Project 4

Alice Pao

Project Summary

This project is meant to utilize collected data to train and test machine to recognize if a house is built before or after 1980. Firstly, I created charts to find potential relationships through different variables and determined features to used for training model.

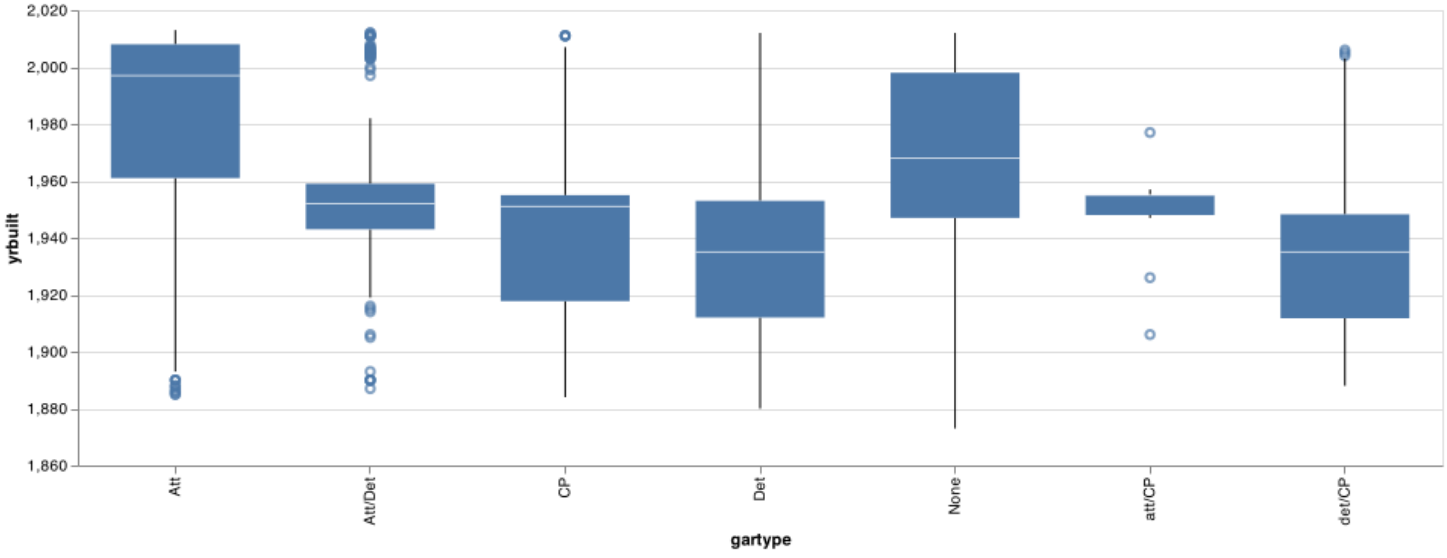
Then, I built a learning model with RandomForest classification and also confirm if the choice of classification is appropriate by listing important features selected by the computer. Lastly, the quality of the model is assessed by using different evaluation metrics including accuracy, recall, and precision rate.

Technical Details

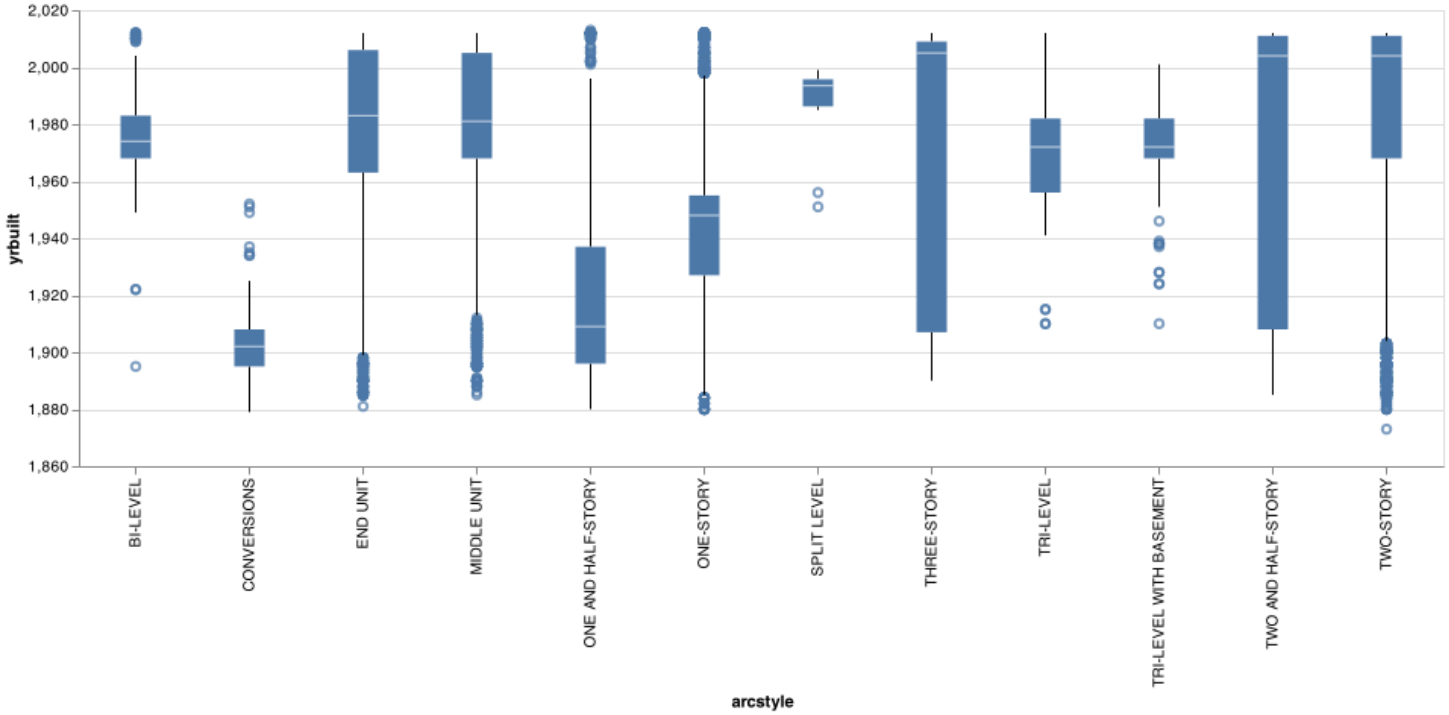
Grand Question 1

Create 2-3 charts that evaluate potential relationships between the house variables and the variable before1980. Explain what you learn from the charts that could help a machine learning algorithm.

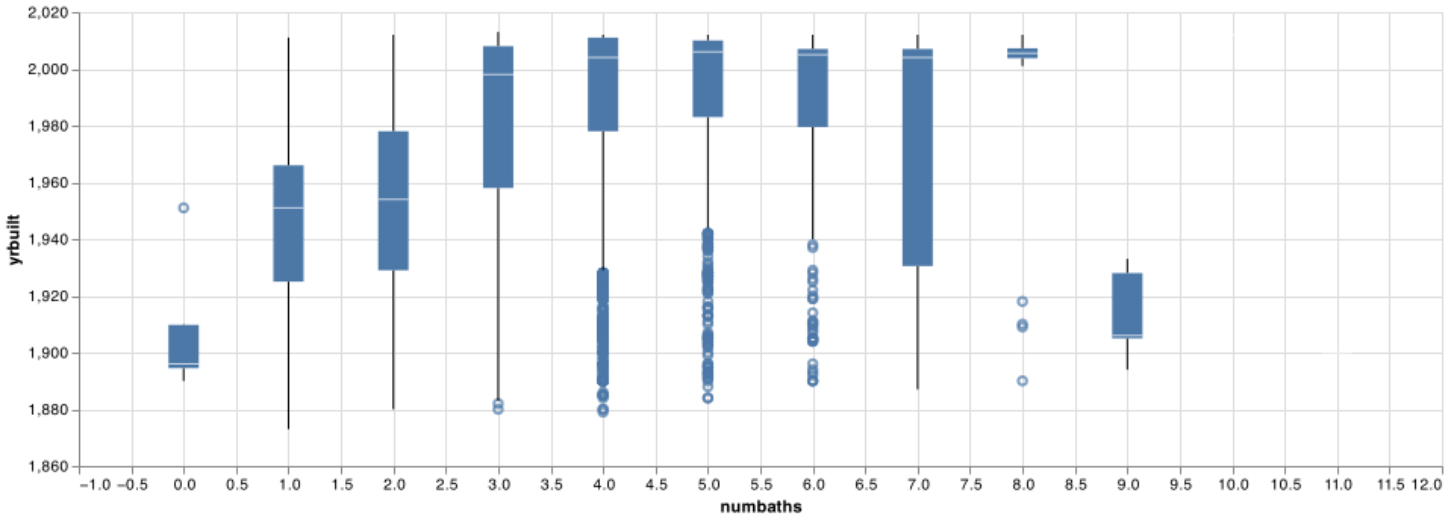
In Chart 1, it shows different garage types. It indicates that Att/Det, CP, Det, att/CP, and det/cp type are all houses built before 1980. Therefore, it is an important feature for training model.



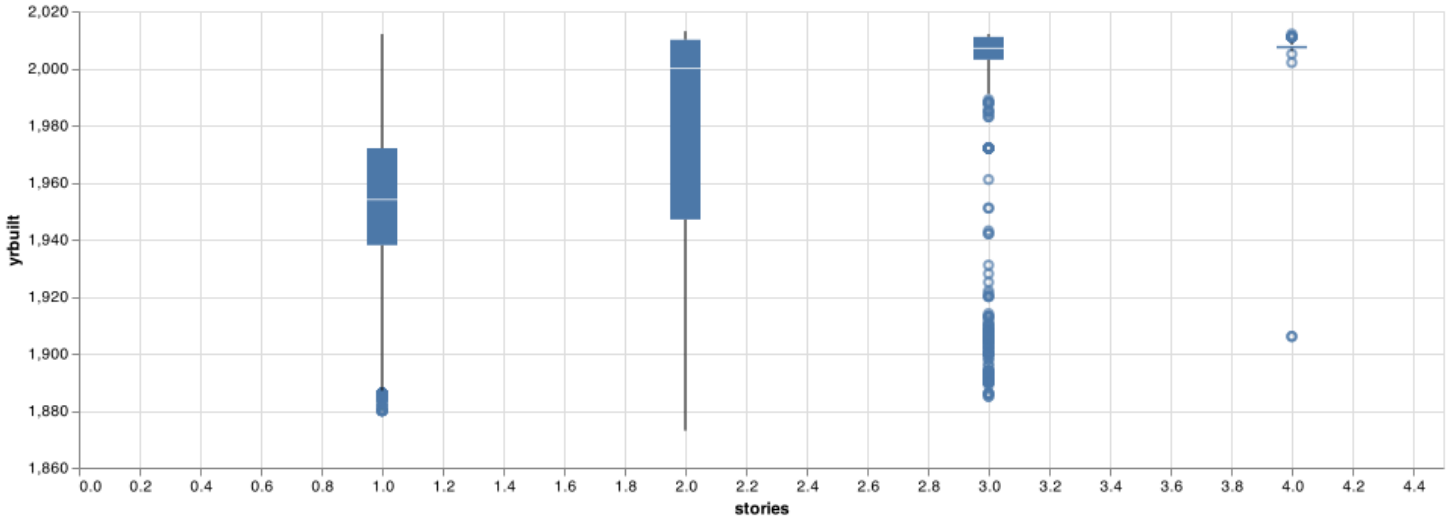
In Chart 2, most BI-level houses are built before 1980. Moreover, One and half story, one story, tri-level, and tri-level with basement style are mostly built before 1980 as well.



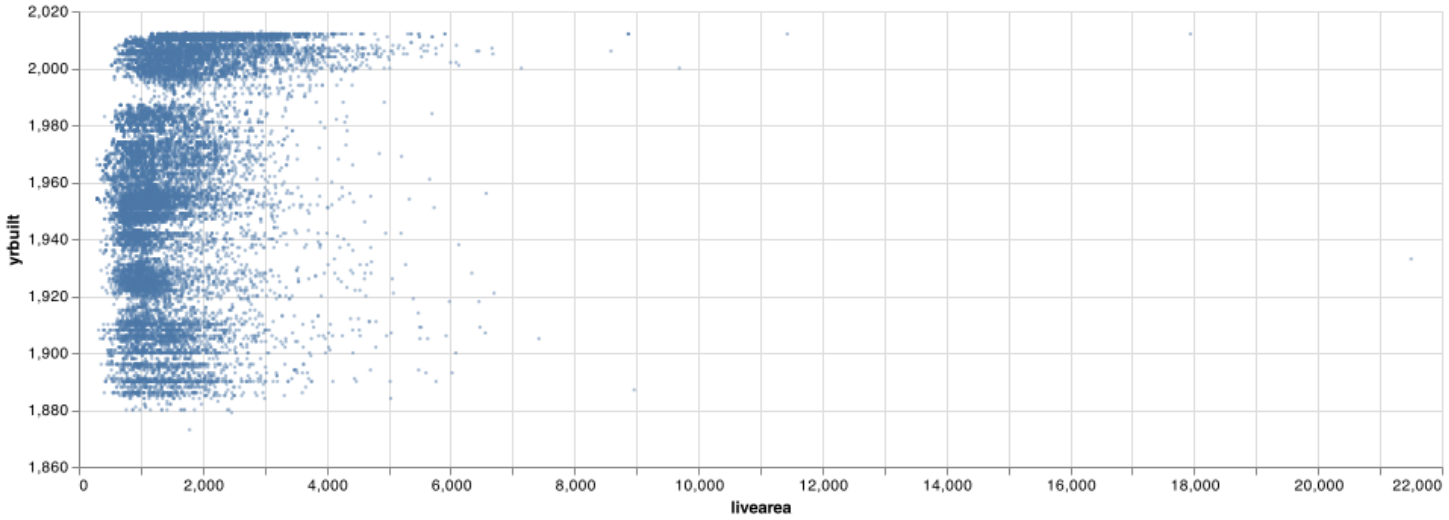
In Chart 3, it summarizes the number of bathrooms from all houses in the dataset. It looks like new houses after 1980 have more number of bathrooms.(mostly 5 and 6 bathrooms)



In Chart 4, one obvious feature is that all one-story houses are built before 1980.



In Chart 5, it records total squares of living area of houses. It is harder to distinguish; but from this chart, we can possilbly determine that a house is built more recently if its living area is larger. For example, one house with almost 18,000 square is built after 2000.



Grand Question 2

Build a classification model labeling houses as being built “before 1980” or “during or after 1980”. Your goal is to reach 90% accuracy. Explain your final model choice (algorithm, tuning parameters, etc) and describe what other models you tried.

I have tried GaussianNB and found out accuracy dropped drastically from 86% to 43%. Therefore, I decided not to use it. I also noticed that DecisionTree and RandomForest classification model have similar accuracy rate.

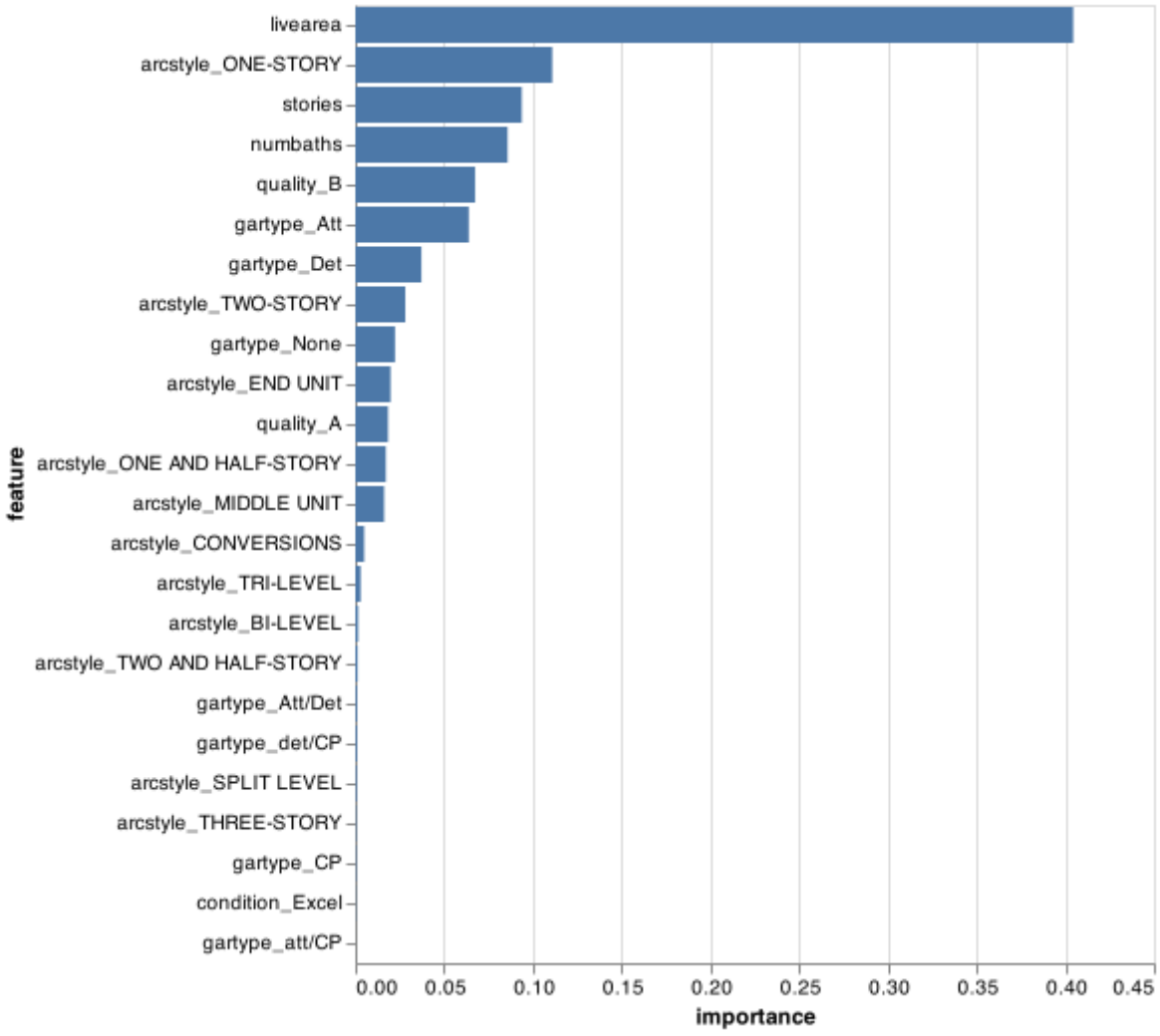
After playing around with these 3 models, I chose RandomForest classification to train, test, and make predictions. One issue I ran into is my accuracy was stagnant in 89% no matter what features I dropped or added unto. Eventaully, I changed the test size from .4 to .3 and it raised the rate to 90%.

Grand Question 3

Justify your classification model by discussing the most important features selected by your model. This discussion should include a chart and a description of the features.

My model thinks living area, one-story house, total story of houses,and number of bathrooms are the most important factors and I think it is right because judging from charts in Question 1, stories of houses show important relationships, including one-story houses are all built before 1980. Number of bathrooms is another critical feature.

As for livearea, even though scatterplot cannot clearly distinguish the relationship, it contains detailed and complete data; therefore, it plays an important role for machine learning too.



Grand Question 4

Describe the quality of your classification model using 2-3 different evaluation metrics. You also need to explain how to interpret each of the evaluation metrics you use.

I believe my classification model has good quality because the accuracy reaches 90%, meaning out of all the total testing samples, 6874 cases, the model got 6207 of them correct.

Additionally, the recall rate is 92%. This indicates that from all houses built before 1980, 3961 of them were predicted correctly.

Lastly, final precision rate hits 93%. It shows that out of all the prediction (4268 cases) for houses built before 1980, only 7% of them were wrongly labeled as houses built before 1980.

Appendix A

```
#%%
# import pacakges
import pandas as pd
import numpy as np
import altair as alt
from sklearn.model_selection import train_test_split
from sklearn import metrics
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier
alt.data_transformers.enable('data_server')

#%%
# import data
df_denver = pd.read_csv('https://github.com/byuidatascience/data4dwellings/raw/master/data-raw/dwelling
df_ml = pd.read_csv('https://github.com/byuidatascience/data4dwellings/raw/master/data-raw/dwellings_ml
df_neighbor_ml = pd.read_csv('https://github.com/byuidatascience/data4dwellings/raw/master/data-raw/dwe

#%%
# Grand Question 1:
df1 = pd.read_csv('https://github.com/byuidatascience/data4dwellings/raw/master/data-raw/dwellings_denv
# df1.gartype.value_counts()

# Variable 1: garage type: gartype, Att: after 1980, att/cp: before 1980
q1chart_1 = (alt.Chart(df1)
    .mark_boxplot(
        size = 85
    )
    .encode(
        x = 'gartype',
        y = alt.Y('yrbuilt', scale = alt.Scale(zero=False))
    )
    .properties(
        width = 900
    )
)
q1chart_1.save('q1chart_1.png')

# Variable 2: type of home: arcstyle: conversions: before 1980, after 1980: end unit, middle unit, thre
q1chart_2 = (alt.Chart(df1)
    .mark_boxplot(
        size = 20
    )
    .encode(
        x = 'arcstyle',
        y = alt.Y('yrbuilt', scale = alt.Scale(zero=False))
    )
    .properties(
        width = 900
    )
)
```

```
q1chart_2.save('q1chart_2.png')

# Variable 3: numbers of bathroom: numbaths, after 1980: more than 3 baths
q1chart_3 = (alt.Chart(df1)
    .mark_boxplot(
        size = 20
    )
    .encode(
        x = 'numbaths',
        y = alt.Y('yrbuilt', scale = alt.Scale(zero=False))
    )
    .properties(
        width = 900
    )
)
q1chart_3.save('q1chart_3.png')

# Variable 4: number of story: stories, after 1980: 2-3 floors
q1chart_4 = (alt.Chart(df1)
    .mark_boxplot(
        size = 20
    )
    .encode(
        x = 'stories',
        y = alt.Y('yrbuilt', scale = alt.Scale(zero=False))
    )
    .properties(
        width = 900
    )
)
q1chart_4.save('q1chart_4.png')

# Variable 5: square footage that is livable: livearea
q1chart_5 = (alt.Chart(df1)
    .mark_circle(
        size = 5,
        opacity=0.5
    )
    .encode(
        x = 'livearea',
        y = alt.Y('yrbuilt', scale = alt.Scale(zero=False))
    )
    .properties(
        width = 900
    )
)
q1chart_5.save('q1chart_5.png')

# %%
# Grand Question 2:
# features
```

```
x = df_ml.filter(['livearea','numbaths', 'stories','condition_Excel', 'quality_A', 'quality_B', 'gartyp
       'gartype_Det', 'gartype_None', 'gartype_att/CP', 'gartype_det/CP','arcstyle_BI-LEVEL', 'arcstyle
       'arcstyle_MIDDLE UNIT', 'arcstyle_ONE AND HALF-STORY',
       'arcstyle_ONE-STORY', 'arcstyle_SPLIT LEVEL', 'arcstyle_THREE-STORY',
       'arcstyle_TRI-LEVEL',
       'arcstyle_TWO AND HALF-STORY', 'arcstyle_TWO-STORY'])

# target
y = df_ml.filter(['before1980'])

# split training & test size: 70% vs. 30%
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=.3, random_state =76)

# create the model
classifier = RandomForestClassifier()

# train the model
classifier.fit(x_train, y_train)

# make predictions
y_predictions = classifier.predict(x_test)

# test classifier, using predict method
y_predictions = classifier.predict(x_test)
# assess classifier performance: accuracy
accuracy = metrics.accuracy_score(y_test, y_predictions)

# %%
# Question 3:
# ideas to bump up accuracy: change model, try differnet variables
classifier.feature_importances_
feature_df = pd.DataFrame({'feature':x.columns,'importance':classifier.feature_importances_})
# >> use this df and create a bar chart
chart3 = (alt.Chart(feature_df)
    .mark_bar()
    .encode(
        y = alt.Y('feature', sort = '-x'),
        x = 'importance'
    ))
chart3.save('question_3.png')

#%%
# Question 4:
# use the confusiion_matrix to get the true vs. predicted results table
metrics.plot_confusion_matrix(classifier, x_test, y_test)

# print out the classification report including precision, recall, and accuracy
print(metrics.classification_report(y_test, y_predictions))
```