Client Report - Project 4: Can you predict that?

See code

Course DS 250

AUTHOR
Bracken Sant

Elevator pitch

By utilizing sklearn, pandas, and altair, I saw if I could predict, wiht a machine learning algorithm, whether a house was built before 1980 and could, therefore, have asbestos within the walls. I checked with different features of the house and saw what effect they had on the accuracy of the prediction and created a model that is effective in predicting whether a house was built before 1980.

▼ Read and format project data

```
denver_data = pd.read_csv("dwellings_denver.csv")
ml_data = pd.read_csv("dwellings_ml.csv")
```

GRAND QUESTION 1

Create 2-3 charts that evaluate potential relationships between the home variables and before1980.

I checked 3 different features of the houses: basement, netprice, and livearea. There doesn't seem to be to big of a correlation alone but when combined they can show better results, which is what the machine learning algorithm is for.

▼ Read and format data into charts to show the distribution of different features within the timeline.

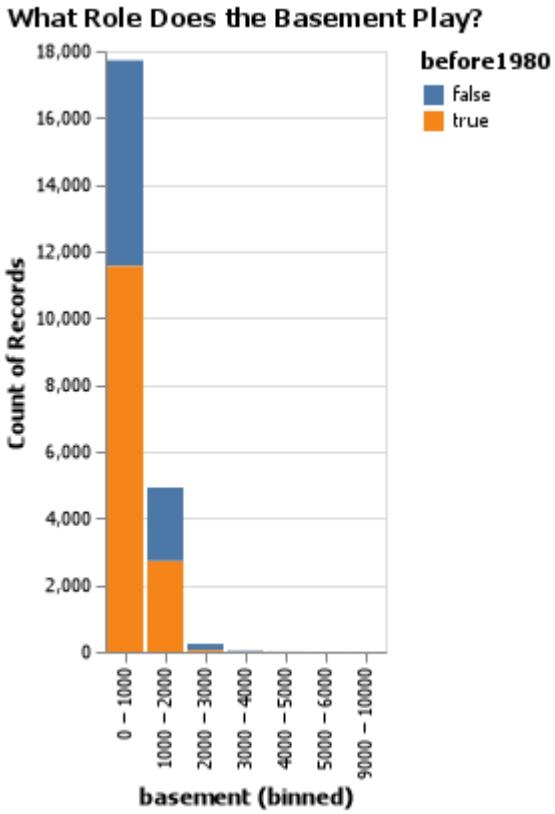
```
al.data_transformers.disable_max_rows()
ml_data["before1980"] = ml_data["yrbuilt"] < 1980
ml_data["finishedBasement"] = (ml_data['basement'] - ml_data['finbsmnt']) == 0
denver_data["before1980"] = denver_data["yrbuilt"] < 1980

chart1 = (al.Chart(ml_data).mark_bar()
        .encode(
            x=al.X("basement:N", bin=al.Bin(nice=True)),
            y=al.Y("count()"),
            color=al.Color("before1980:N"))
        .properties(title="What Role Does the Basement Play?"))

chart2 = (al.Chart(ml_data).mark_bar()
         .encode(
```

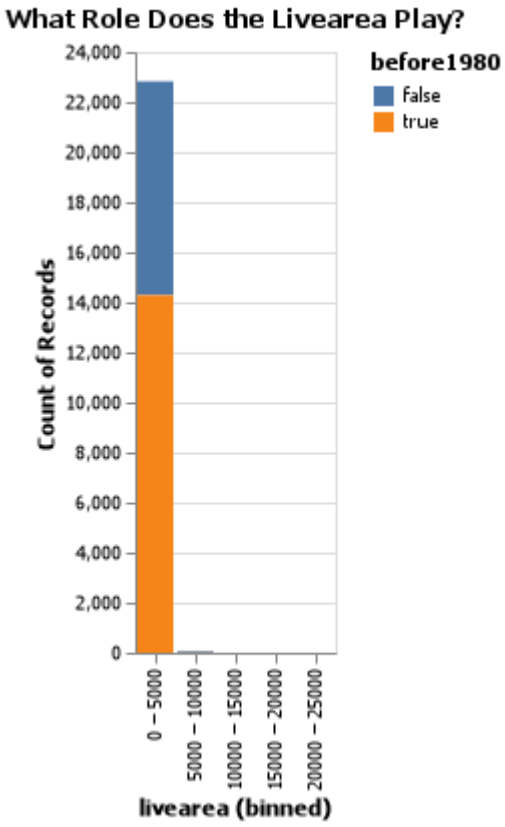
```
        .encode(
            x=al.X("netprice:N", bin=al.Bin(nice=True)),
            y=al.Y("count()"),
            color=al.Color("before1980:N"))
        .properties(title="What Role Does the Netprice Play?"))

chart3 = (al.Chart(ml_data).mark_bar()
        .encode(
            x=al.X("livearea:N", bin=al.Bin(nice=True)),
            y=al.Y("count()"),
            color=al.Color("before1980:N"))
        .properties(title="What Role Does the Livearea Play?"))
```



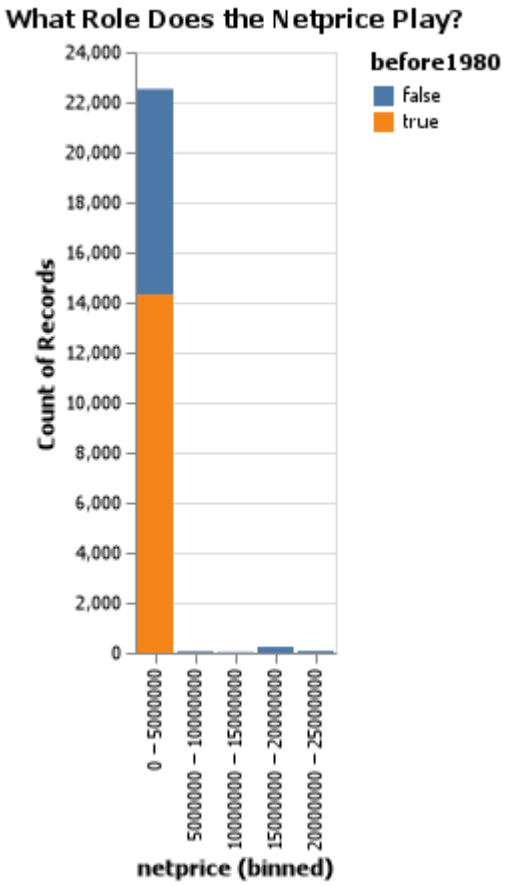
Basement Chart

The basement chart shows that the distribution of before 1980 is primarily within the 0 to 2000.



LiveArea Chart

The livearea chart shows that the distribution of before 1980 is primarily within the 0 to 5000, however, that is also true with the after 1980.



NetPrice Chart

The netprice chart shows that the distribution of before 1980 is primarily within the 0 to $500,000, however, that is also true with most of the after 1980 as well. It appears that there are not any houses with a netprice of over $500,000 before 1980.

GRAND QUESTION 2

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

I didn't mess with the parameters in RandomForestClassifier, the only thing that I changed was what home variables to use. I began with all of the variables and it came out to 92.86% I then took the top 15 variables from feature importances and used them, then I trimmed it down by getting out variables that didn't seem to be associated with the needed evalutation of before1980 and got these 10 variabes that produced a 92.17%, less than one percent lower than using all of the variables.

▼ Read and format data

```
#0.92857
#X = ml_data.drop("before1980", axis=1).drop("parcel", axis=1).drop("yrbuilt", axis=1)
#0.92173
```

```
X = ml_data[["livearea", "basement", "arcstyle_ONE-STORY", "stories", "gartype_Att",
         "numbaths", "netprice", "quality_C", "nocars", "condition_AVG"]]
y = ml_data['before1980']
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=.3, random_state=24,
         stratify=y)

rf = RandomForestClassifier(random_state=24)
rf.fit(X_train, y_train)

y_pred = rf.predict(X_test)
print("Accuracy: %.3f" % accuracy_score(y_test, y_pred))
```

```
Accuracy: 0.922
```

As you can see from the accuracy score, it is above 90% so the model is affective with this data.

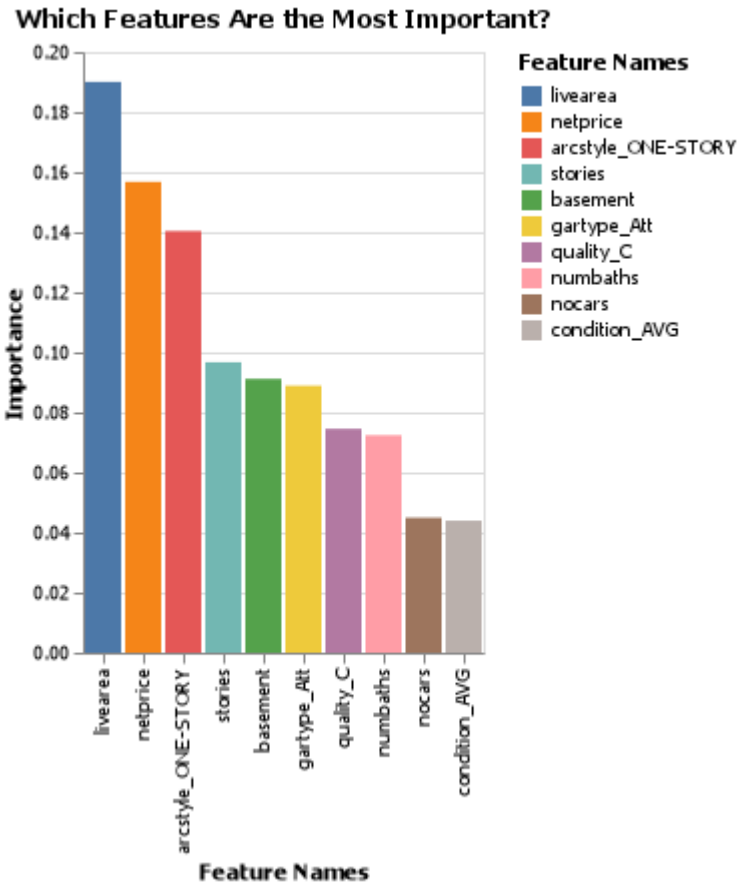
GRAND QUESTION 3

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.

According to the feature_importances the most important features in the model are livearea, netprice, and arcstyle_ONE_STORY._

▼ Read and format data

```
feat_imports = (pd.DataFrame({'feature names': X_train.columns, 'importances' :
         rf.feature_importances_}).sort_values('importances', ascending=False))
chart4 = al.Chart(feat_imports).mark_bar().encode(
        al.X("feature names:N", sort=feat_imports["feature names"].to_list(), title="Feature
         Names"),
        al.Y("importances:Q", title="Importance"),
        al.Color("feature names:N", sort=feat_imports["feature names"].to_list(),
         title="Feature Names")
    ).properties(title="Which Features Are the Most Important?")
```



The Features

I think that it makes sense that the netprice and arcstyle_ONE_STORY correlate well with the age of the house because it wasn't as common to have a house with more than one story before 1980 than it is now, and the netprice should get lower as time goes on because of the upkeep that is needed with older houses.

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 decided to use precision and recall. Precision is calculated by number of true positives over number of true positves + number of false positives, so it is more mindful of false positives. Recall is calculated by number of true positives over number of true positives + number of false negatives, so it is more mindful of false negatives.

▼ Read and format data

```
y_pred = rf.predict(X_test)

print("Precision: %.3f" % precision_score(y_test, y_pred))

print("Recall: %.3f" % recall_score(y_test, y_pred))
```

```
Precision: 0.940
Recall: 0.935
```

What our study would more likely lean towards is the recall grade because it is better to have a false positive than to give false negatives when you are testing something that can endanger peoples' lives. The quality of the model is not only shown through the accuracy with 92.2% but also with the precision at 94.0% and the recall at 93.5%.