**Machine Learning Model Mockup**

**Team Test Data**

**Design Study**

Identify question to be answered: which factors influence home values

Identify target variable: home values

Identify model: RandomForestRegressor, HistGradientBoostingRegressor (secondary)

**Dependencies**

[NumPy](https://www.askpython.com/python-modules/numpy/python-numpy-module)

[pandas](https://www.askpython.com/python-modules/pandas/python-pandas-module-tutorial)

[matplotlib](https://www.askpython.com/python-modules/matplotlib/python-matplotlib)

Sklearn

**Prepare Database**

Identify data sources

Component datasets

* Census.csv:

1990 Census data on communities

Selected features: counties (object)

Employees (numeric)

Establishments (numeric)

Observations: 60

* Housing.csv:

1999 Census data on housing in communities in California

Features (11):

* + City
  + Longitude
  + Latitude
  + Population
  + Median Age
  + Median Income
  + Median House Value
  + Total Rooms
  + Bedrooms
  + Households
  + Ocean Proximity

Observations: 20,641

* Weather data:

Weather for specific date called through weather API

Features:

* + Max Temp
  + Humidity
  + Cloudiness
  + Wind Speed
  + Description

Review data for normality and cross-influences

Identify outliers

Decide on protocol for missing or outlying values

Identify relevant/desirable features

Merge Weather, Census, Housing on common columns city, county

Drop observations with null values

Output database: Clean\_MergedDF.csv

Observations: 11,454

**Create Model**

***Load and preprocess***

Read dataset into pandas DataFrame

Input data: Clean\_MergedDF.csv

1. Pandas read csv Clean\_MergedDF.csv
2. Preprocess:
   1. Drop any null values
   2. Drop unnecessary columns
   3. Rename columns as necessary

***Split*** data into training and testing

1. Set X to independent variables
2. Set y to target variable identified in study design

***Tune model***

Random\_forest\_tuning = RandomForestRegressor(random\_state = )

param\_grid = {

'n\_estimators': [100, 200, 500],

'max\_features': ['auto', 'sqrt', 'log2'],

'max\_depth' : [4,5,6,7,8],

'criterion' :['squared\_error', 'absolute\_error']

}

GSCV = GridSearchCV(estimator=random\_forest\_tuning, param\_grid=param\_grid, cv=5)

GSCV.fit(X\_train, y\_train)

GSCV.best\_params\_

***Fit Random Forest regression to dataset***

First Pass:

* Number of trees: n\_estimators ({n\_estimators})
* Criterion
* Random\_state

from sklearn.ensemble import RandomForestRegressor

regressor = RandomForestRegressor(n\_estimators = 100, random\_state = 0)

regressor.fit(X,y)

***Test prediction***

y\_pred**=**regressor.predict()

y\_pred

***Create visualization graph***

***Interpret graph and test prediction***

***Compute, print, graph feature importances***

features = sorted(zip(X\_train.columns, regressor.feature\_importances\_), key = lambda x: x[1], reverse=True)

features[:11]

***Adjust model***

N\_estimators

Features

Secondary Model