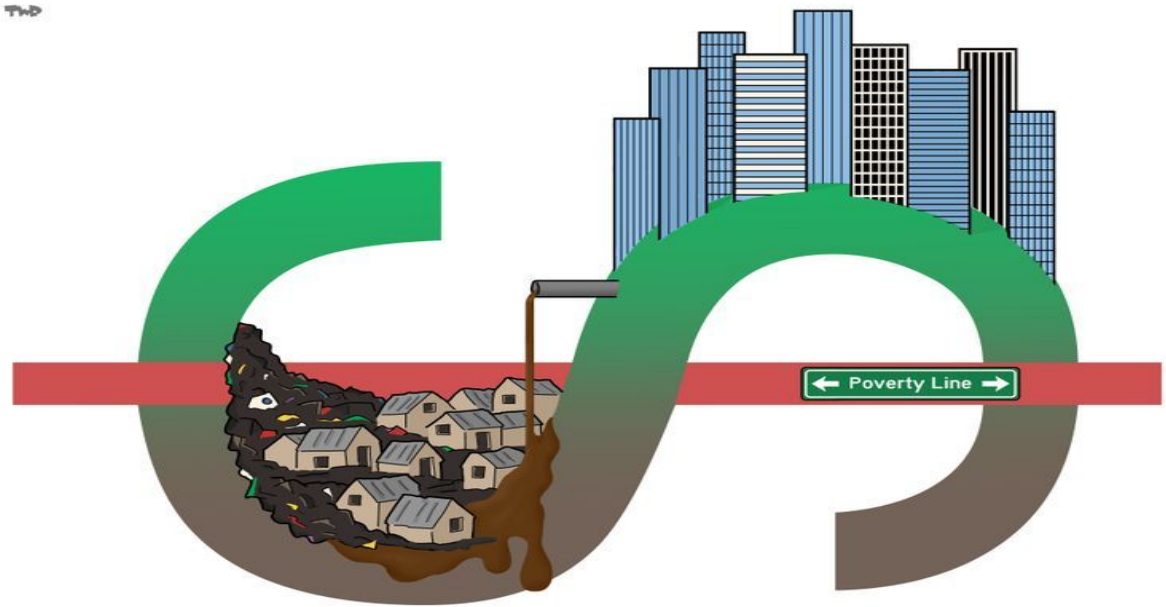
Understanding Poverty

Building a Supervised Learning classifier for Poverty Thresholds



Problem: Poverty

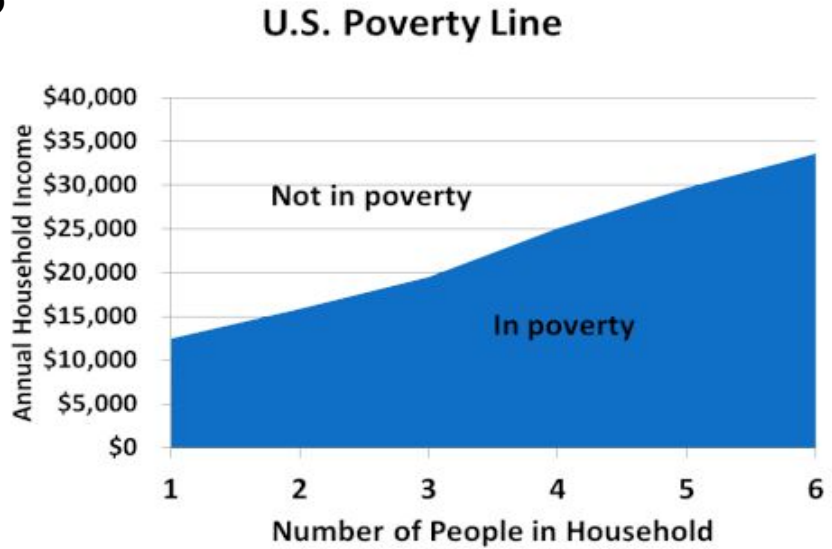
- Affects 40.6 million americans
- The U.S. spends only 16.2 percent of its GDP on social programs, compared to 21.3 percent that similarly developed countries do
- The U.S. is 36th out of 175 developed countries in rates of childhood poverty

What is the Poverty Threshold?

The Census Bureau assigns each person or family one out of 48 possible poverty thresholds.

- Threshold of 100: $12,752 for one person under 65
- $11,756 for one person in the household over 65

A single person with a threshold of 501,
the highest threshold, makes at least $63,887



Goal of project

What traits are most indicative of whether or not an individual will be have a poverty threshold equal to or below 200?

testing several different supervised learning models for the most accurate classification

I predict most impactful variables will include rent to income ratio and education

Why is this important?

- **Understanding who is affected by changes to poverty threshold calculations**
- **Help focus targeting efforts of programs**
- **Help focus how and where to focus political messaging**
- **Identifying if someone qualifies for federal aid programs**

Image Courtesy: The Times of India

The Dataset

The Dataset

- 148,594 rows and 31 variables

- The variables surround educational attainment, race, health care coverage type, food stamp recipients, metropolitan type, income, rent paid, and ownership type.

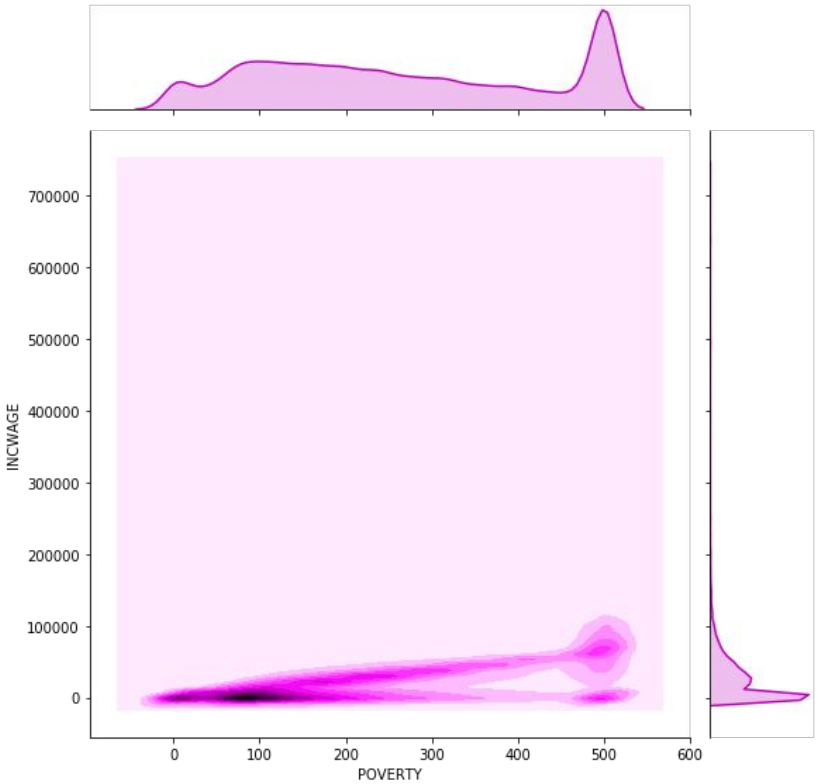
- target variable is ‘lowerpov’:
  - value 1 in case of a poverty threshold less than or equal to 200
  - value of 0 otherwise

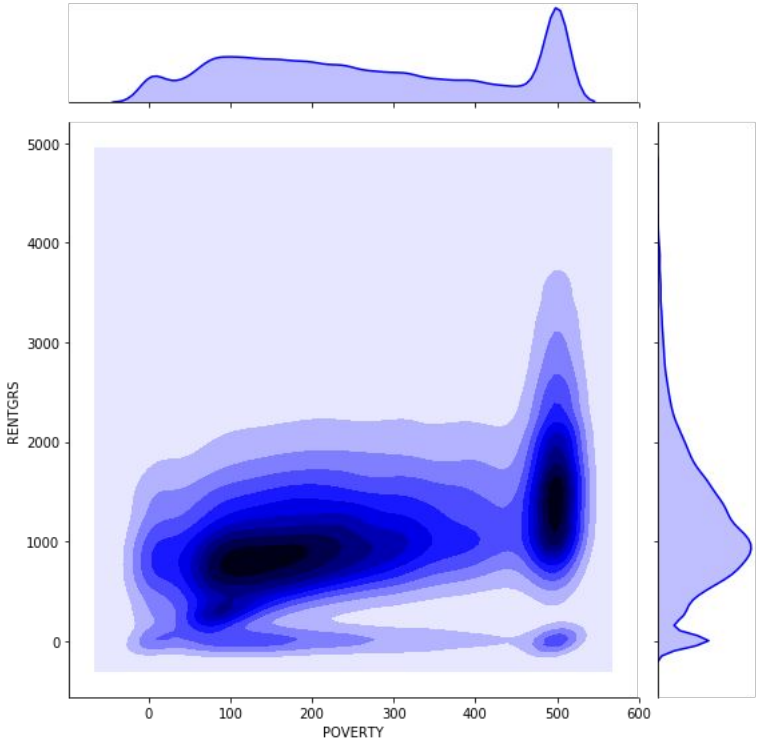
Limits of the Data

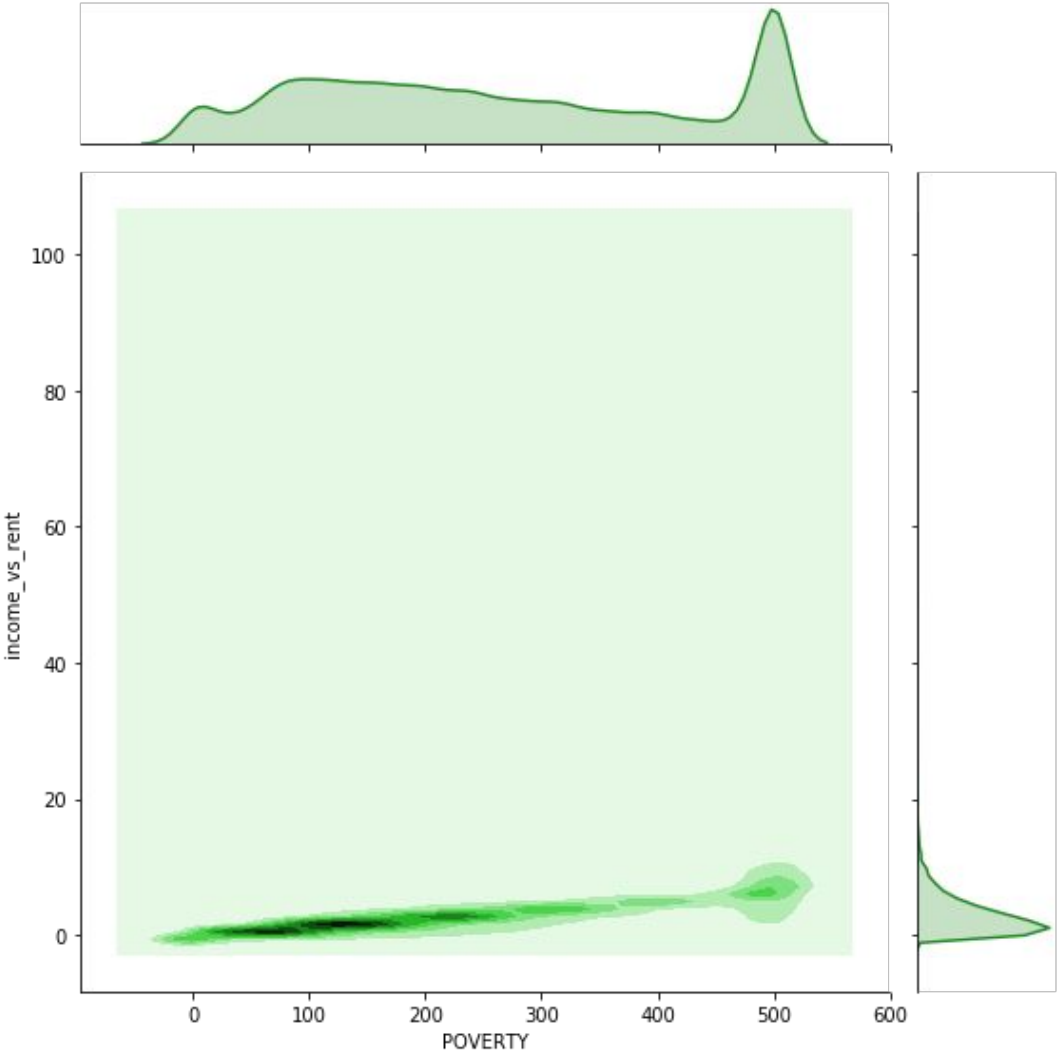
- Reduced to single renters only

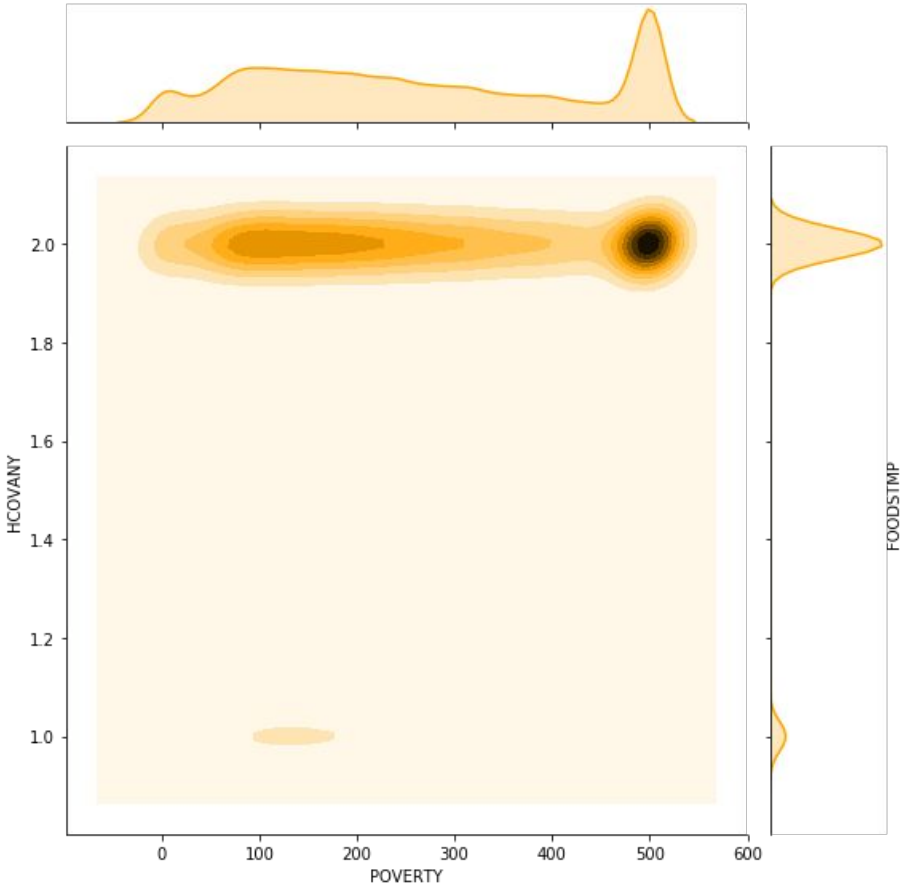
- Excludes people below 26 years old

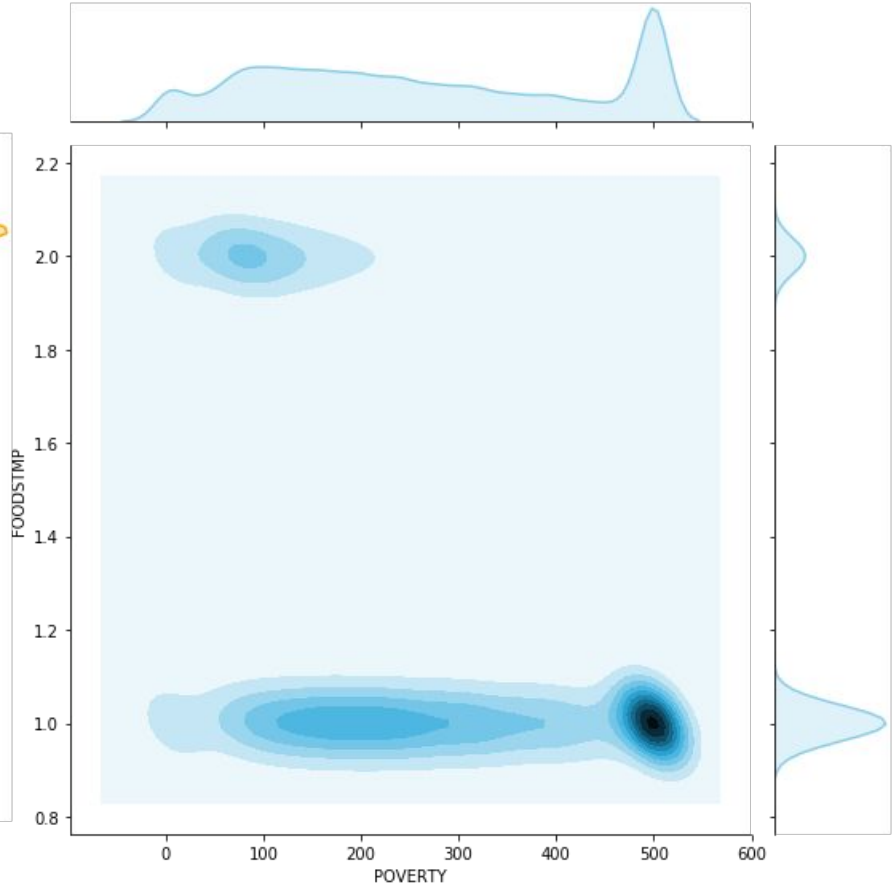
- Poverty status cannot be determined for people in:
  - Institutional group quarters (such as prisons or nursing homes)
  - College dormitories
  - Military barracks
  - Living situations without conventional housing (and who are not in shelters)

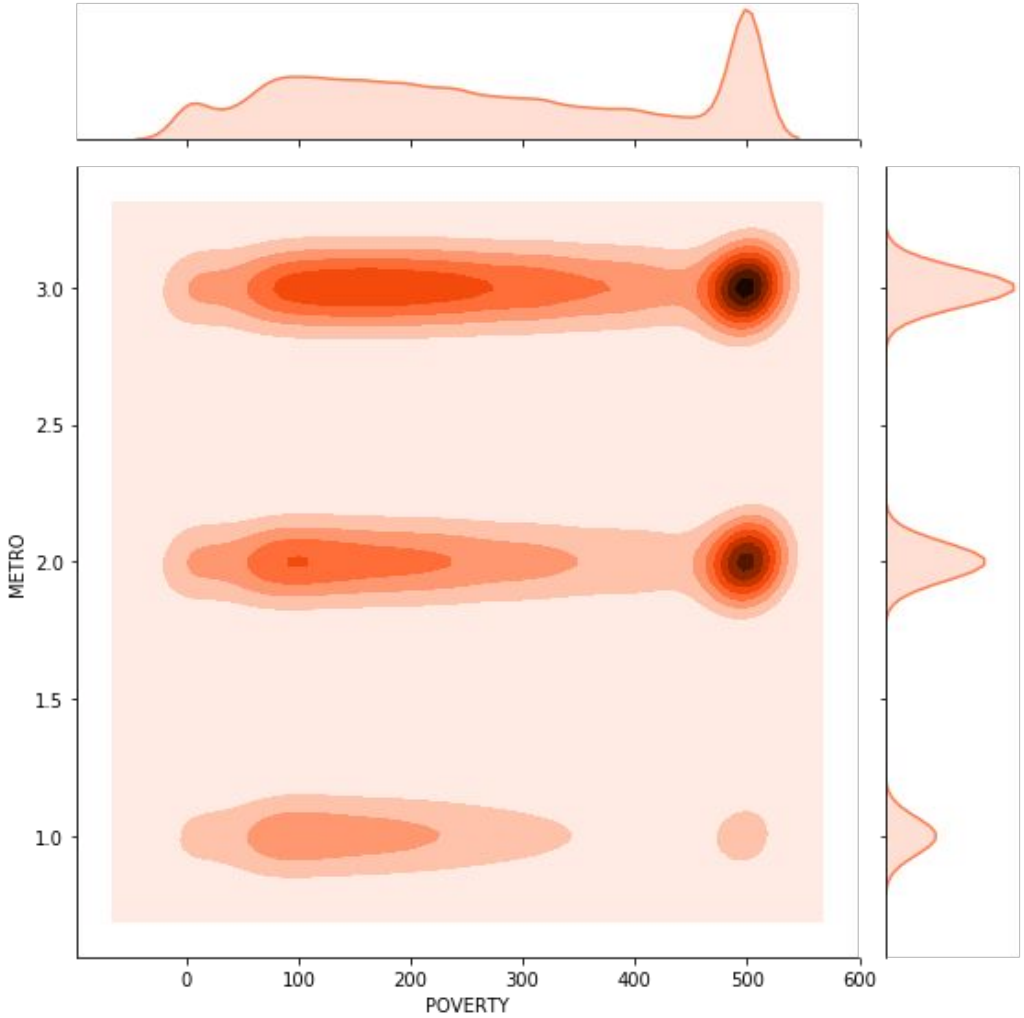

INCWAGE
POVERTY
0
100000
200000
300000
400000
500000
600000
700000
0
100
200
300
400
500
600

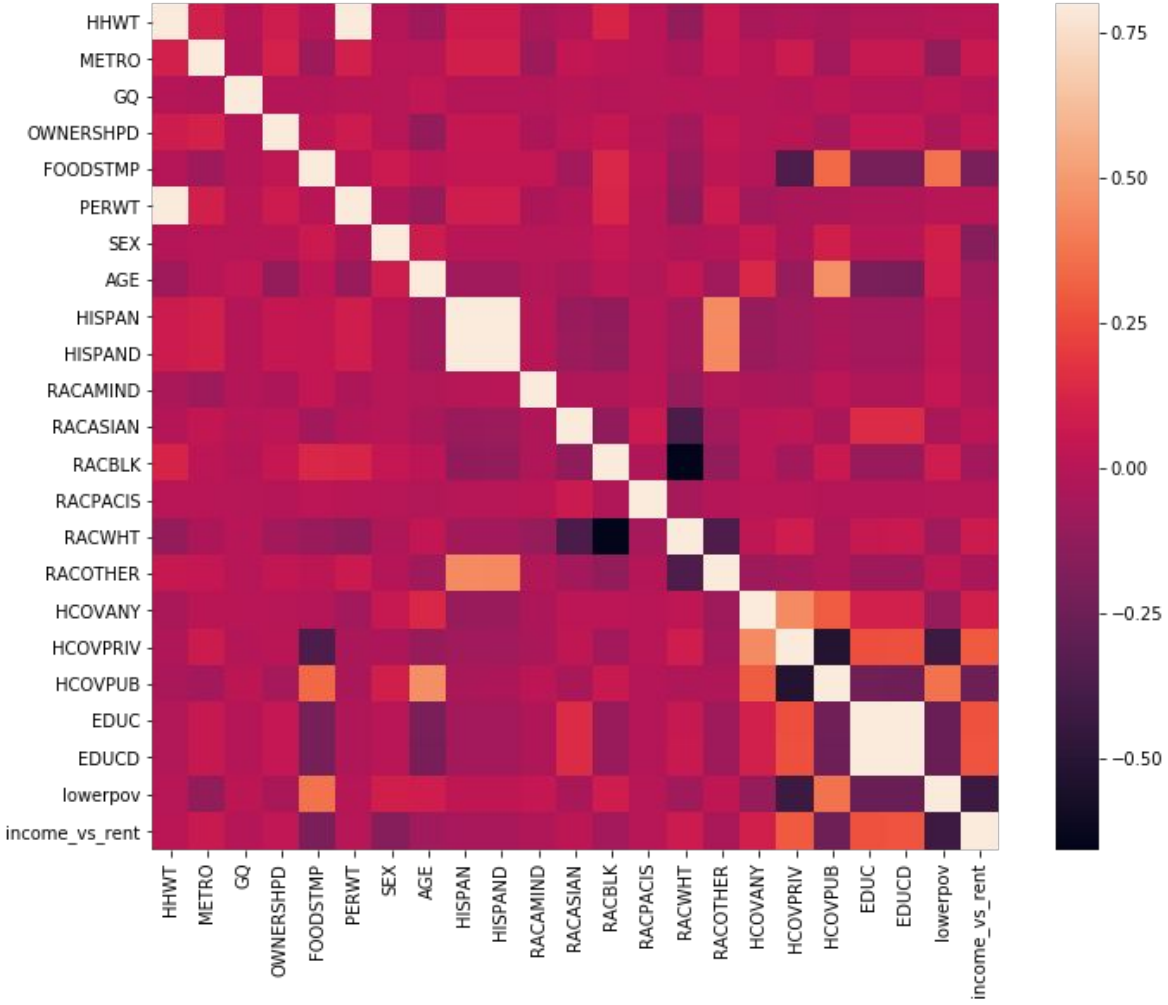

RENTGRS
POVERTY
0
1000
2000
3000
4000
5000
0
100
200
300
400
500
600


income_vs_rent
POVERTY
0
20
40
60
80
100
0
100
200
300
400
500
600


HCOVANY
2.0
1.8
1.6
1.4
1.2
1.0
0
100
200
300
400
500
600
POVERTY


FOODSTMP
2.2
2.0
1.8
1.6
1.4
1.2
1.0
0.8
0
100
200
300
400
500
600
POVERTY


3.0
2.5
2.0
1.5
1.0
METRO
0
100
200
300
400
500
600
POVERTY


HHWT
METRO
GQ
OWNERSHPD
FOODSTMP
PERWT
SEX
AGE
HISPAN
HISPAND
RACAMIND
RACASIAN
RACBLK
RACPACIS
RACWHT
RACOTHER
HCOVANY
HCOVPRIV
HCOVPUB
EDUC
EDUCD
lowerpov
income_vs_rent
0.75
0.50
0.25
0.00
−0.25
−0.50

Variable Correlations

	lowerpov
HISPAN	0.030791
RACASIAN	-0.044368
RACBLK	0.086742
RACPACIS	-0.001068
RACWHT	-0.076643

	lowerpov
HCOVANY	-0.106239
HCOVPRIV	-0.429340
HCOVPUB	0.370129

	lowerpov
FOODSTMP	0.371749
SEX	0.094411
AGE	0.088128
EDUCD	- 0.263232
income_vs_rent	-0.427177
METRO	-0.118219

Approach

1) Test Using 10 components PCA 10 Select KBest

2) gridsearch cv to find ideal parameters for each classifier

3)

- Native Bayes Classifier
- Knn classifier
- Random Forest
- Decision tree
- Logistic regression
- Svm classifier
- Gradient boosted classifier

4) Use AUC and classification report to evaluate best model

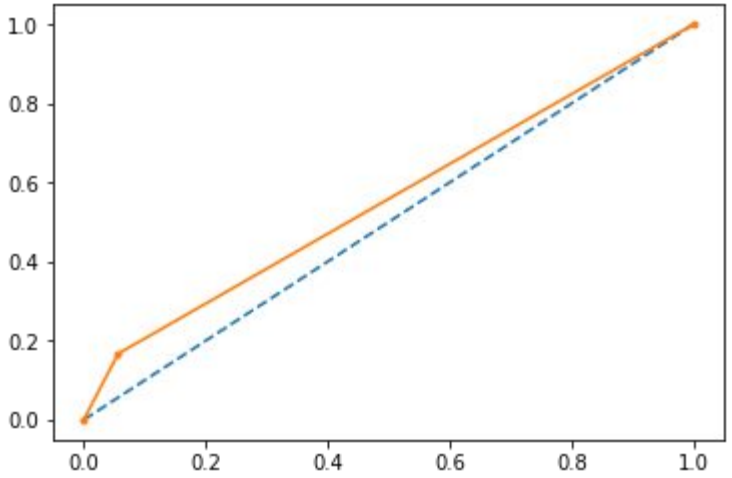
Worst Select K Best Classifier: Naive Bayes

With 20% Holdout: 0.6164743093643796

Testing on Sample: 0.6193924384564653

Naive Bayes Classification report :

	precision	recall	f1-score	support
0	0.61	0.94	0.74	17201
1	0.68	0.17	0.27	12518



AUC: 0.555

Select K Best Classifier: KNN

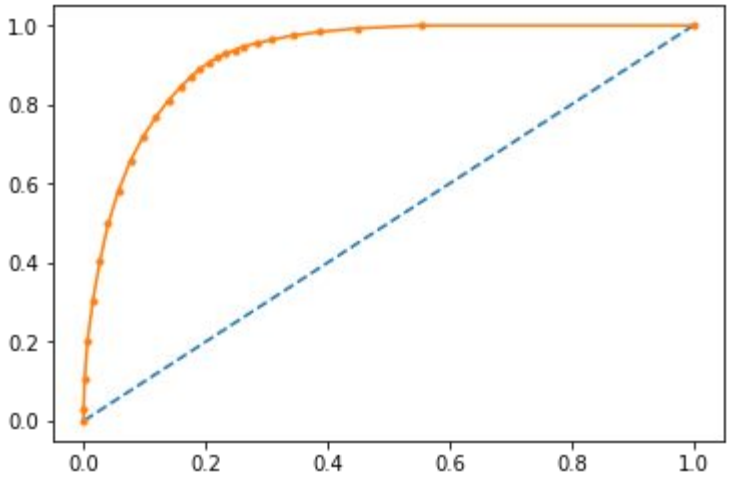
{'n_neighbors': 25}
0.8339973350202565

With 20% Holdout: 0.8341465056024765

Testing on Sample: 0.8441727122225662

KNN report :

	precision	recall	f1-score	support
0	0.90	0.82	0.86	17201
1	0.78	0.87	0.82	12518



AUC: 0.923

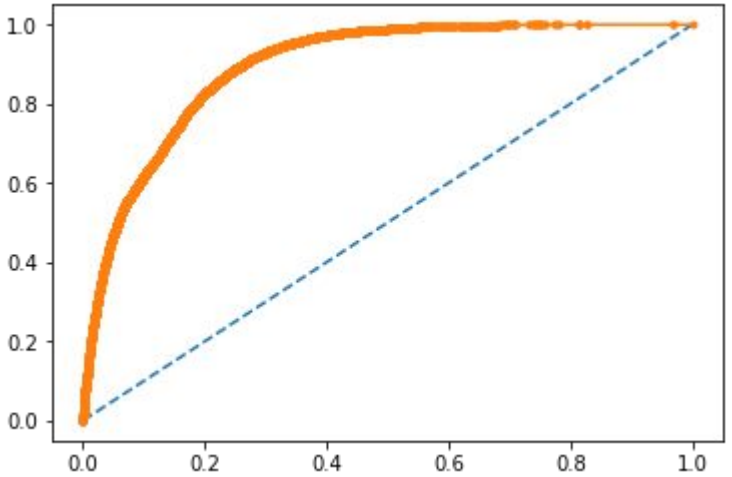
Select K Best Classifier: Logistic Regression

{'C': 0.1, 'penalty': 'l1'}
0.8095481647980403

With 20% Holdout: 0.8078333725899256
Testing on Sample: 0.8097971654306365

Logistic regression report :

	precision	recall	f1-score	support
0	0.84	0.82	0.83	17201
1	0.76	0.79	0.77	12518



AUC: 0.896

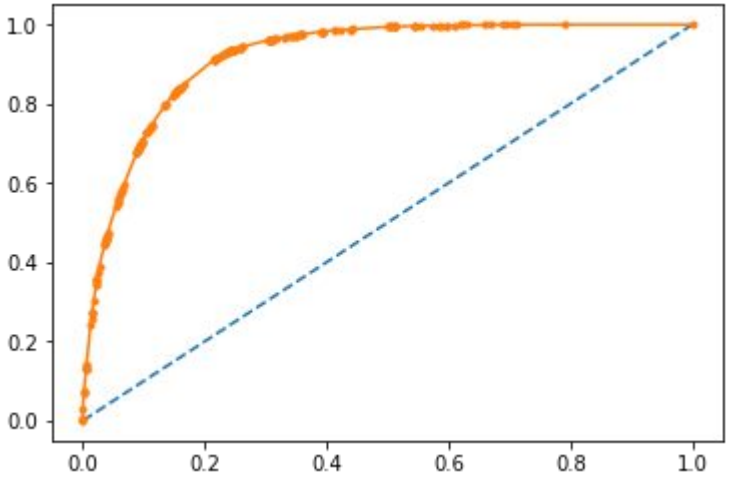
Select K Best Classifier: Decision Tree

{'max_depth': 8,
'max_features': 8}
0.8398791337469884

With 20% Holdout: 0.8408762071402134
Testing on Sample: 0.8404578919741039

Decision Tree report :

	precision	recall	f1-score	support
0	0.88	0.84	0.86	17201
1	0.79	0.84	0.82	12518



AUC: 0.919

Best Select K Best Classifier: Random Forest

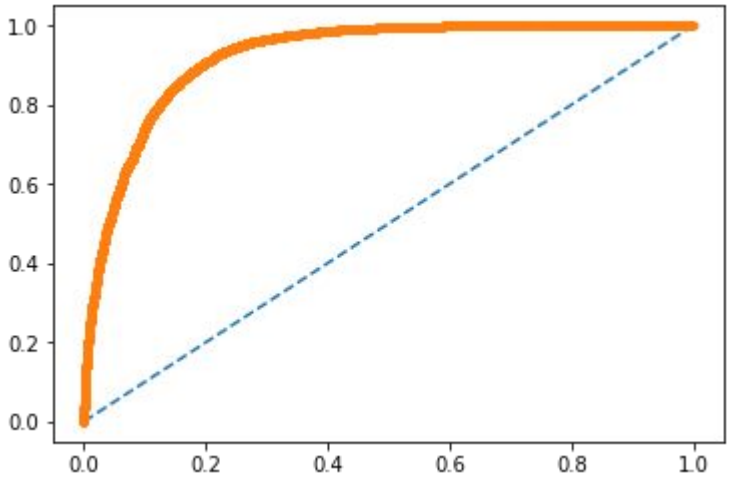
{'max_depth': 8,
'max_features': 7,
'n_estimators': 200}
0.8403300267843924

With 20% Holdout: 0.8462599683704027

Testing on Sample: 0.8449331736140087

Random Forest report :

	precision	recall	f1-score	support
0	0.90	0.83	0.86	17154
1	0.79	0.87	0.83	12565



AUC: 0.925

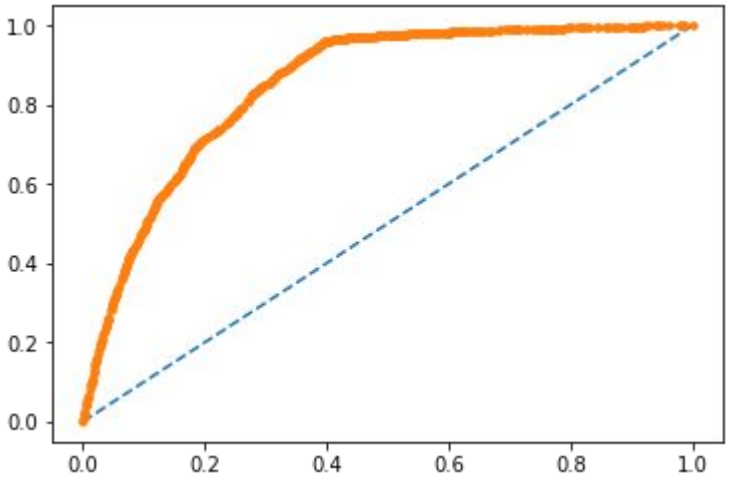
Worst PCA Classifier: Naive Bayes

With 20% Holdout: 0.7659073320098254

Testing on Sample: 0.7677564370028399

Naive Bayes Classification report :

	precision	recall	f1-score	support
0	0.85	0.72	0.78	17201
1	0.68	0.83	0.75	12518



AUC: 0.850

PCA Classifier: Random Forest

{'max_depth': 8, 'max_features': 6,
'n_estimators':750}

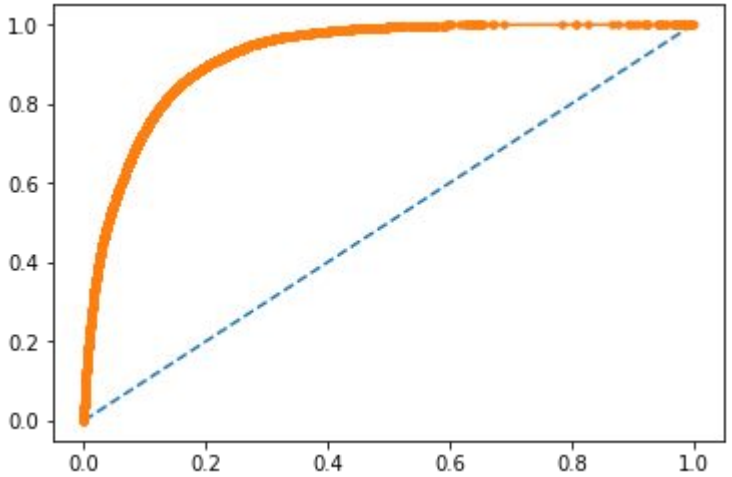
0.8403300267843924

With 20% Holdout: 0.8372758168175242

Testing on Sample: 0.8405722976701616

Random Forest report :

	precision	recall	f1-score	support
0	0.89	0.83	0.86	17154
1	0.79	0.87	0.82	12565



AUC: 0.923

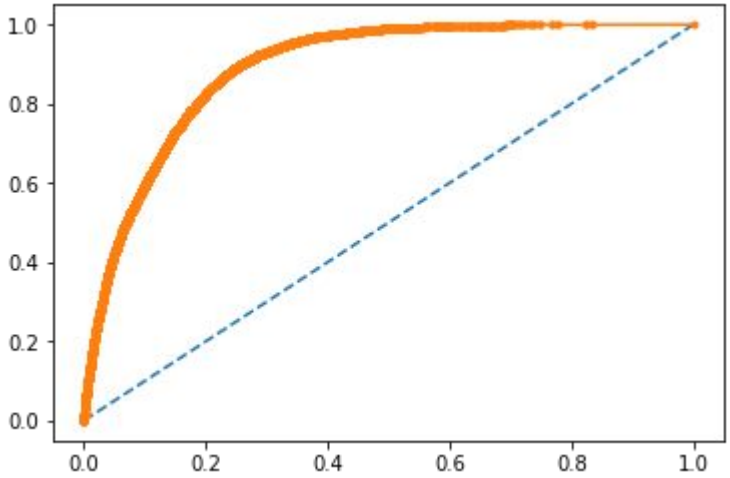
Select K Best Classifier: Logistic Regression

{'C': 0.001, 'penalty': 'l1'}
0.8111969527706367

With 20% Holdout: 0.8089774218513409
Testing on Sample: 0.810739329986406

Logistic regression report :

	precision	recall	f1-score	support
0	0.85	0.82	0.83	17201
1	0.76	0.80	0.78	12518



AUC: 0.892

PCA Classifier: KNN

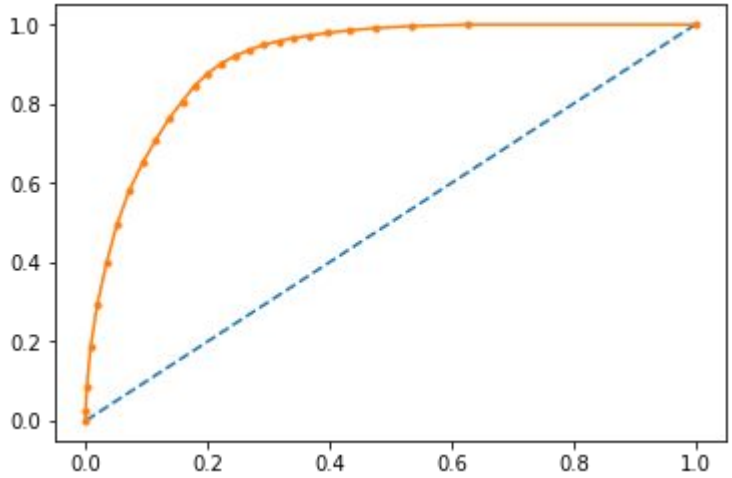
{'n_neighbors': 25}
0.8158673970685223

With 20% Holdout: 0.816312796527474

Testing on Sample: 0.8315948154030446

KNN report :

	precision	recall	f1-score	support
0	0.90	0.80	0.85	17201
1	0.76	0.88	0.81	12518



AUC: 0.911

Best PCA classifier: Gradient Boosted using PCA

{'max_depth': 7,
'n_estimators': 50}

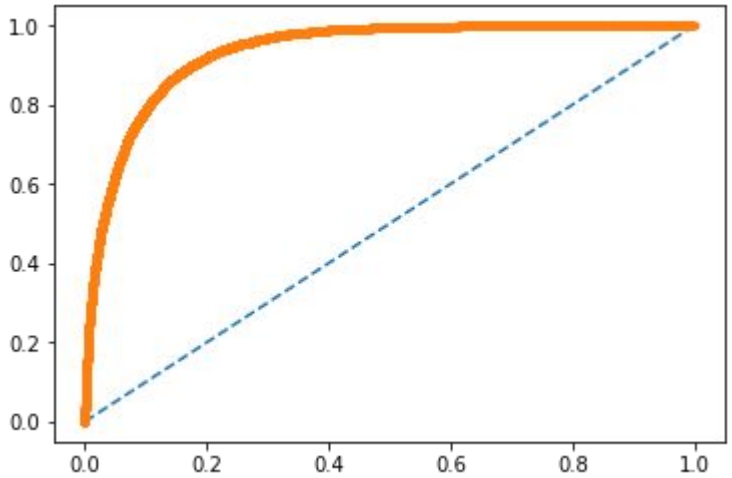
0.8448389571584317

With 20% Holdout: 0.8448130825397894

Testing on Sample: 0.8595636432157422

Gradient Boosting report :

	precision	recall	f1-score	support
0	0.90	0.85	0.88	17201
1	0.81	0.87	0.84	12518



AUC: 0.936

Thank You!