Project 4: Classifying Homes

Data Science Programming CSE 250

Travis Smith

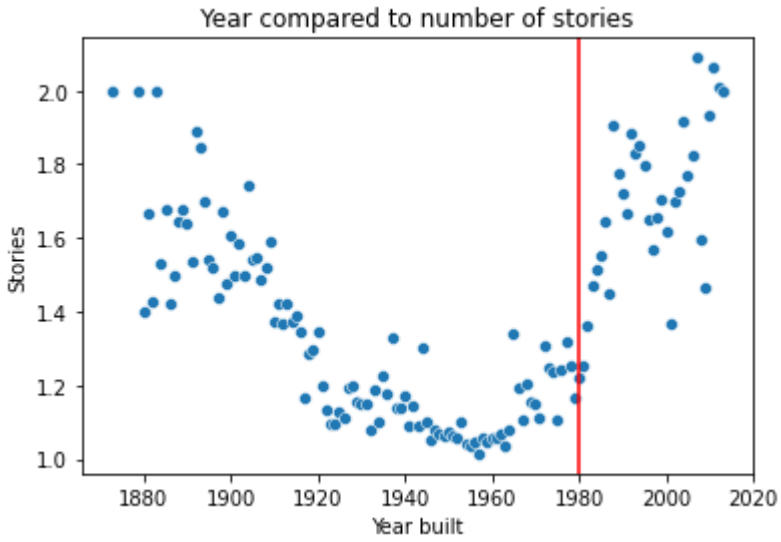
Project Summary

This project is to explore machine learning and to be able to create a model to make predictions. In this project we are using homes in Denver that have many features such as how many levels the home has or the living area. The goal is to accurately predict if the home was built before 1980 or not.

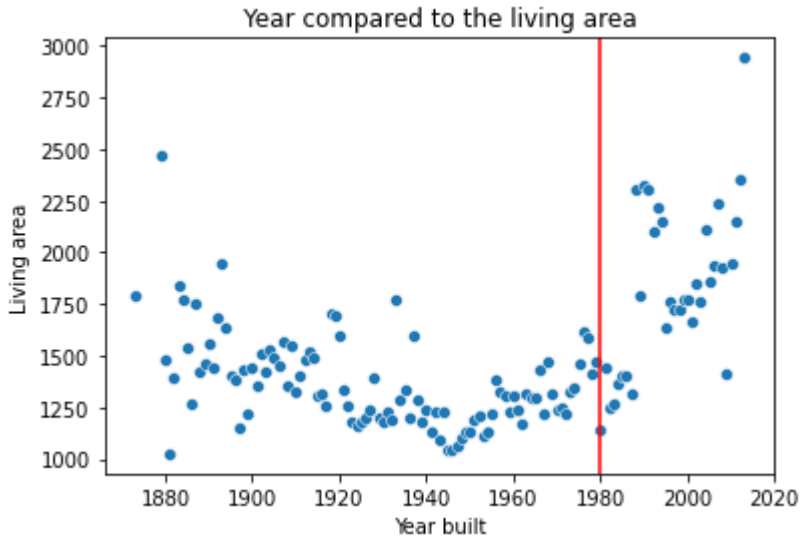
Technical Details

Grand Question 1:

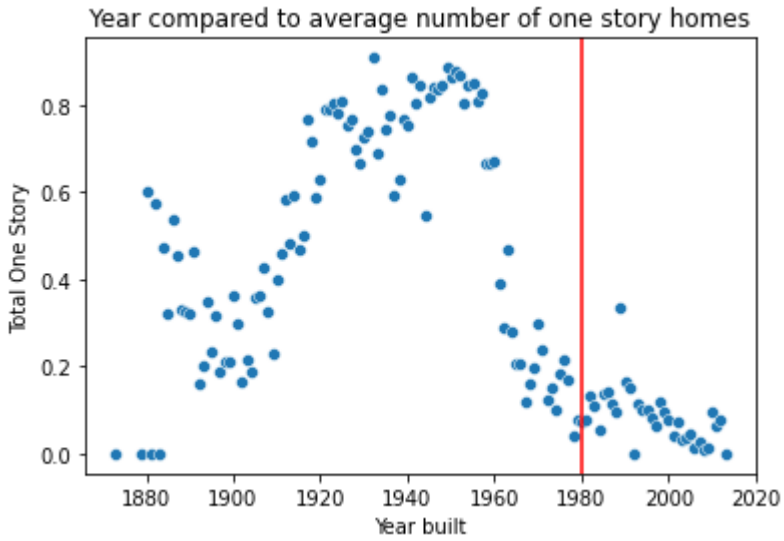
Create 2-3 charts that evaluate potential relationships between the house variables and the variable before1980 Explain what you learn from the charts that could help a machine learning algorithm.



This chart shows that the number of stories was lowest around 1960, but then started to increase but after 1980 it increaced more then before.



This chart shows that after 1980 the living area started to increase. This makes it easier to guess what year the home was built looking at the increase after 1980.



The amount of one level homes greatly went down around 1960 then it leveled out around 1980. After 1980 the amount of one level homes leveled out.

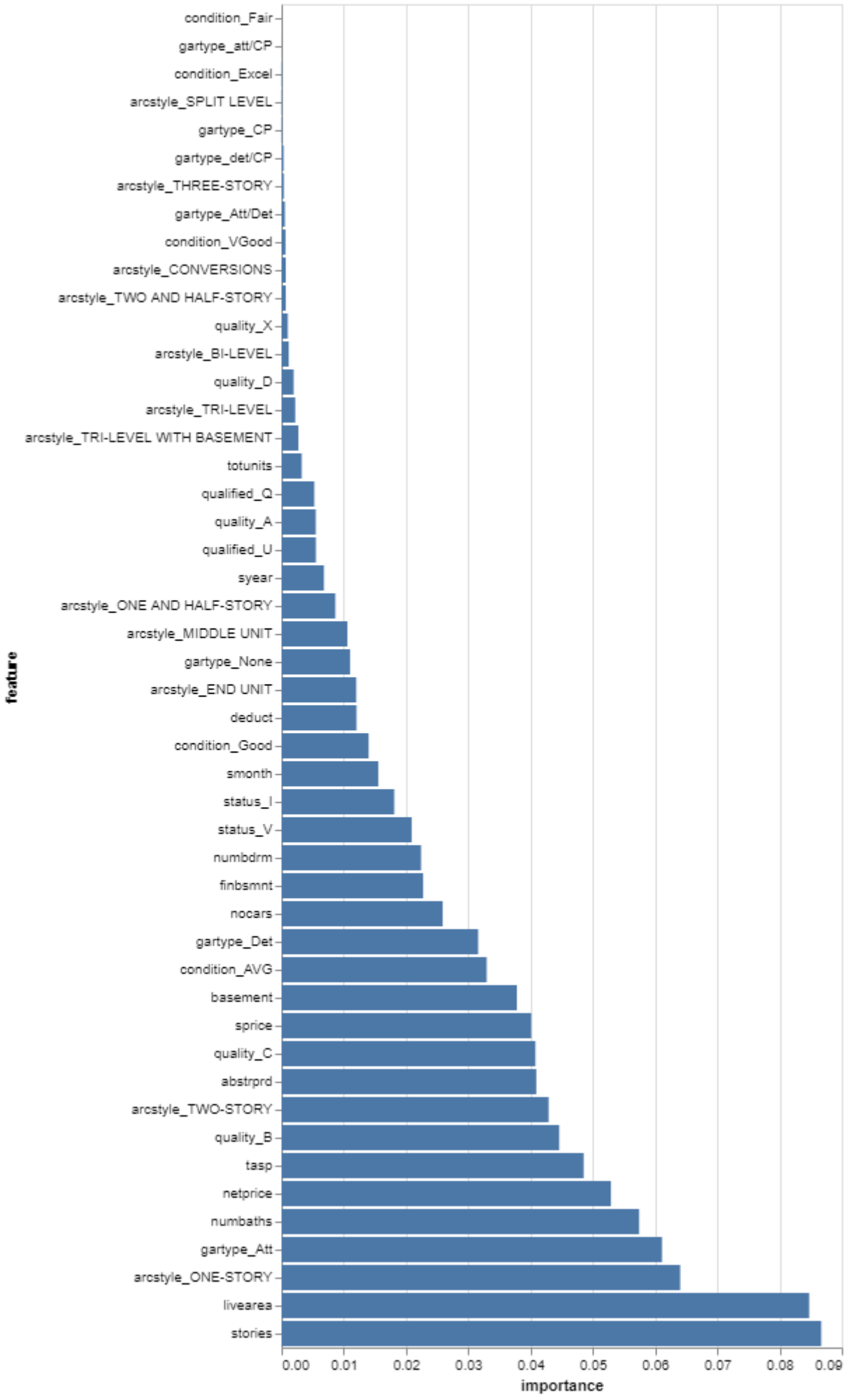
Grand Question 2:

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 got an accuracy of 92.34% What I did for my learning model was I used a random forest classifier, I used 20% for testing leaving 80% for training, I only dropped the parcel ID ('parcel'), the year it was built ('yrbuilt'), and obviously the data that I am testing for ('before1980'). I found that using all the provided features were helpful and it seemed that a lot of it did influence in the decision.

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.


condition_Fair
gartype_att/CP
condition_Excel
arcstyle_SPLIT LEVEL
gartype_CP
gartype_det/CP
arcstyle_THREE-STORY
gartype_Att/Det
condition_VGood
arcstyle_CONVERSIONS
arcstyle_TWO AND HALF-STORY
quality_X
arcstyle_BI-LEVEL
quality_D
arcstyle_TRI-LEVEL
arcstyle_TRI-LEVEL WITH BASEMENT
totunits
qualified_Q
quality_A
qualified_U
syear
arcstyle_ONE AND HALF-STORY
arcstyle_MIDDLE UNIT
gartype_None
arcstyle_END UNIT
deduct
condition_Good
smonth
status_I
status_V
numbdrm
finbsmnt
nocars
gartype_Det
condition_AVG
basement
sprice
quality_C
abstrprd
arcstyle_TWO-STORY
quality_B
tasp
netprice
numbaths
gartype_Att
arcstyle_ONE-STORY
livearea
stories
feature
0.00
0.01
0.02
0.03
0.04
0.05
0.06
0.07
0.08
0.09
importance

In this chart we can see that the top 5 most important features are (in order of importance): stories, livearea, arcstyle_ONE-STORY, gartype_Att, and numbaths. These help the learning model determine the year of the home more than all the other features.

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.

	precision	recall
0	0.90	0.89
1	0.93	0.94

Accuracy = 92.34%

In this project we are trying to find homes with asbestos which is hazardous. So it is important to get it right. I may be accurate but I could be getting the true positives wrong and leaving homes with asbestos. So it is important to have a model that has a similar accuarcy, precision and recall.

Appendix A

```
import pandas as pd
import altair as alt
import seaborn as sns
import matplotlib.pyplot as plt

# import dataframes
homes = pd.read_csv('dwellings_ml.csv')
denver = pd.read_csv('dwellings_denver.csv')
# neighborhoods = pd.read_csv('dwellings_neighborhoods_ml.csv')

# Make charts possible and show data
alt.data_transformers.enable('data_server')
homes.head(3)

# Grand question 1
# Group all homes by the year and average all the columns
q1 = homes.groupby(['yrbuilt']).mean()

# Make chart to show corilation with yrbuilt and stories
sns.scatterplot(data=q1, x='yrbuilt', y='stories')
plt.axvline(1980,0, color='red')

# Make chart to show corilation with yrbuilt and livearea
sns.scatterplot(data=q1, x='yrbuilt', y='livearea')
plt.axvline(1980,0, color='red')

# Make chart to show corilation with yrbuilt and one story homes
```

```
sns.scatterplot(data=q1, x='yrbuilt', y='arcstyle_ONE-STORY')
plt.axvline(1980,0, color='red')

# Grand question 2
x = homes.drop(columns=['before1980','parcel', 'yrbuilt'])
y = homes.filter(['before1980'])

from sklearn.model_selection import train_test_split
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.2,
random_state=42)

from sklearn import metrics
from sklearn.ensemble import RandomForestClassifier

# create the model
classifier = RandomForestClassifier(n_estimators=25, max_depth=15,
random_state=10)

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

# make predictions
y_predictions = classifier.predict(x_test)

# test how accurate predictions are
print('Accuracy score:', (metrics.accuracy_score(y_test,
y_predictions)*100).round(2), '%')

# Grand question 3
# Importance features ranked by bar chart

feature_df = pd.DataFrame({'feature':x.columns,
'importance':classifier.feature_importances_})
# feature_df.sort_values(by='importance',ascending=False).head(8)

import altair as alt

(alt.Chart(feature_df)
    .mark_bar()
    .encode(
        x = 'importance',
        y = alt.Y('feature', sort='x')
    ))

# Grand question 4
# Accuracy, recall, precision

print(metrics.classification_report(y_test, y_predictions))
```