DS - 670 –Capstione Big Data & Business Analytics

Mir Ahmed, Kaushik Raj Devarashetty, Madhuri koganti, leonsteen sterling

Project Proposal: Predicting House Prices and Identifying Ideal Locations in NJ

Saint Peter’s University | 2641 John F. Kennedy Blvd, Jersey City, NJ 07306

2023

# Project Proposal: Predicting House Prices and Identifying Ideal Locations in NJ

# Introduction

In this project, we aim to build two types of predictive models using various machine learning algorithms to predict the house price for purchase and identify ideal locations for purchase or rent in New Jersey. The project will use publicly available datasets from various sources such as Zillow, Freddie Mac, NJ Department of Education, and the Census Bureau. The project's primary goal is to help real estate investors and buyers as well as new residents and potential movers make informed decisions based on the models’ predictions.

# Project Group

Kaushik Raj Devarashetty

Madhuri Koganti

Mir Ahmed

Leonsteen Sterling

# Problem Statement

Moving to a new state can be a challenging task, especially when trying to determine the ideal location to live. People who are planning to move to New Jersey may not have enough knowledge about the state and its counties, which can make it difficult to determine where to live. Additionally, there are various factors to consider, such as the cost of living, crime rate, schools, and accessibility to amenities, which can be overwhelming to evaluate.

To tackle this challenge, we aim to develop a recommendation system that will assist people who are moving to NJ from other states in finding the appropriate county to rent or own a home using predictive modeling. The system will collect and analyze data from various sources such as real estate listings, census data, and public records to identify key factors that influence the quality of life in each county. We will use machine learning algorithms to predict the best counties that match an individual's preferences and needs, such as number of bedrooms, quality of schools, crime rate, and cost of living.

From a research project perspective, we would like our problem statement to be as follows:

**What type of crime has the most impact on house prices for a specific county?**

# Data Sources

## Zillow Home Value Index Data:

The dataset provides information on the Zillow Home Value Index by county, city, or ZIP code, which would be used as the target variable for building regression and/or classification models.

<https://www.zillow.com/research/data/>

## Mortgage Value Data:

This dataset provides information on the average mortgage rates for different regions in NJ, which could be used as a predictor variable in the regression and/or classification models.

[http://www.freddiemac.com/pmms/#](http://www.freddiemac.com/pmms/)

## Tax Rate Data:

This dataset provides information on the tax rates for different regions in NJ, which could be used as a predictor variable in the regression and/or classification models.

<https://www.state.nj.us/treasury/taxation/lpt/statdata.shtml>

## County Lines and Shape Data (GeoJSON):

This dataset provides geographic boundaries for different counties in NJ and could be used to visualize location-based features.

<http://data.ci.newark.nj.us/dataset/new-jersey-counties-polygon/resource/95db8cad-3a8c-41a4-b8b1-4991990f07f3>

## NJ Department of Education Data for School Performance:

This dataset provides information on school ratings for different regions in NJ, which could be used as a predictor variable in the classification and/or regression models.

<https://www.schooldigger.com/go/NJ/schoolrank.aspx>

## Crime Data:

These datasets provide information on crime rates for different regions in NJ, which could be used as a predictor variable in the classification and/or regression models.

For 2017: <https://ucr.fbi.gov/crime-in-the-u.s>

For 2018,2019,2020: <https://www.njsp.org/ucr/current-crime-data.shtml>

## Poverty and Median Income Data:

This dataset provides information on poverty and median income rates for different regions in NJ, which could be used as predictor variables in both the regression and classification models.

<https://www.census.gov/programs-surveys/saipe/data/api.html>

## NJ Population History by County

These datasets will give us historical population counts (estimated) from 2010 till 2021.

<https://www.nj.gov/labor/labormarketinformation/demographics/population-household-estimates/>

## NJ Municipalities by County

These datasets will give us all municipalities by county for NJ.

<https://data.nj.gov/Reference-Data/Municipalities-of-New-Jersey/k9xb-zgh4>

## NJ Food Desert – USDA Foot Atlas

This dataset will provide us flags for low access areas (food deserts) by county, which could be used as a predictor variable in the classification and/or regression models.

<https://www.ers.usda.gov/data-products/food-access-research-atlas/>

## NJ Area Deprivation Index – Neighborhood Atlas

The Area Deprivation Index (ADI) is based on a measure created by the Health Resources & Services Administration (HRSA) over three decades ago, and has since been refined, adapted, and validated to the Census Block Group neighborhood level by Amy Kind, MD, PhD and her research team at the University of Wisconsin-Madison. It allows for rankings of neighborhoods by socioeconomic disadvantage in a region of interest (e.g., at the state or national level). It includes factors for the theoretical domains of income, education, employment, and housing quality. It can be used to inform health delivery and policy, especially for the most disadvantaged neighborhood groups. "Neighborhood" is defined as a Census Block Group.

<https://www.neighborhoodatlas.medicine.wisc.edu/download>

# Methodology

## Regression Model:

For the regression model, we will use the Zillow Home Value Index as the target variable and use different algorithms such as Linear Regression, Random Forest, K-Nearest Neighbor, and Support Vector Machines to predict the house prices for different regions in NJ. The regression model will use predictor variables such as mortgage rates, tax rates, poverty rates, and median income rates. We will use the Root Mean Squared Error (RMSE) to evaluate the performance of the regression models.

## Classification Model:

For the classification model, we will use various algorithms such as Logistic Regression, Random Forest, K-Nearest Neighbor, Support Vector Machines, and Neural Networks to identify ideal locations for purchase or rent in NJ. The classification model will use predictor variables such as school performance metrics, crime rates, poverty rates, and median income rates. We will use accuracy, precision, recall, and F1-score to evaluate the performance of the classification models.

# Expected Outcome

The expected outcome of this project is to develop an app with most possible accurate predictive models for real estate investment/ rentals in New Jersey. These models will provide insights and recommendations to potential investors/ new residents to make informed decisions about property investments/ rentals in the state. This project will also provide an opportunity to understand the impact of various factors such as school performance, crime rates, and poverty levels on housing prices and rental values in New Jersey.

# Exploratory Data Analysis

Below are the packages we have used on Jupyter notebook to create these visualizations:

* matplotlib
* seaborn
* plotly

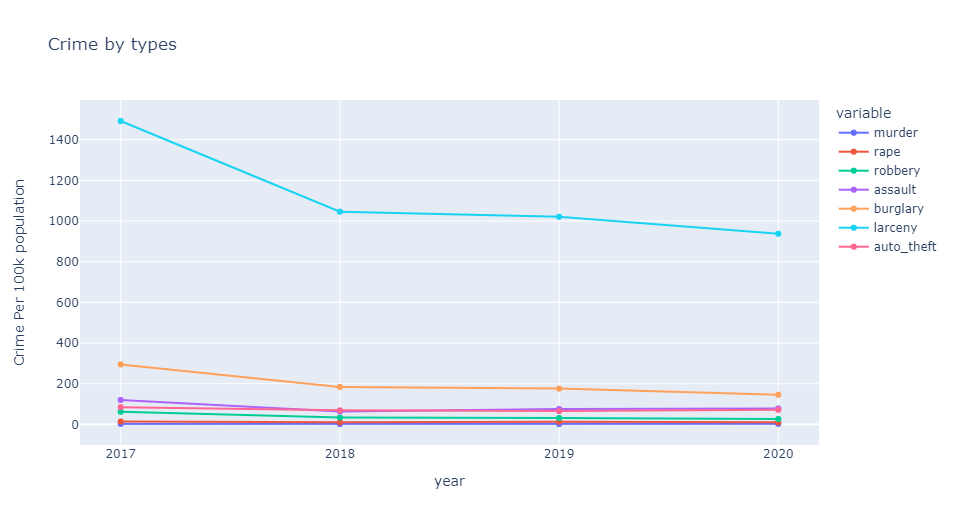
Besides these packages specific to visualization, we have also used the following packages for our exploratory data analysis:

* Pandas
* Numpy
* Pandas Profiling

First, we looked at each dataset separately and observed the trends. Below are the trend analyses of our datasets:

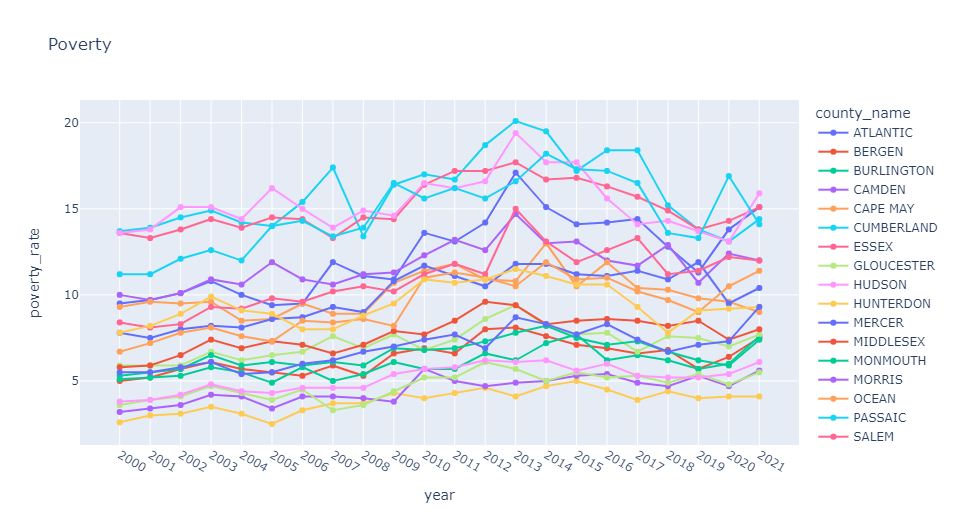
## Crime

Crime rates are going down in most of the counties across time, except Cape May.



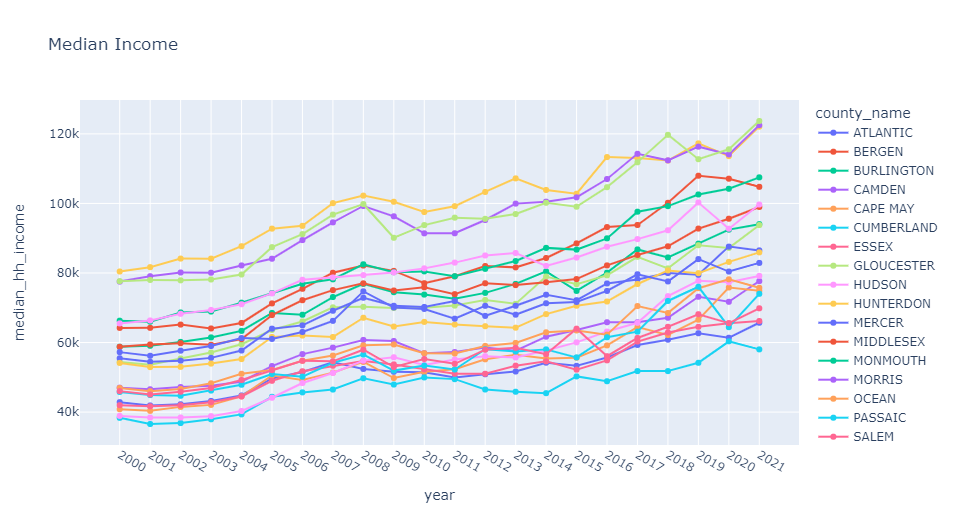
Most frequent violent crime in NJ is larceny which is also going down along with all other violent crimes over time.

## Poverty



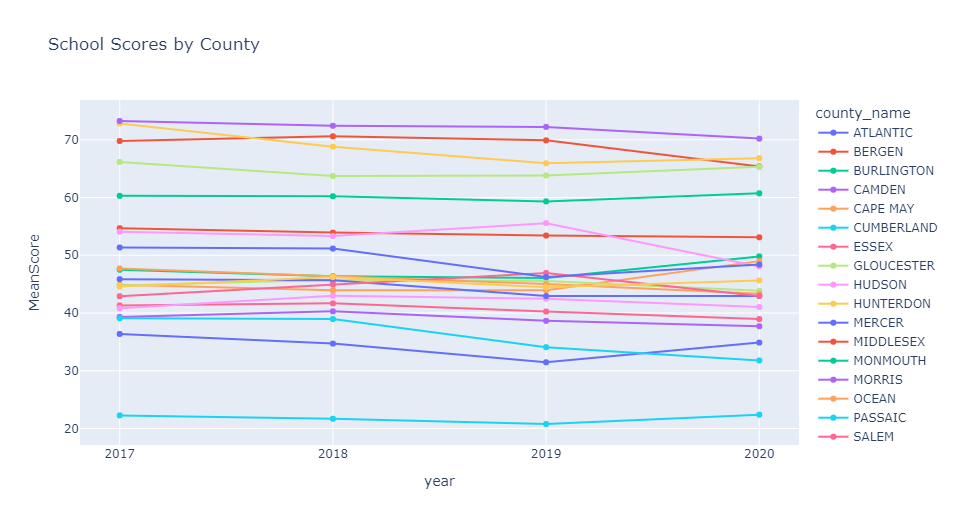
Poverty rates of counties are going up and down (zagged) from 2000 through 2021. But ultimately poverty rates in 2021 are more than what they were back in 2000.

## Median household income

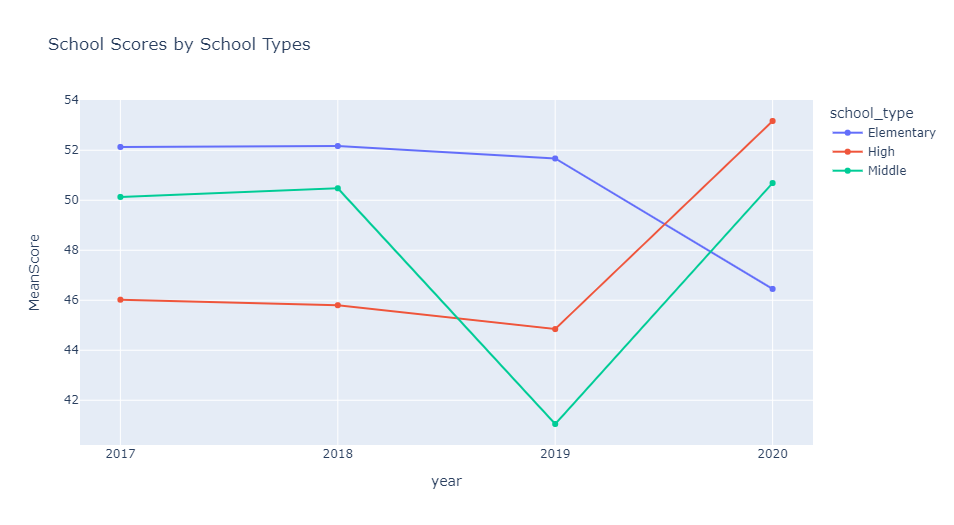


Median income is steadily going up across time.

## School performance

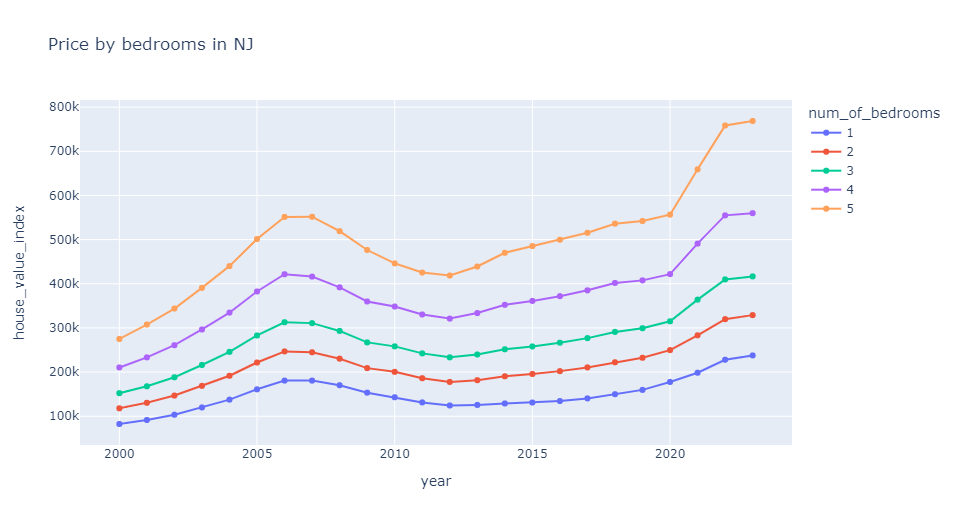


On average, NJ schools maintained their quality/ranking over time with a couple of exceptions like Passaic, Sussex, and Bergen.

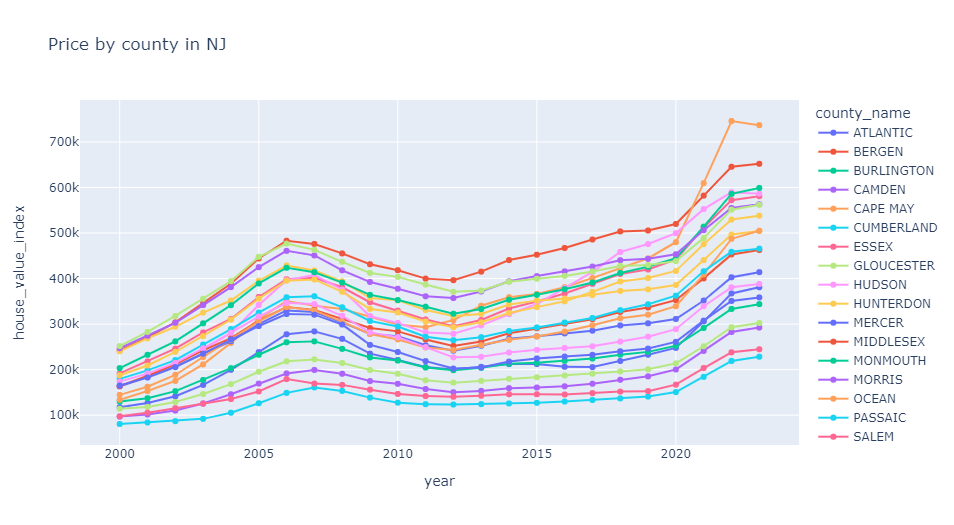


Mean scores for elementary schools went down. Middle schools saw a steep decline from 2018-2019 then got back up in 2020. High school scores went up by a lot from 2019 onwards.

## Zillow House Value Index

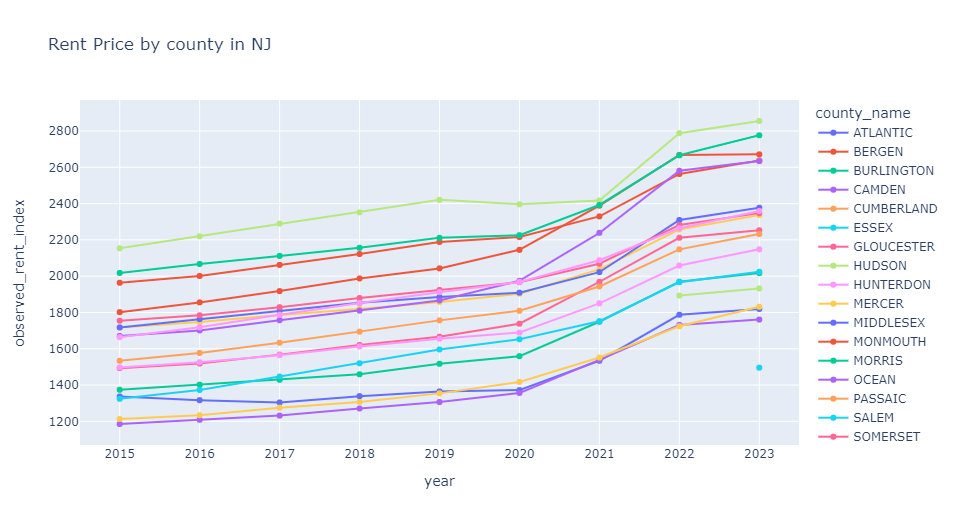


House prices for all bedrooms are going up over time except for a short decline from 2007 to 2012.



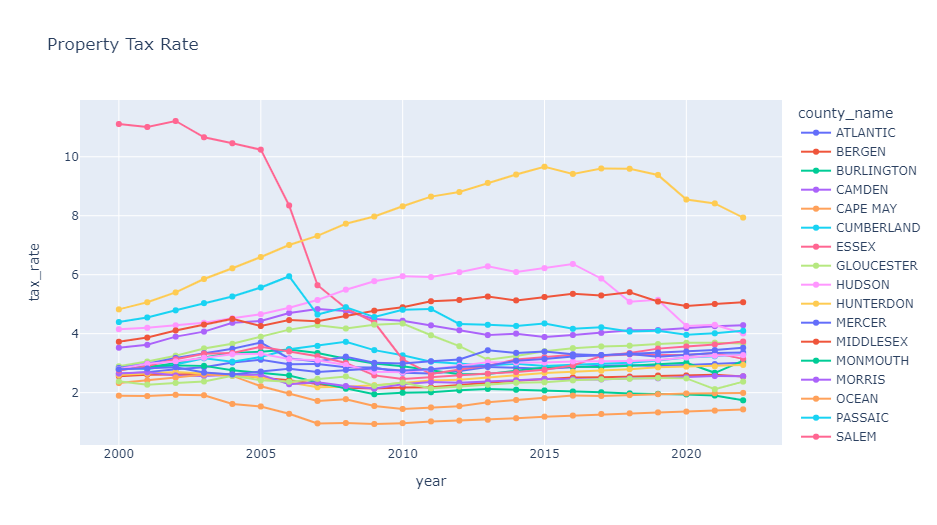
House prices for all counties are going up over time except for a short decline from 2007 to 2012.

## Zillow Observed Rent Index



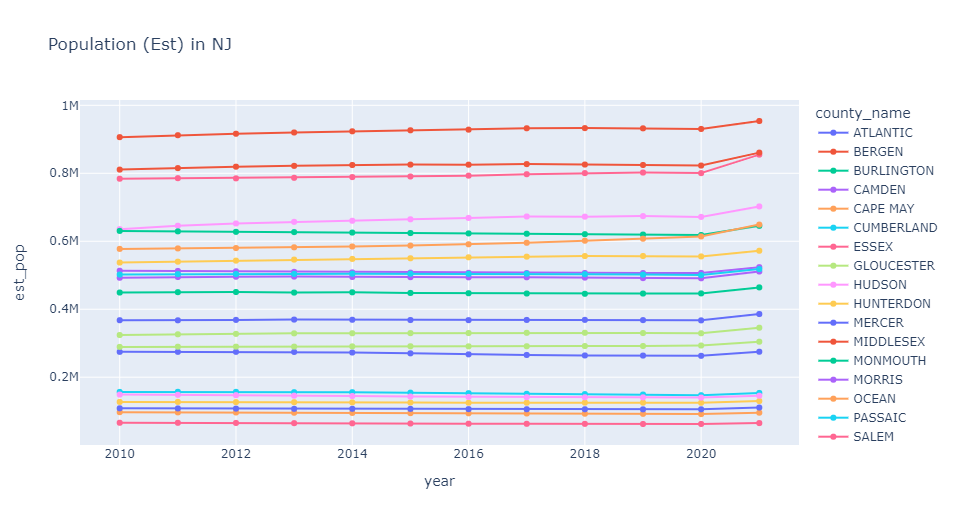
Rental prices for all counties are going up over time.

## Property Tax



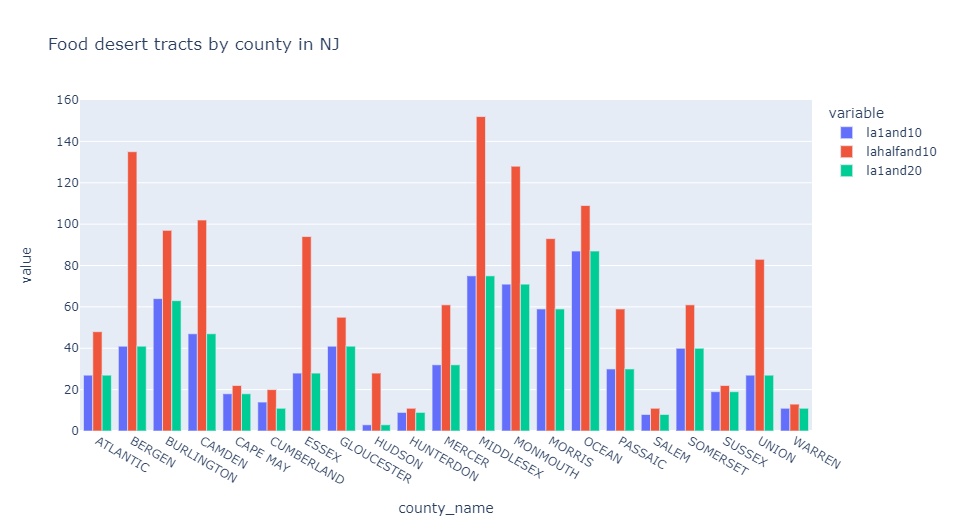
Essex property tax saw a huge decline from 2005 through 2011. Union property tax was on the rise from 2000 till 2015 and then it started to decline. Hudson also saw an increase in property tax up until 2016 and then it went down. The rest of the counties kind of maintained the same property tax across time.

## Population



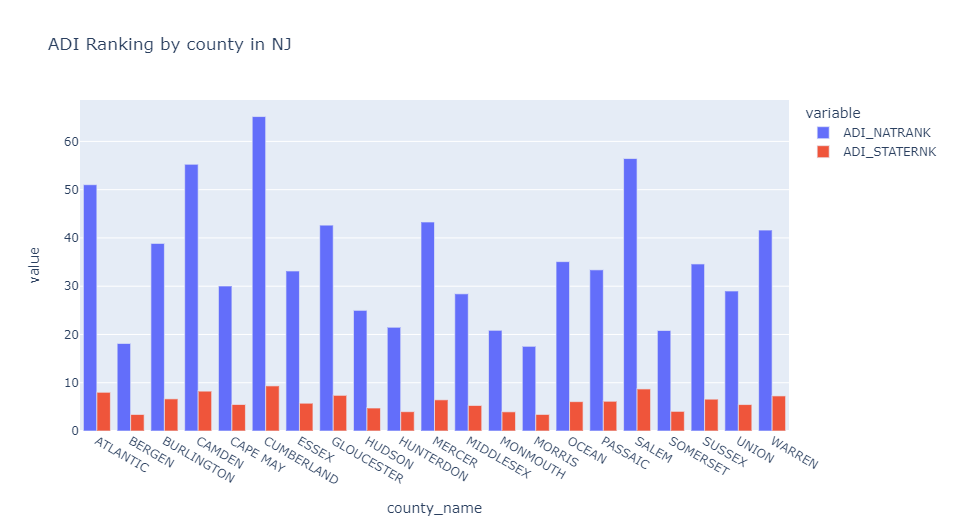
Population counts (estimated) are increasing very slightly over the years for all the counties.

## Food Desert



Food desert counts are mostly the same for both 1 mile radius in urban areas and 10 mile radius in rural areas and 1 mile radius in urban areas and 20 mile radius in rural areas. Counts are high for half mile radius in urban areas and 10 mile radius for rural areas for Middlesex, Bergen, and Monmouth.

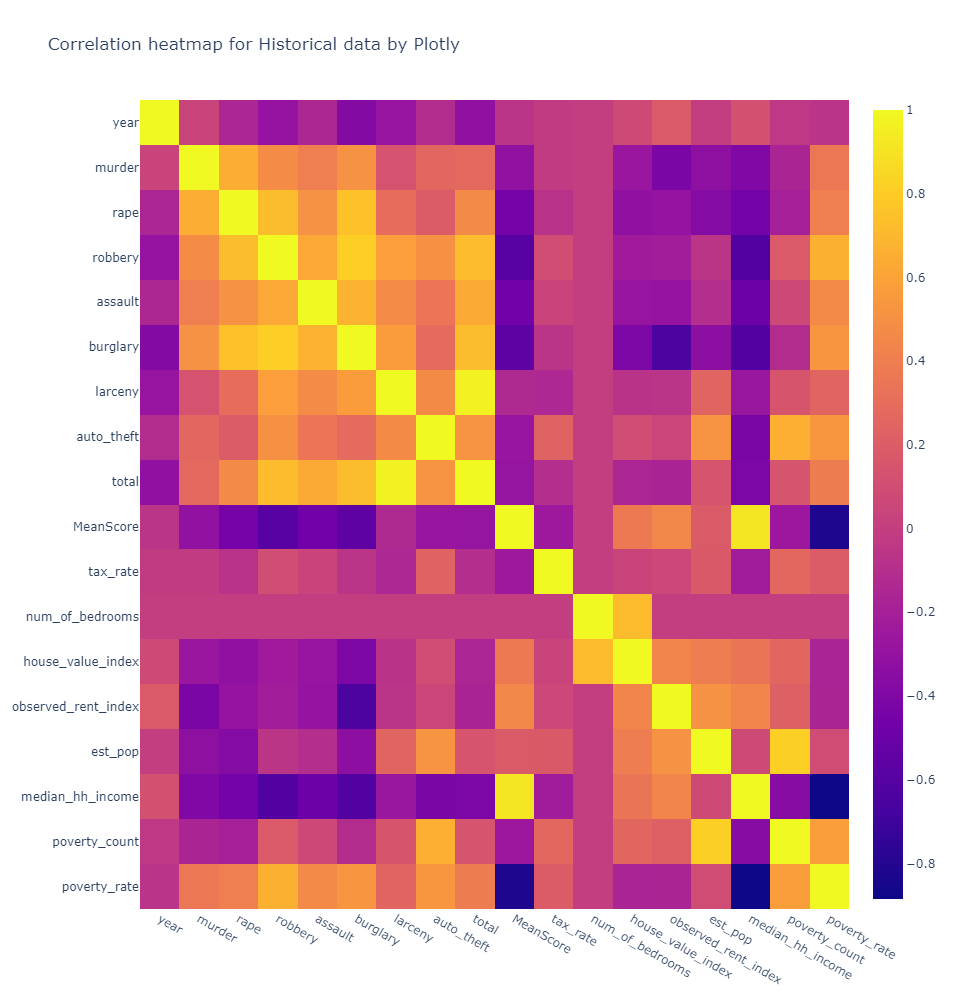
## Area Deprivation Index



Counties like Cumberland, Salem, Camden and Atlantic are ranked pretty far down the list of counties nationwide in terms of area deprivation index, which translates low socio economic and educational opportunities.

## Correlation heatmap for historical data

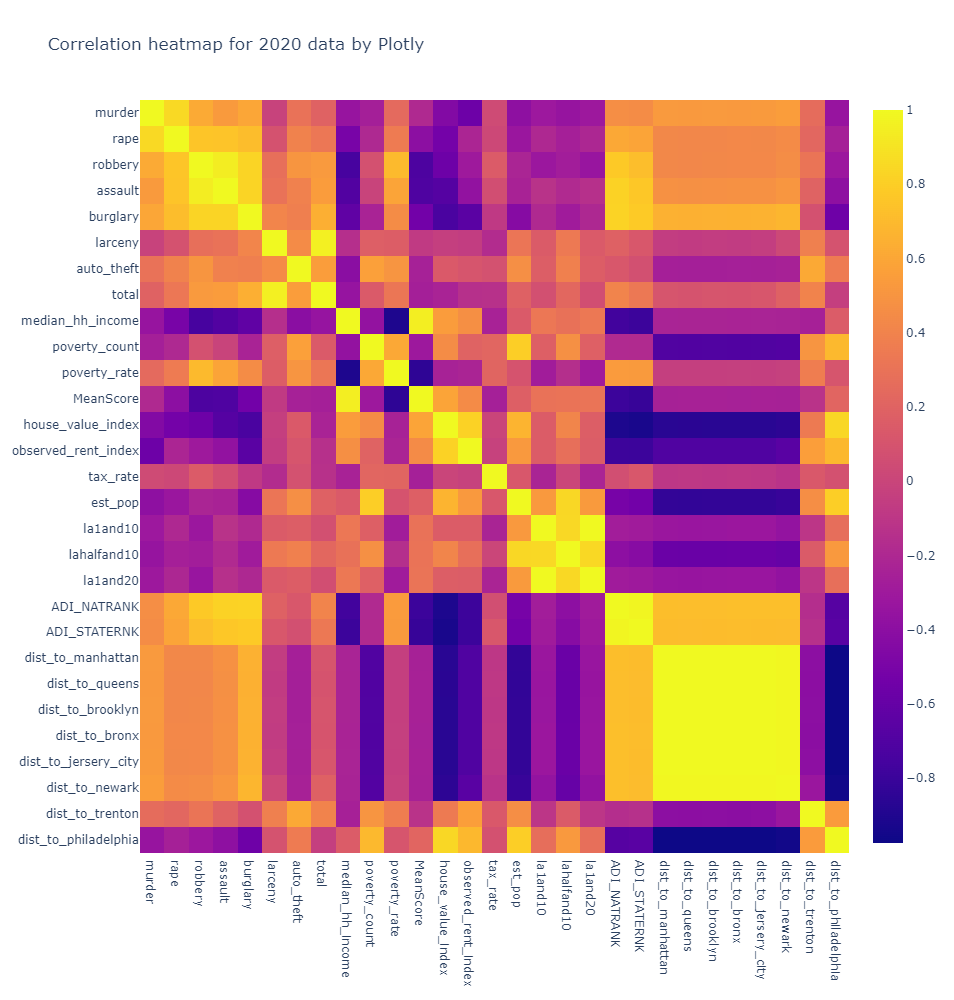
We merged all the data sets using county names and created a correlation heatmap which shows the correlations between all variables.



House value index has strong positive correlation with number of bedrooms and has weak positive correlation with school score, rent index, population, and median household income. It has a weak negative correlation with poverty rate but weak positive correlation with poverty count.

## Correlation heatmap for historical data

Then we just focused on data from 2020 as we had the most amount of data points as well as the latest for that year.



The House Price index has a strong negative correlation with distance to all major cities except Trenton and Philadelphia as well as Area Deprivation Index (ADI) - both national and state rankings and burglary. The House Price Index has a strong positive correlation with the Observed Rent Index and moderately strong positive correlation with population, median household income, school scores. The House Price Index also have somewhat of a weak to moderate positive correlation to food desert count at half mile radius.

The poverty rate has a strong negative correlation with school score and median household income and moderately strong positive correlation between different violent crimes. Out of those violent crimes, burglary has the biggest correlation with house price index. Next, we have assault which also has a moderate negative correlation with house prices.

# Answer to Our Problem Statement

Based on our exploratory analysis, we came to the following conclusion:

**The number of burglaries in a county has the most adverse effect on Zillow house price index.**

So, we should avoid counties for house purchase as well as rentals where the number of burglaries is high.

# Data Pre-Processing

For linear regression model, we joined all the datasets using county names and year and added minimum, maximum, and average values for numerical variables those are at a lower grain than county. Below is the list of columns used for training linear regression model:

Data columns (total 49 columns):

# Column Non-Null Count Dtype

--- ------ -------------- -----

0 county\_name 420 non-null object

1 year 420 non-null int64

2 min\_tax\_rate 420 non-null float64

3 avg\_tax\_rate 420 non-null float64

4 max\_tax\_rate 420 non-null float64

5 murder 420 non-null float64

6 rape 420 non-null float64

7 robbery 420 non-null float64

8 assault 420 non-null float64

9 burglary 420 non-null float64

10 larceny 420 non-null float64

11 auto\_theft 420 non-null float64

12 avg\_std\_cnt\_ele 420 non-null float64

13 avg\_exp\_ele 420 non-null float64

14 avg\_score\_ele 420 non-null float64

15 min\_std\_cnt\_ele 420 non-null int64

16 min\_exp\_ele 420 non-null float64

17 min\_score\_ele 420 non-null float64

18 max\_std\_cnt\_ele 420 non-null int64

19 max\_exp\_ele 420 non-null float64

20 max\_score\_ele 420 non-null float64

21 avg\_std\_cnt\_mid 420 non-null float64

22 avg\_exp\_mid 420 non-null float64

23 avg\_score\_mid 420 non-null float64

24 min\_std\_cnt\_mid 420 non-null int64

25 min\_exp\_mid 420 non-null float64

26 min\_score\_mid 420 non-null float64

27 max\_std\_cnt\_mid 420 non-null int64

28 max\_exp\_mid 420 non-null float64

29 max\_score\_mid 420 non-null float64

30 avg\_std\_cnt\_high 420 non-null float64

31 avg\_exp\_high 420 non-null float64

32 avg\_score\_high 420 non-null float64

33 min\_std\_cnt\_high 420 non-null int64

34 min\_exp\_high 420 non-null float64

35 min\_score\_high 420 non-null float64

36 max\_std\_cnt\_high 420 non-null int64

37 max\_exp\_high 420 non-null float64

38 max\_score\_high 420 non-null float64

39 est\_pop 420 non-null int64

40 apr\_30 420 non-null float64

41 points\_30 420 non-null float64

42 apr\_15 420 non-null float64

43 points\_15 420 non-null float64

44 num\_of\_bedrooms 420 non-null int64

45 house\_value\_index 420 non-null float64

46 median\_hh\_income 420 non-null int64

47 poverty\_count 420 non-null int64

48 poverty\_rate 420 non-null float64

This dataset only has records from 2017 to 2020.

For all other models, we joined the following datasets because of their vast available history-

* Zillow house price index
* Population
* Poverty and median income
* Property tax
* Mortgage rate

Below is the list of variables used for training all other models:

Data columns (total 23 columns):

# Column Non-Null Count Dtype

--- ------ -------------- -----

0 county\_name 1260 non-null object

1 year 1260 non-null int64

2 num\_of\_bedrooms 1260 non-null int64

3 house\_value\_index 1260 non-null float64

4 county\_name 1260 non-null object

5 est\_pop 1260 non-null int64

6 year 1260 non-null int64

7 county\_name 1260 non-null object

8 median\_hh\_income 1260 non-null int64

9 poverty\_count 1260 non-null int64

10 poverty\_rate 1260 non-null float64

11 st\_abb 1260 non-null object

12 year 1260 non-null int64

13 state\_code 1260 non-null object

14 county\_code 1260 non-null object

15 county\_name 1260 non-null object

16 year 1260 non-null int64

17 tax\_rate 1260 non-null float64

18 year 1260 non-null int64

19 apr\_30 1260 non-null float64

20 points\_30 1260 non-null float64

21 apr\_15 1260 non-null float64

22 points\_15 1260 non-null float64

This dataset has records from 2010 to 2021.

# Regression modeling – House Price Prediction

## Linear Regression Model

Before we started with our modeling, we checked whether we have any outliers on our pre-processed dataset using boxplots in terms of house prices against number of bedrooms:

Chart, box and whisker chart

Description automatically generated

We do have a couple of outliers, but we did not exclude them from our modeling because:

1. They are not drastically variant from the population
2. They might be prices for condos in gentrified locations.

As we have a lot of variables to work with, we took the approach for stepwise linear regression which only takes in features those are statistically important to the response variable and at the same time not compromising the accuracy of the model. In our case house price index is the response variable and all other features will be explanatory variables.

Below is the python function for stepwise linear regression:

**import** statsmodels.formula.api **as** smf

**def** forward\_selected(data, response):

"""Linear model designed by forward selection.

Parameters:

-----------

data : pandas DataFrame with all possible predictors and response

response: string, name of response column in data

Returns:

--------

model: an "optimal" fitted statsmodels linear model

with an intercept

selected by forward selection

evaluated by adjusted R-squared

"""

remaining **=** set(data**.**columns)

remaining**.**remove(response)

selected **=** []

current\_score, best\_new\_score **=** 0.0, 0.0

**while** remaining **and** current\_score **==** best\_new\_score:

scores\_with\_candidates **=** []

**for** candidate **in** remaining:

formula **=** "{} ~ {}"**.**format(response,

' + '**.**join(selected **+** [candidate]))

score **=** smf**.**ols(formula, data)**.**fit()**.**rsquared\_adj

scores\_with\_candidates**.**append((score, candidate))

scores\_with\_candidates**.**sort()

best\_new\_score, best\_candidate **=** scores\_with\_candidates**.**pop()

**if** current\_score **<** best\_new\_score:

remaining**.**remove(best\_candidate)

selected**.**append(best\_candidate)

current\_score **=** best\_new\_score

formula **=** "{} ~ {}"**.**format(response,

' + '**.**join(selected))

model **=** smf**.**ols(formula, data)**.**fit()

**return** model

When we passed our pre-processed data and all other required parameters the function selected this equation to be optimal for our model:

Selected features for the model:

house\_value\_index ~ num\_of\_bedrooms + county\_name + year + avg\_tax\_rate

----------------------------------

Adjusted R squared for the model:

0.8480220894888519

Below is the summary of the model:

OLS Regression Results

==============================================================================

Dep. Variable: house\_value\_index R-squared: 0.856

Model: OLS Adj. R-squared: 0.848

Method: Least Squares F-statistic: 102.7

Date: Tue, 04 Apr 2023 Prob (F-statistic): 4.02e-151

Time: 15:53:55 Log-Likelihood: -5280.8

No. Observations: 420 AIC: 1.061e+04

Df Residuals: 396 BIC: 1.071e+04

Df Model: 23

Covariance Type: nonrobust

=============================================================================================

coef std err t P>|t| [0.025 0.975]

---------------------------------------------------------------------------------------------

Intercept -2.388e+07 6.42e+06 -3.720 0.000 -3.65e+07 -1.13e+07

county\_name[T.BERGEN] 2.694e+05 2.4e+04 11.226 0.000 2.22e+05 3.17e+05

county\_name[T.BURLINGTON] 1.133e+04 2.28e+04 0.498 0.619 -3.34e+04 5.61e+04

county\_name[T.CAMDEN] -1.642e+04 3.39e+04 -0.484 0.629 -8.31e+04 5.03e+04

county\_name[T.CAPE MAY] 1.749e+05 4.08e+04 4.292 0.000 9.48e+04 2.55e+05

county\_name[T.CUMBERLAND] -8.254e+04 2.3e+04 -3.594 0.000 -1.28e+05 -3.74e+04

county\_name[T.ESSEX] 1.968e+05 2.4e+04 8.213 0.000 1.5e+05 2.44e+05

county\_name[T.GLOUCESTER] -1.01e+04 2.72e+04 -0.372 0.710 -6.35e+04 4.33e+04

county\_name[T.HUDSON] 2.844e+05 5.11e+04 5.567 0.000 1.84e+05 3.85e+05

county\_name[T.HUNTERDON] 1.463e+05 2.29e+04 6.401 0.000 1.01e+05 1.91e+05

county\_name[T.MERCER] 8.052e+04 2.39e+04 3.366 0.001 3.35e+04 1.28e+05

county\_name[T.MIDDLESEX] 1.56e+05 5.28e+04 2.953 0.003 5.21e+04 2.6e+05

county\_name[T.MONMOUTH] 1.711e+05 3.03e+04 5.649 0.000 1.12e+05 2.31e+05

county\_name[T.MORRIS] 2.049e+05 2.45e+04 8.358 0.000 1.57e+05 2.53e+05

county\_name[T.OCEAN] 6.919e+04 3.08e+04 2.243 0.025 8546.729 1.3e+05

county\_name[T.PASSAIC] 1.373e+05 3.35e+04 4.099 0.000 7.15e+04 2.03e+05

county\_name[T.SALEM] -6.029e+04 2.5e+04 -2.413 0.016 -1.09e+05 -1.12e+04

county\_name[T.SOMERSET] 1.917e+05 2.46e+04 7.794 0.000 1.43e+05 2.4e+05

county\_name[T.SUSSEX] 4.647e+04 2.31e+04 2.012 0.045 1064.663 9.19e+04

county\_name[T.UNION] 3.116e+05 1.36e+05 2.292 0.022 4.43e+04 5.79e+05

county\_name[T.WARREN] 2.809e+04 2.44e+04 1.151 0.250 -1.99e+04 7.61e+04

num\_of\_bedrooms 9.368e+04 2482.491 37.735 0.000 8.88e+04 9.86e+04

year 1.184e+04 3176.188 3.727 0.000 5593.728 1.81e+04

avg\_tax\_rate -2.236e+04 2.11e+04 -1.061 0.289 -6.38e+04 1.91e+04

==============================================================================

Omnibus: 85.411 Durbin-Watson: 1.267

Prob(Omnibus): 0.000 Jarque-Bera (JB): 209.520

Skew: 1.009 Prob(JB): 3.19e-46

Kurtosis: 5.811 Cond. No. 3.69e+06

==============================================================================

Notes:

[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.

[2] The condition number is large, 3.69e+06. This might indicate that there are strong multicollinearity or other numerical problems.

So, this stepwise linear regression model gives us 85% accuracy.

## Random Forest Regressor

Next, we tried random forest regression, where we used the other pre-processed dataset with more history.

Below is the model scores:

Training Data Score: 0.9937723196024478

Testing Data Score: 0.9707401005841938

Mean Squared Error: 806404319.2723241

Root Mean Squared Error: 28397.25900984678

The model performed really well compared to linear regression. But we have to keep in mind that random forest output is basically an average on outputs from all the decision trees and it might be prone to being skewed due to overfitting. We also tried hyper parameter tuning but the tuned model did not give us better accuracy. So, the base model was picked for our application.

# Classification modeling – County Recommendation

## Random Forest Classifier

After multiple trials of manually adding and/or removing features, we came up with the finalized list of features for this model:

* Poverty rate
* Median household income
* House price
* Year
* Number of bedrooms

Below is the feature importance scores from random forest classification model:

[(0.3609237487342005, 'poverty\_rate'),

(0.3103848368846969, 'median\_hh\_income'),

(0.15761129628656217, 'house\_value\_index'),

(0.12514809140252514, 'year'),

(0.04593202669201531, 'num\_of\_bedrooms')]

Below are the model scores:

Training Data Score: 1.0

Testing Data Score: 0.946031746031746

Below is the classification report:

precision recall f1-score support

ATLANTIC 0.85 0.85 0.85 13

BERGEN 1.00 1.00 1.00 15

BURLINGTON 1.00 0.94 0.97 16

CAMDEN 1.00 1.00 1.00 14

CAPE MAY 1.00 0.94 0.97 16

CUMBERLAND 0.94 1.00 0.97 16

ESSEX 0.89 0.84 0.86 19

GLOUCESTER 0.90 1.00 0.95 18

HUDSON 0.80 1.00 0.89 12

HUNTERDON 1.00 1.00 1.00 14

MERCER 1.00 1.00 1.00 12

MIDDLESEX 1.00 1.00 1.00 17

MONMOUTH 1.00 1.00 1.00 13

MORRIS 0.82 0.82 0.82 17

OCEAN 0.94 1.00 0.97 17

PASSAIC 1.00 0.77 0.87 13

SALEM 1.00 1.00 1.00 16

SOMERSET 0.85 0.79 0.81 14

SUSSEX 0.94 1.00 0.97 17

UNION 1.00 0.92 0.96 13

WARREN 1.00 1.00 1.00 13

accuracy 0.95 315

macro avg 0.95 0.95 0.95 315

weighted avg 0.95 0.95 0.95 315

Again, we tried hyper parameter tuning but the tuned model did not give us better accuracy. So, the base model was picked for our application.

## Deep Learning (Keras Sequential)

For this model, we used the keras sequential model builder from tensorflow package. We used label encoder to convert the county name column to encoded categorical column. We then used the following function to identify the number of decision nodes in each dense layer of our model:

**def** FindLayerNodesLinear(n\_layers, first\_layer\_nodes, last\_layer\_nodes):

layers **=** []

nodes\_increment **=** (last\_layer\_nodes **-** first\_layer\_nodes)**/** (n\_layers**-**1)

nodes **=** first\_layer\_nodes

**for** i **in** range(1, n\_layers**+**1):

layers**.**append(math**.**ceil(nodes))

nodes **=** nodes **+** nodes\_increment

**return** layers

We wanted 4 layers to our model with 11 features and 21 unique outputs (21 unique county names). So, the function looked like this with all our parameters:

FindLayerNodesLinear(4,11,21)

The output to the function is:

[11, 15, 18, 21]

Based on this output, we created the dense layers as follows:

*# Define the model*

deep\_model **=** Sequential()

*# Add first layer*

deep\_model**.**add(Dense(units**=**11, activation**=**'relu', input\_dim**=**11))

*# Add second layer (deep)*

deep\_model**.**add(Dense(units**=**15, activation**=**'relu'))

*# Add third layer (deep)*

deep\_model**.**add(Dense(units**=**18, activation**=**'relu'))

*# Add output layer*

deep\_model**.**add(Dense(units**=**21, activation**=**'softmax'))

Below is the model summary:

Model: "sequential\_1"

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Layer (type) Output Shape Param #

=================================================================

dense\_1 (Dense) (None, 11) 132

dense\_2 (Dense) (None, 15) 180

dense\_3 (Dense) (None, 18) 288

dense\_4 (Dense) (None, 21) 399

=================================================================

Total params: 999

Trainable params: 999

Non-trainable params: 0

Then we compile the model using the following parameters and trained it via the following code:

*# Compile and fit the model*

deep\_model**.**compile(optimizer**=**'adam',

loss**=**'categorical\_crossentropy',

metrics**=**['accuracy'])

*# Fit the model to the training data*

deep\_model**.**fit(

X\_train\_scaled,

y\_train\_categorical,

epochs**=**100,

shuffle**=True**,

verbose**=**2)

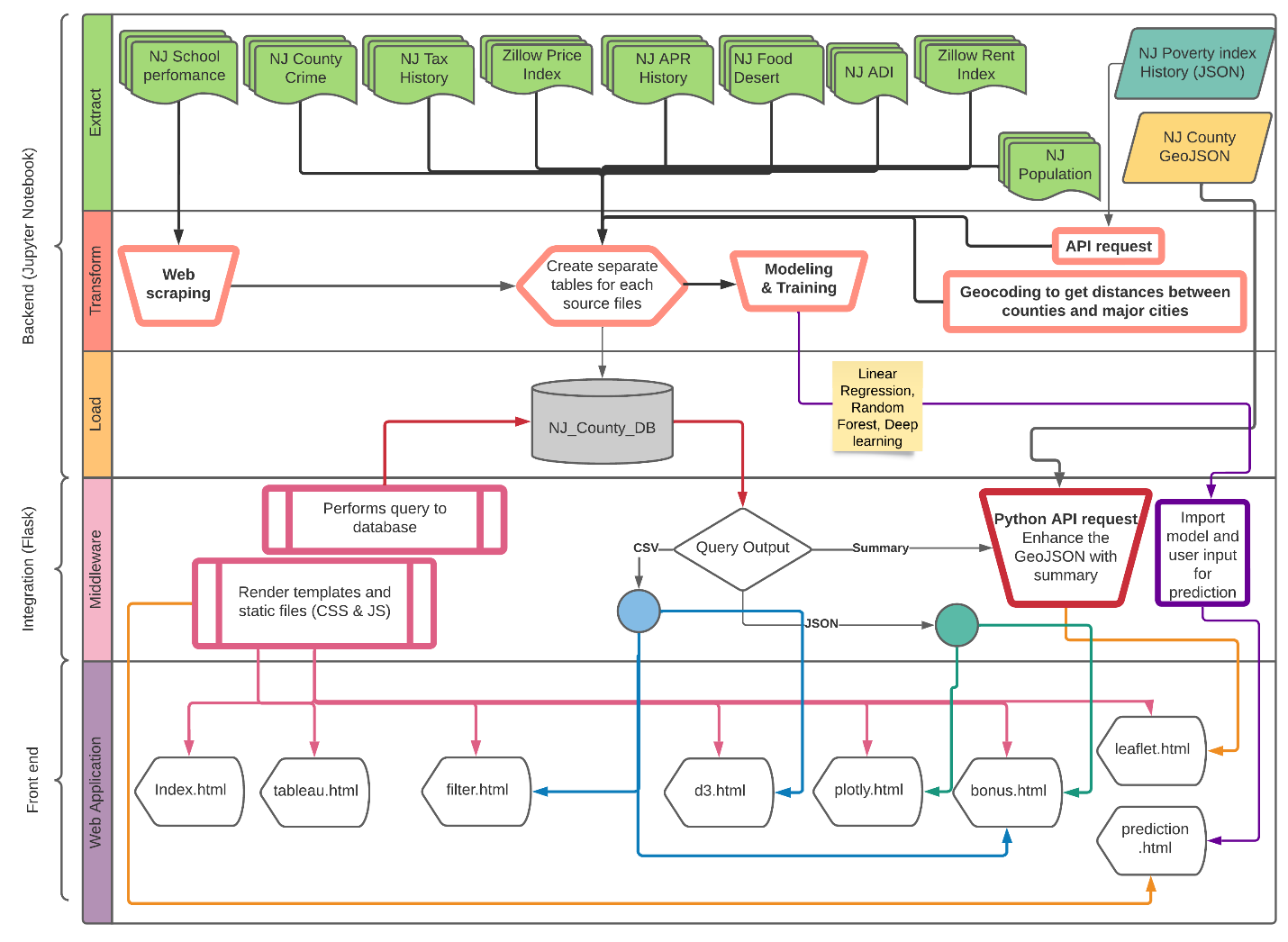
Below are the model performance scores:

Deep Neural Network - Loss: 0.04665800929069519, Accuracy: 0.9904761910438538

So, the model shows low loss and high accuracy. We did not perform any hyper parameter tuning for the deep learning model.

# Flask Application Framework

Below is the blueprint of flask application built with python 3.8.8:



Packages required for this application are as follows:

1. Flask == 2.0.3
2. Pandas == 1.4.3
3. Numpy == 1.21.5
4. Sqlalchemy == 1.4.39
5. Joblib == 1.1.0
6. Sklearn == 0.0
7. Tensorflow == 2.8.0
8. Requests == 2.28.1

All the notebooks and codes for this project will be found here:

<https://github.com/mirahmed07/DS_Masters_SPU/tree/master/DS_670>

# Conclusion

In this project, we aim to build an app that would expose users to two types of predictive models using various machine learning algorithms which would allow users to get house prices predictions and identify ideal locations for purchase or rent in NJ. The project's primary goal is to help new movers make informed decisions based on the predicted house price and ideal locations for purchase or rent in NJ.

**References**

“Moving to New Jersey - A Complete Guide 2023.” *Movingist*, 2023, Retrieved March 10, 2023. <https://movingist.com/moving-to-new-jersey/>

“Moving to New Jersey.” *Moving to New Jersey - A Complete Relocation Guide 2023*, Retrieved March 10, 2023. <https://www.moverjunction.com/moving-to-new-jersey>