Machine Learning and Real Estate

Helping buyers find home pricing deals using Machine Learning.

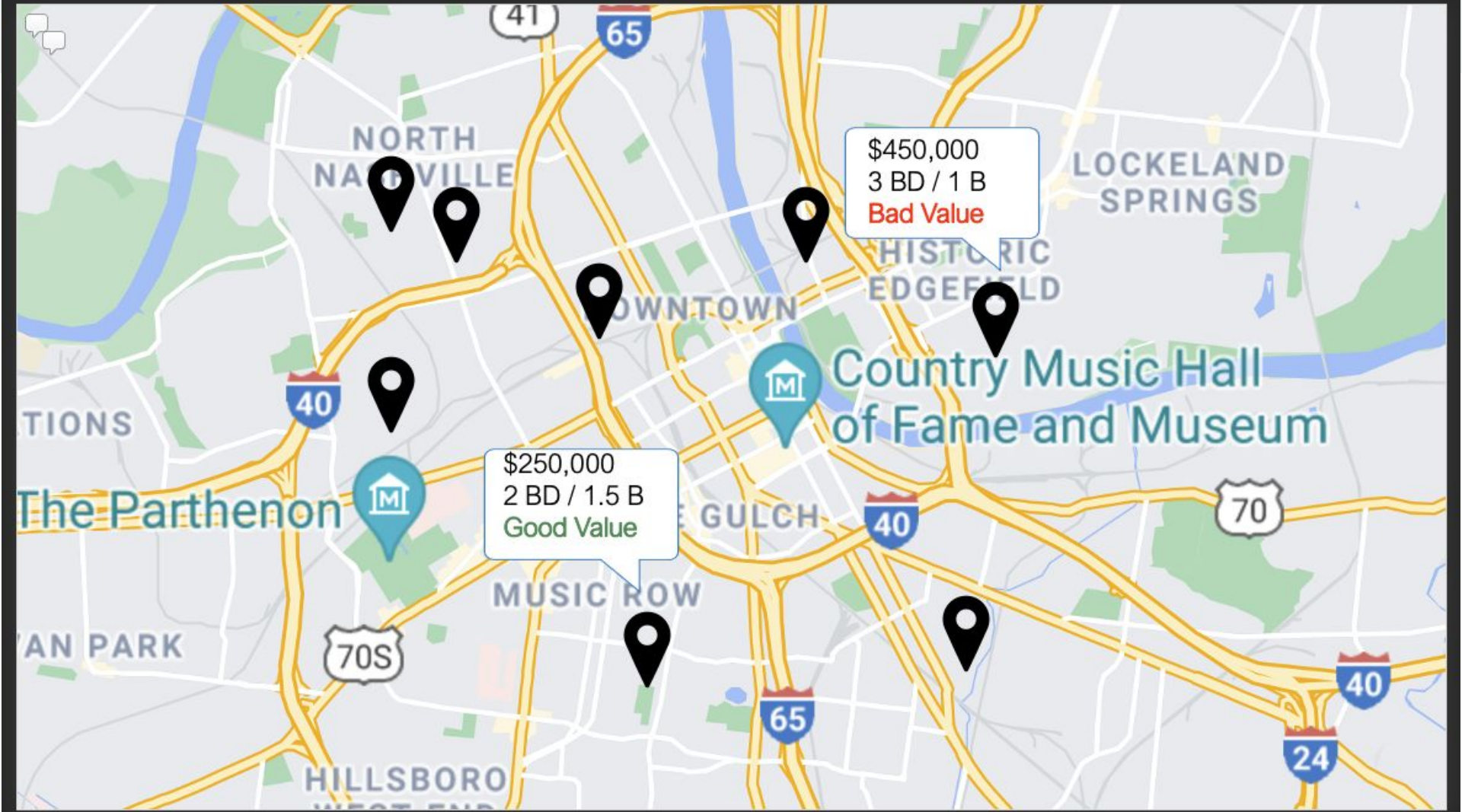
Nashville, ranks in the top 10 of fastest growing metros.

Nashville, TN and and the surrounding boroughs are predicted to be one of the top five housing markets in the country again this year. Historic low and near zero Federal Reserve rates have fueled far more buyers than sellers. With an imbalance of buyers to sellers, home prices can exceed the normalize value of the property. Additionally, many buyers exacerbate the problem by over bidding the value of the home feeding the continuation of over priced homes and non accommodating sellers.

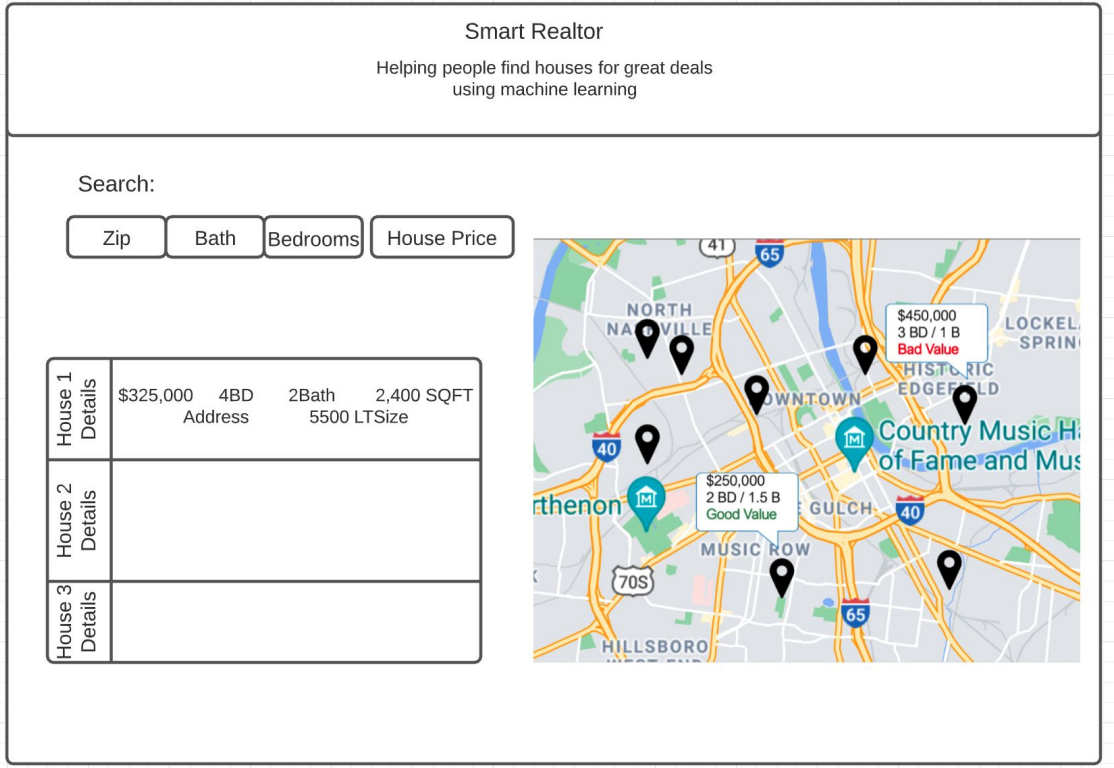
The Grey team will use Machine Learning to examine homes for sale and compare the current asking price along with other predefined features to previous sold homes to determine if the listing price is fair, above, or below market value.

The Problem

Many buyers are finding that the market is moving faster than they can process if the home asking price is a good deal (fair or below market value) or a bad deal (above market value). With Machine Learning, the Grey Team will process the current listing of homes for sale and develop a pop map of with location markers colored coded for fair, below, and above market value. Using the map provided by the Machine Learning algorithm prospective buyers can concentrate their efforts and resources on homes in the fair to below market value gaining instant equity in their purchase.


$450,000
3 BD / 1 B
Bad Value
$250,000
2 BD / 1.5 B
Good Value
NORTH
LOCKELAND
SPRINGS
DOWNTOWN
Country Music Hall
of Fame and Museum
The Parthenon
GULCH
MUSIC ROW
HILLSBORO
41
65
40
70S
70
24

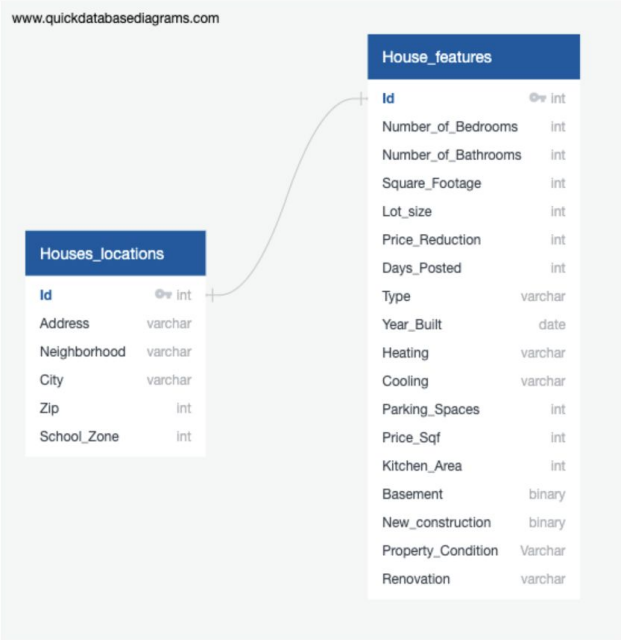
Need Screenshot of Dashboard Mock-Up



Sudo ML code

```
def baseline_model():
  # create model
  model = Sequential()
  model.add(Dense(13, input_dim= len(X_train[0]), kernel_initializer='normal', activation='relu'))
  model.add(Dense(1, kernel_initializer='normal'))
  # Compile model
  model.compile(loss='mean_squared_error', optimizer='adam')
  return model
#evaluate the model
estimators = []
estimators.append(('standardize', StandardScaler()))
estimators.append(('mlp', KerasRegressor(build_fn=baseline_model, epochs=1000, batch_size=5, verbose=0)))
pipeline = Pipeline(estimators)
kfold = KFold(n_splits=2)
results = cross_val_score(pipeline, X, y, cv=kfold)
print("Standardized: %.2f (%.2f) MSE" % (results.mean(), results.std()))
```

Sudo Database Connections and EDA



Example of Accuracy after the Epoch have computed

This standard error is extreme. We Understand, We will continue to troubleshoot this and might have to consider removing features or Even changing models (potential multiple linear regression possible or xg boost regressor). Data is scaled.

```
Standardized: -183162003456.00 (42217578496.00) MSE
```

Smart Realtor

Helping people find houses for great deals
using machine learning

Search:

Zip	Bath	Bedrooms	House Price

House 1 Details	$325,000 4BD 2Bath 2,400 SQFT Address 5500 LTSize
House 2 Details	
House 3 Details