A Decision Tree Analysis on NYC poverty Status

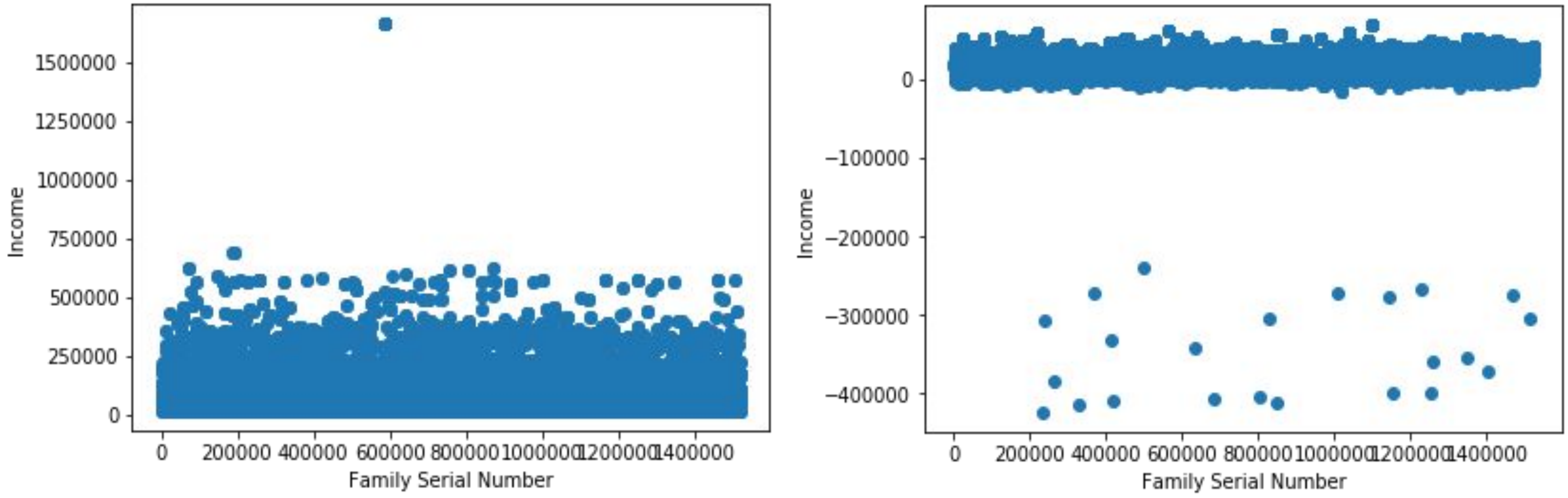
By: Xinchun Chen, Abhishek Nimmakayala, and David Amankwah

Introduction

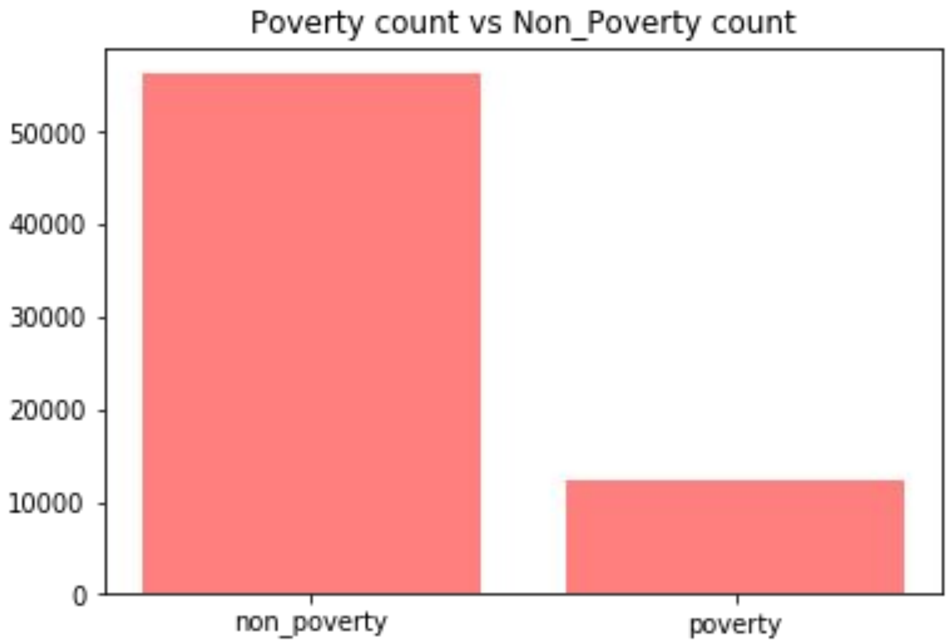
- Poverty is one of many topics that have been researched by Economists and Data Scientists.
- The main objective of this project is to predict the poverty status of families in New York using a decision tree model analysis.
- The dataset used is the NYCgov poverty measure, which is generated annually by the poverty research unit of the Mayor's Office of Economic Opportunity (NYC Opportunity).

Exploratory Data Analysis

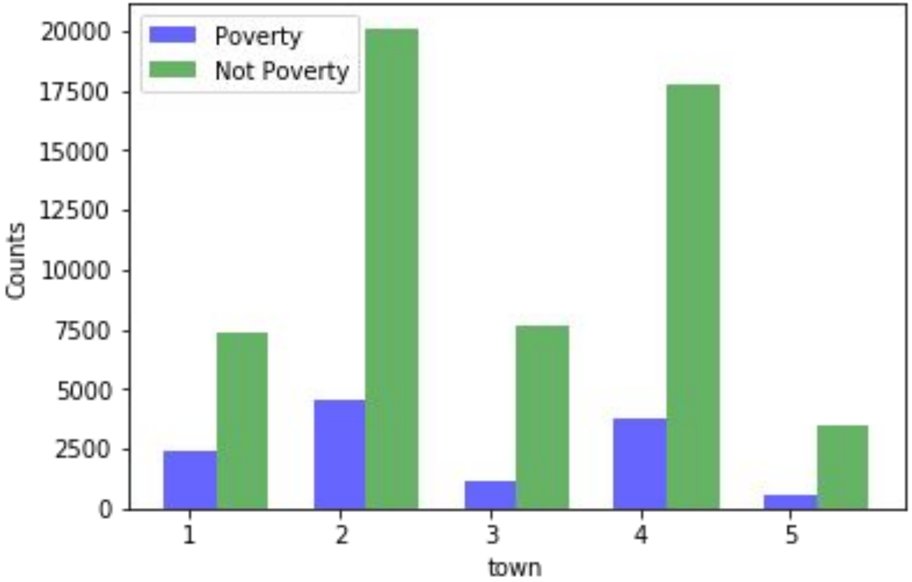
Yearly Income in Poverty vs Non Poverty Families



Exploratory Data Analysis - Cont’d



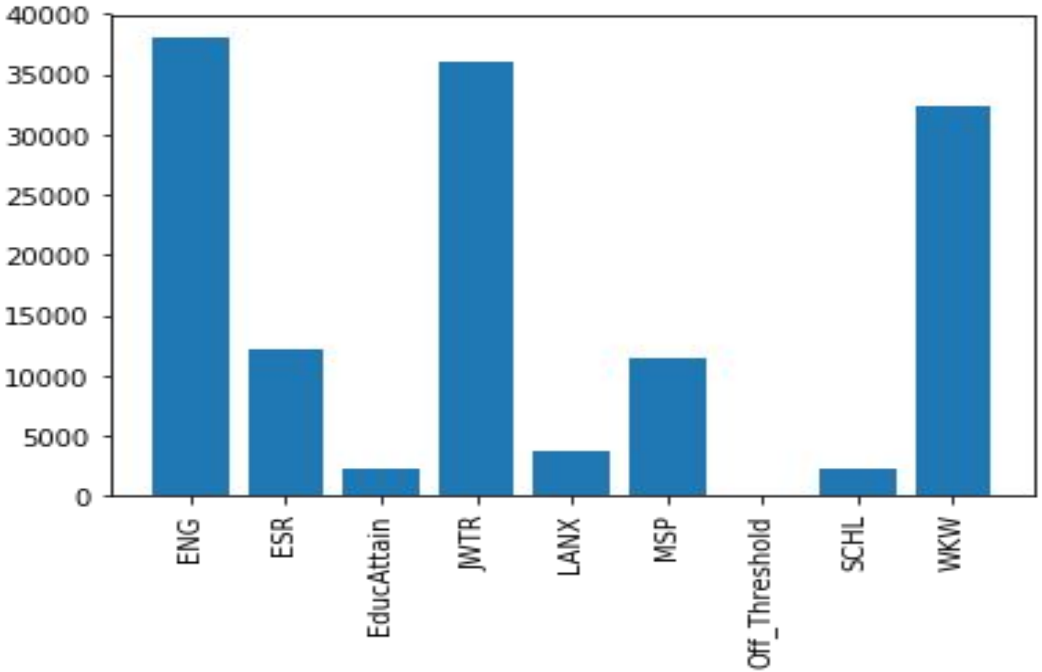
Exploratory Data Analysis - Cont'd



Graph presents the poverty and non-poverty count in different towns of New York.

1: Bronx
2: Brooklyn
3: Manhattan
4: Queens
5: Staten Island

Missing Data?



<u>Variable Keys</u>
ENG : English
ESR : Employment Status
EducAttain: Educational Attainment
JWTR: Means of Transportation to work
LANX: Other Languages spoken home
MSP: Married, spouse present/absent
Off_Threshold: Fed. Poverty Threshold
SCHL: Educational Attainment
WKW: Weeks worked in the past 12 months

```
[('ENG', 38015), ('ESR', 12198), ('EducAttain', 2240), ('JWTR', 36009), ('LANX', 3738), ('MSP', 11419), ('Off_Threshold', 2),
('SCHL', 2241), ('WKW', 32406)]
```

Before Imputation vs After Imputation

```
ENG
1.0    16275
2.0     6832
3.0     5366
4.0     2156
Name: ENG, dtype: int64
ESR
1.0    32618
2.0      875
3.0     2253
4.0       17
6.0    20683
Name: ESR, dtype: int64
EducAttain
1.0    20618
2.0    13081
3.0    12397
4.0    20308
Name: EducAttain, dtype: int64
```

```
ENG
1.0    39370
2.0    14161
3.0    10547
4.0     4566
Name: ENG, dtype: int64
ESR
1.0    39317
2.0     1136
3.0     2849
4.0       18
6.0    25324
Name: ESR, dtype: int64
EducAttain
1.0    21830
2.0    13315
3.0    12639
4.0    20860
Name: EducAttain, dtype: int64
```

Decision Tree:

- Preprocessing:
  1. Feature selection
  2. Instance selection
  3. Imbalance classes
- Implementing tree
- Random Forest

Decision Tree

Feature selection:

- Redundant and irrelevant variables: UnitID, Agecateg, Federal Poverty Status, Total Income, etc.
- The number of attributes down to 60 from 79.

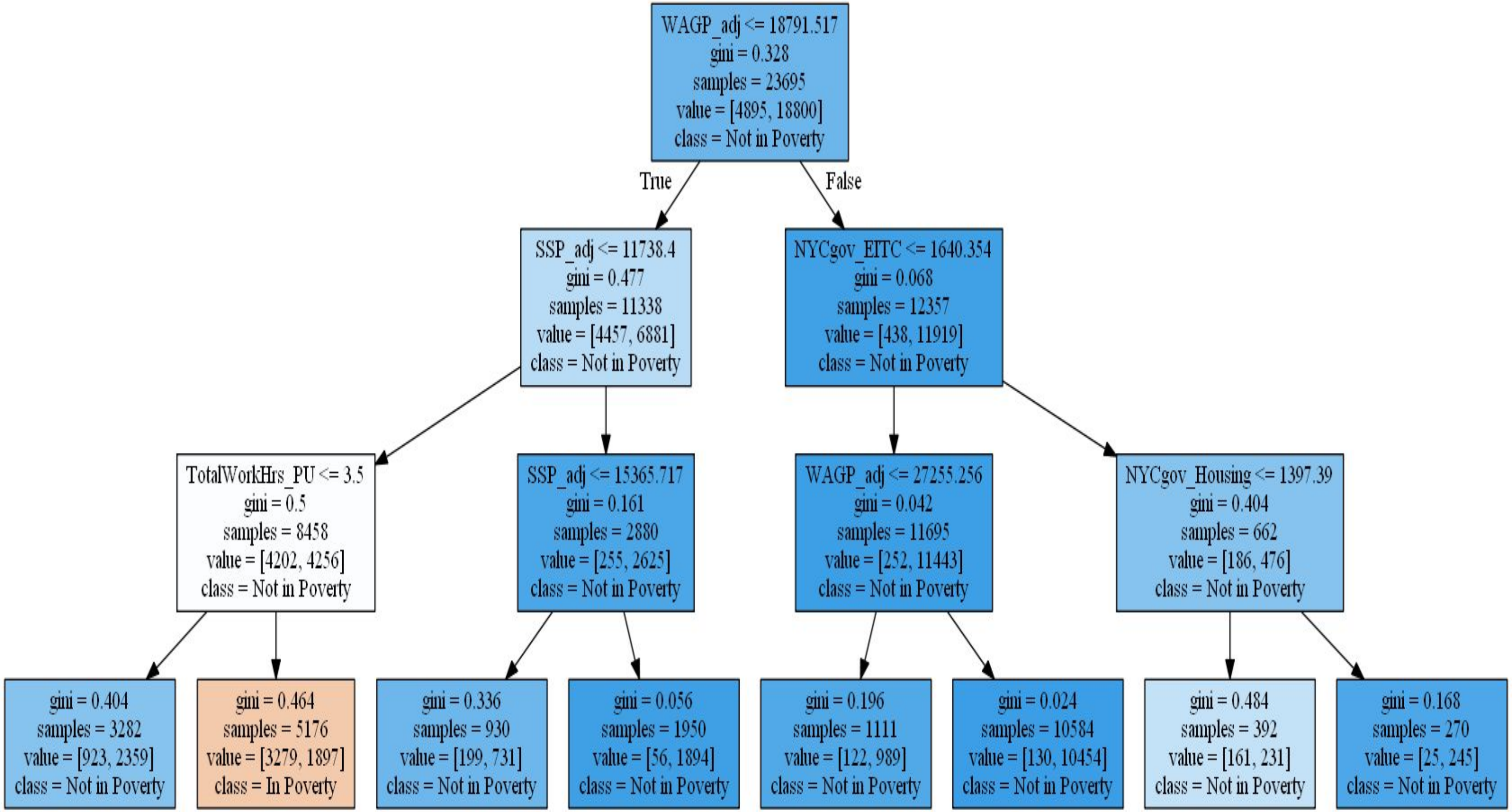
Instance selection:

- Remove all other household members except the head of the household.
- The number of observations down to 30000 from 68000

Decision Tree

Imbalanced classes:

- Imbalanced classes in the training set will cause the model bias toward the dominated class.
- The dataset contains about 80% of non-poverty individuals, which dominate the poverty class.



WAGP_adj <= 18791.517
gini = 0.328
samples = 23695
value = [4895, 18800]
class = Not in Poverty
True
False
SSP_adj <= 11738.4
gini = 0.477
samples = 11338
value = [4457, 6881]
class = Not in Poverty
NYCgov_EITC <= 1640.354
gini = 0.068
samples = 12357
value = [438, 11919]
class = Not in Poverty
TotalWorkHrs_PU <= 3.5
gini = 0.5
samples = 8458
value = [4202, 4256]
class = Not in Poverty
SSP_adj <= 15365.717
gini = 0.161
samples = 2880
value = [255, 2625]
class = Not in Poverty
WAGP_adj <= 27255.256
gini = 0.042
samples = 11695
value = [252, 11443]
class = Not in Poverty
NYCgov_Housing <= 1397.39
gini = 0.404
samples = 662
value = [186, 476]
class = Not in Poverty
gini = 0.404
samples = 3282
value = [923, 2359]
class = Not in Poverty
gini = 0.464
samples = 5176
value = [3279, 1897]
class = In Poverty
gini = 0.336
samples = 930
value = [199, 731]
class = Not in Poverty
gini = 0.056
samples = 1950
value = [56, 1894]
class = Not in Poverty
gini = 0.196
samples = 1111
value = [122, 989]
class = Not in Poverty
gini = 0.024
samples = 10584
value = [130, 10454]
class = Not in Poverty
gini = 0.484
samples = 392
value = [161, 231]
class = Not in Poverty
gini = 0.168
samples = 270
value = [25, 245]
class = Not in Poverty

Decision Tree

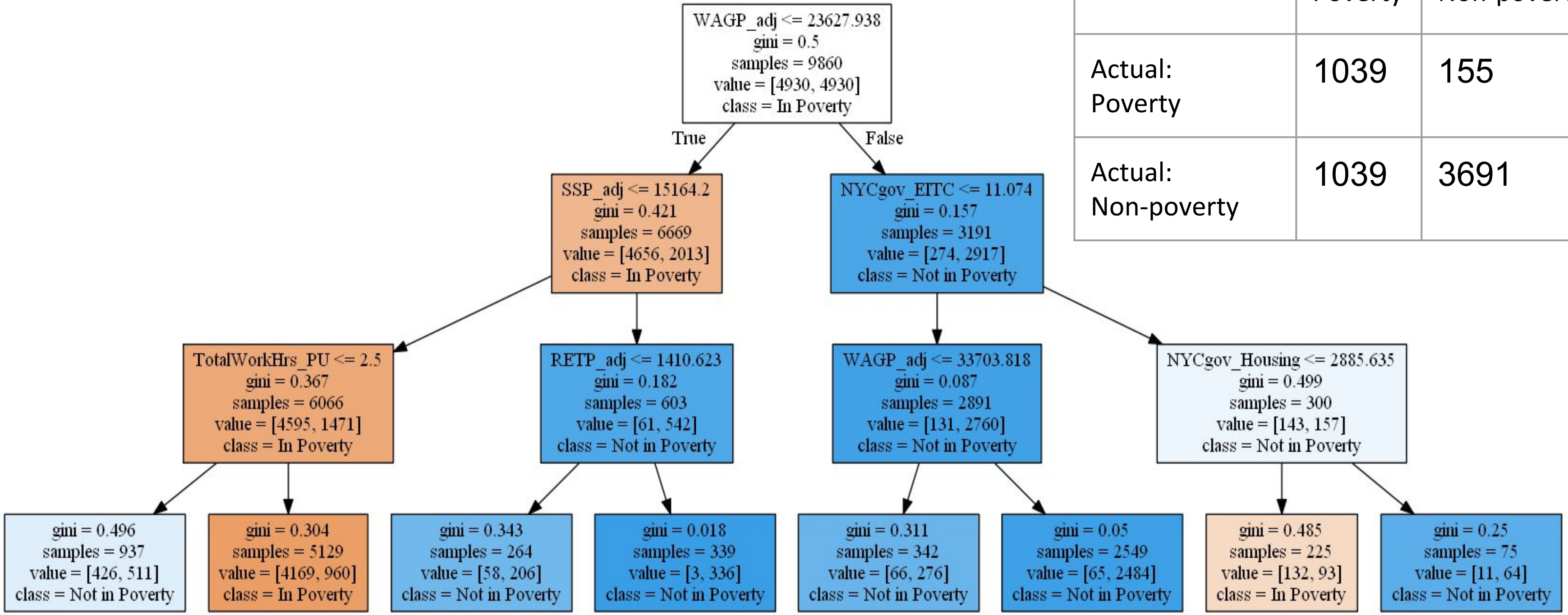
Imbalanced classes:

- Use *resample* function to randomly draw from non-poverty group in the training set without replacement, downsize the dominated class.
- The new training set would have more balanced distribution between two classes.

Decision Tree

Accuracy on training set: 0.83
Accuracy on testing set: 0.79

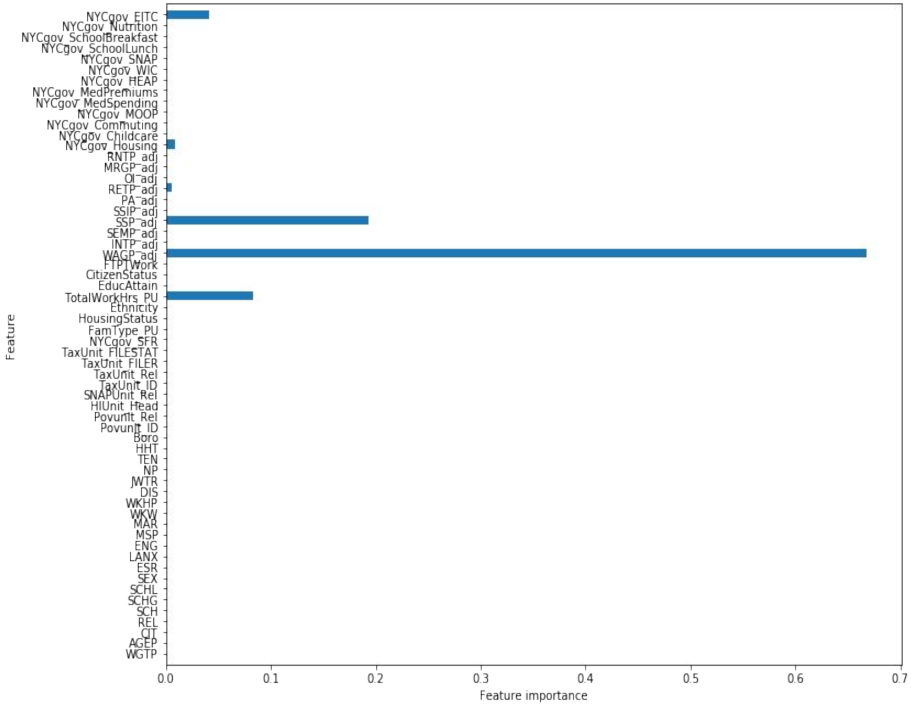
	Predict: Poverty	Predict: Non-poverty
Actual: Poverty	1039	155
Actual: Non-poverty	1039	3691

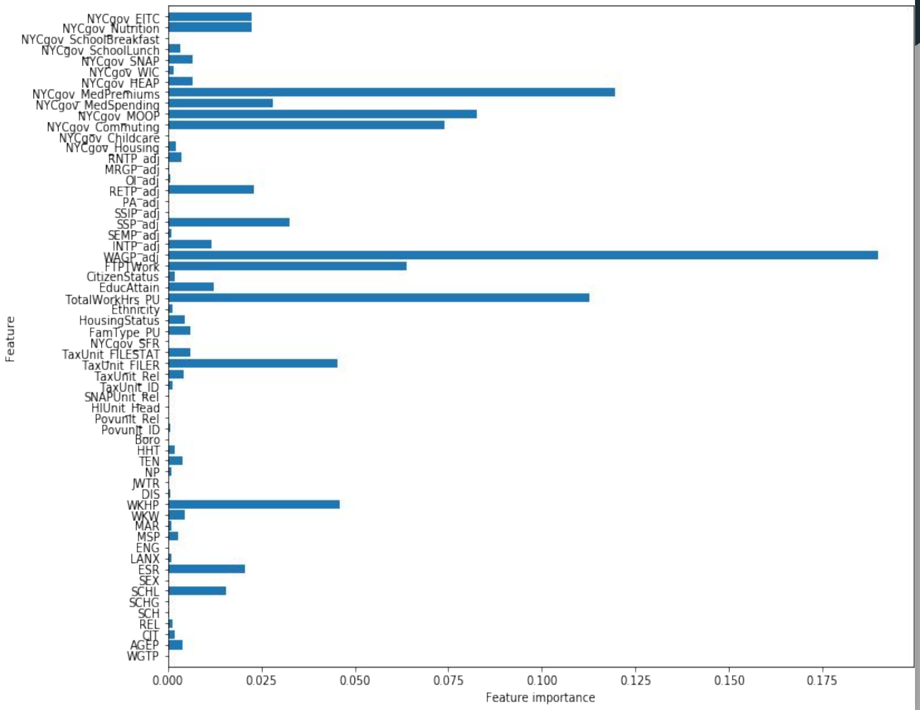


Random Forest

- Single decision tree would produce unstable result due to the variation of the data.
- Random Forest combine many unique trees to produce a more stable and robust result.
- We created a random forest that contains 1000 tree.

Random Forest

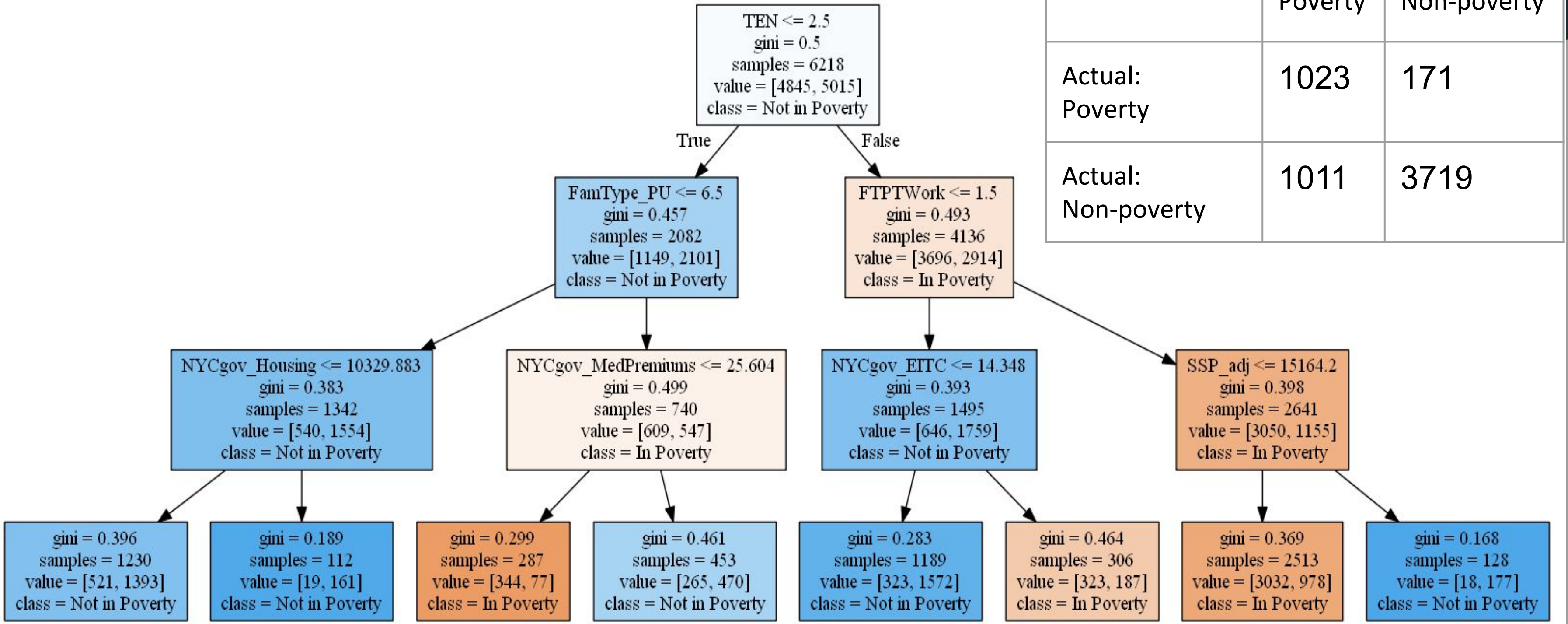




Random Forest

Accuracy on training set: 0.83
Accuracy on testing set: 0.80

	Predict: Poverty	Predict: Non-poverty
Actual: Poverty	1023	171
Actual: Non-poverty	1011	3719



Conclusion and Limitation

- Base on the feature importance, the most important feature on deciding people’s poverty status is the wage for past 12 months, followed by medical premium cost, and working hour.
- There are potentially high correlation between wage, medical premium cost and working hour(person has less wage income may also has less working hour and medical premium spending)
- This correlation will cause the model underestimate the impact of other independent variables.
- Run correlation map for all the attributes and identify the collinearity.

Reference:

Mayor's Office For Economic Opportunity. NYCgov Poverty Measure Data(2016). Retrieved from: https://data.cityofnewyork.us/City-Government/NYCgov-Poverty-Measure-Data-2016-/y9gu-cxxw

"How to Handle Imbalanced Classes in Machine Learning"(July 5,2017), EliteDataScience. Retrieved from:https://elitedatascience.com/imbalanced-classes

THANK YOU!!