Project 4: Can you predict that?

Course CSE 250
Christian Lira Gonzalez

Elevator pitch

Throughout this project we will learn about Machine Learning, We will take a close sight to ceratin data and analyze it based on the information that we can extract from it and the applying the principles learned on the topic, generate charts that can help us visualize and generate a knowledge on such data.
We analyze data on Houses in the Colorado state comparing the before and after "1980".

GRAND QUESTION 1

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

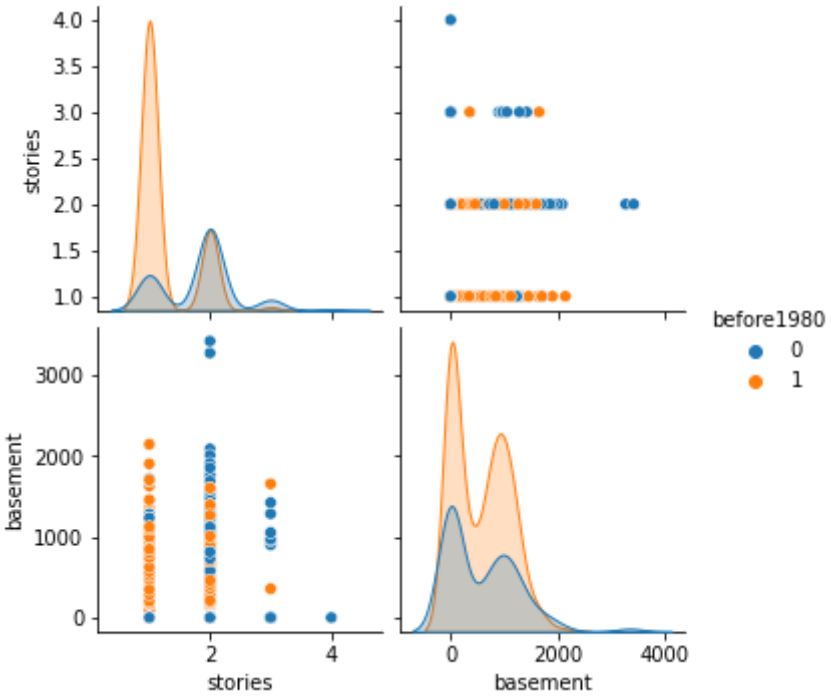
The following three charts are related to the data set that we have and are relatable comparissons on such information. Below the Charts and their own descriptions

TECHNICAL DETAILS

```
#%%
import pandas as pd
import numpy as np
import seaborn as sns
import altair as alt
#%%
from sklearn.model_selection import train_test_split
from sklearn import tree
from sklearn.naive_bayes import GaussianNB
from sklearn.ensemble import GradientBoostingClassifier
from sklearn import metrics
# %%
dwellings_denver = pd.read_csv("https://github.com/byuidatascience/data4dwellings/raw/master/dat
dwellings_ml = pd.read_csv("https://github.com/byuidatascience/data4dwellings/raw/master/data-ra
dwellings_neighborhoods_ml = pd.read_csv("https://github.com/byuidatascience/data4dwellings/raw/
#%%
#Query/Chart 1
que_1 = dwellings_ml.filter(["livearea", "finbsmnt", "before1980"]).sample(500)
sns.pairplot(que_1, hue = "before1980")
#%%
#Query/Chart 2
que_2 = dwellings_ml.filter(["stories", "basement", "before1980"]).sample(500)
sns.pairplot(que_2, hue = "before1980")
#%%
#Query/Chart 3
que_3 = dwellings_ml.filter(["numbdrm", "livearea", "before1980"]).sample(500)
sns.pairplot(que_3, hue = "before1980")
```

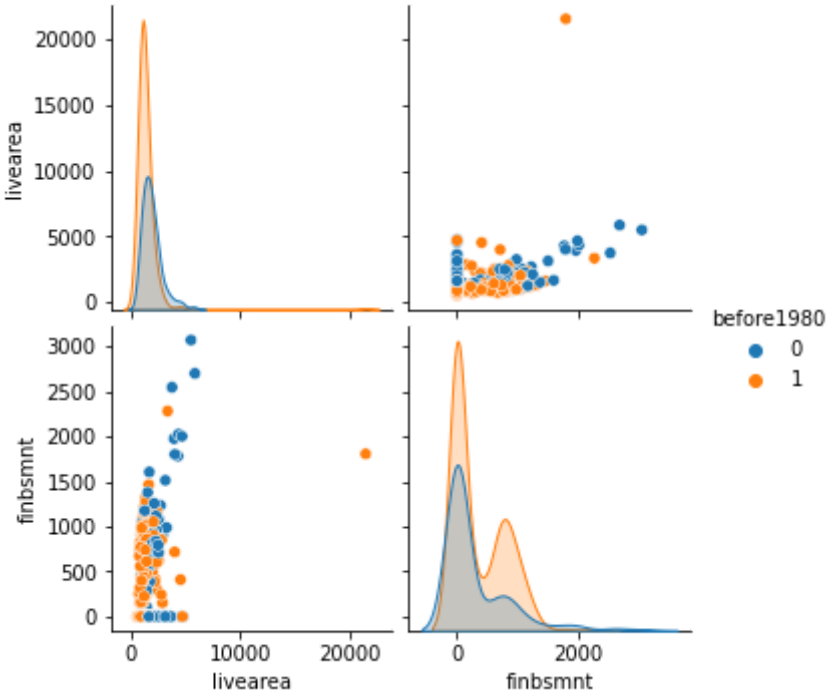
First Chart Comparisson of Basement / Stories:

In this first chart we compare the houses before and after 1980 that contain a basement and also are more than just one store. In the charts we can see that the relationship can help us understand how the living area was distributed in the houses. And we can learn that older houses tended to have less stores but at least to have a basement, contrary to newer houses that contain in their mayority more than just one store but some time they do not include a basement.



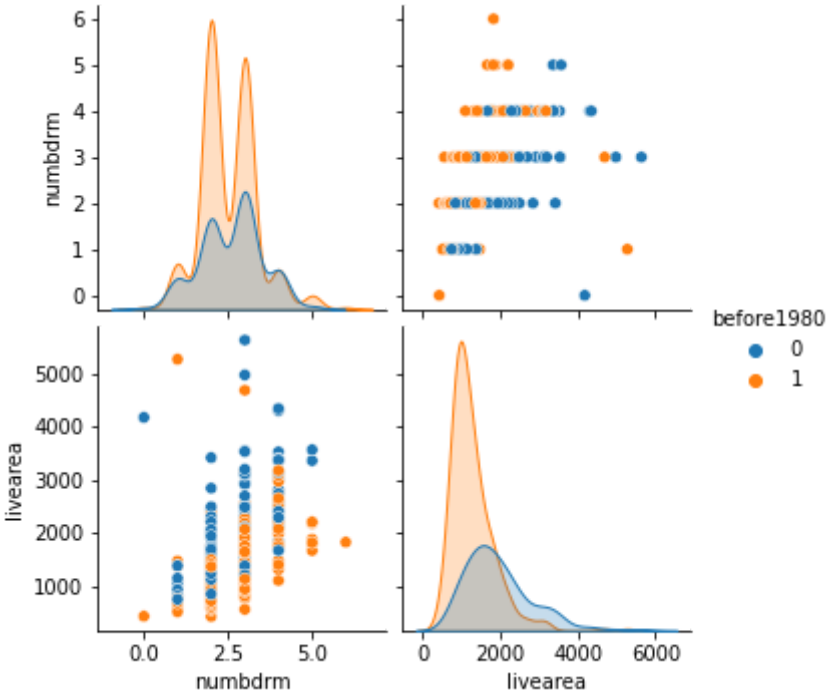
Second Chart Comparisson of Live Area / Basement

What we can observe in this Secon Chart is really interesting. First we learn that newer houses have less living area than houses built befeore 1980. Secondly as appreciated in the chart we also have more finished basements in older houses than in newer. So in these two we have two very interesting factors that can tell us a lot about the living quality back then and the now.



Third Chart Comparisson of Number of Bedrooms / Live Area

On this third chart we can apreciate the comparisson between the living area of a house before 1980 and after compared to the number of bedrooms that it contains. And as We can observe in the chart most houses that were built before 1980 had more bedrooms but in some of these cases they had a little bit more space than the new houses which have less rooms and in the mayority of these less living area than older houses.



GRAND QUESTION 2

Can you build a classification model (before or after 1980) that has at least 90% accuracy for the state of Colorado to use (explain your model choice and which models you tried)?

During this part of the project i tried to be analitical on the differences that could exist between some of the classification models, but i did not find any spectacular or great difference between them but mostly related to their accuracy. I decided to use the "RandomForestClassifier" which gave me an accuracy of 92 percent, which means that is very precise and nearly exact.
some others that i tried and gave me similar results were:

- LogisticRegression
- GaussianNB
- VotingClassifier
  But i finally ended up opting for: "RandomForestClassifier"

TECHNICAL DETAILS

```
#%%
import pandas as pd
import numpy as np
import seaborn as sns
import altair as alt
# %%
dwellings_denver = pd.read_csv("https://github.com/byuidatascience/data4dwellings/raw/master/dat
dwellings_ml = pd.read_csv("https://github.com/byuidatascience/data4dwellings/raw/master/data-ra
dwellings_neighborhoods_ml = pd.read_csv("https://github.com/byuidatascience/data4dwellings/raw/
#%%
X_pred = dwellings_ml.drop(['before1980', 'yrbuilt', 'parcel'], axis=1)
y_pred = dwellings_ml.filter(["before1980"], axis = 1)
# Use of models...
X_train, X_test, y_train, y_test = train_test_split(
    X_pred,
    y_pred,
    test_size = .34,
    random_state = 76)
#%%
from sklearn.model_selection import cross_val_score
from sklearn.linear_model import LogisticRegression
from sklearn.naive_bayes import GaussianNB
from sklearn.ensemble import RandomForestClassifier
from sklearn.ensemble import VotingClassifier
clf = RandomForestClassifier(n_estimators=50, random_state=1)
clf = clf.fit(X_train, y_train)
y_pred = clf.predict(X_test)
print(metrics.classification_report(y_test, y_pred))
```

```
accuracy                           0.92
```

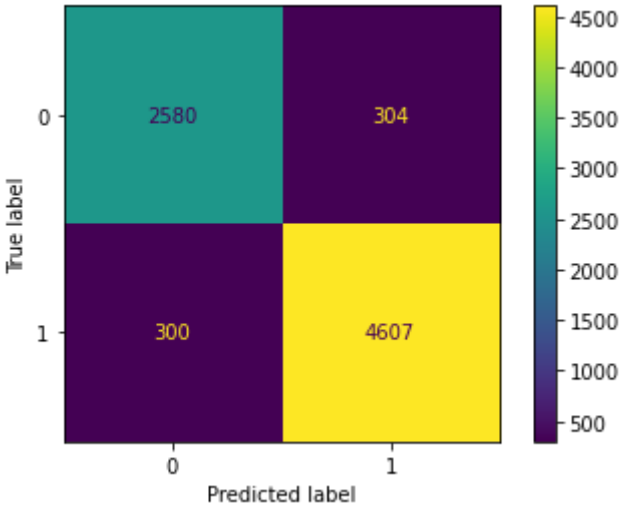
GRAND QUESTION 3

Will you justify your classification model by detailing the most important features in your model (a chart and a description are a must)?

By using the Classification model described ("RandomForestClassifier") we can obtain a really good percentage of accuracy, the following table helps us appreciate the True Positives, True Negatives, False Positives and False Negatives. And by the numbers each of these sections demostrate we can realize of the accuracy of the test the models can run.

TECHNICAL DETAILS

```
#%%
#question 3
from sklearn.model_selection import cross_val_score
from sklearn.linear_model import LogisticRegression
from sklearn.naive_bayes import GaussianNB
from sklearn.ensemble import RandomForestClassifier
from sklearn.ensemble import VotingClassifier
clf = RandomForestClassifier(n_estimators=50, random_state=1)
clf = clf.fit(X_train, y_train)
y_pred = clf.predict(X_test)
metrics.plot_confusion_matrix(clf, X_test, y_test)
```



GRAND QUESTION 4

Can you describe the quality of your classification model using 2-3 evaluation metrics? You need to provide an interpretation of each evaluation metric when you provide the value.

I decided that the two main evaluation metrics for this project specifically and that can better desccribe the Classification Model are: Accuracy and Recall. And i further identify each of them as factors of the model and its viability on the project.

- Accuracy: Accuracy is the ratio of the total number of correct predictions and the total number of predictions.
- Recall: The recall is the measure of our model correctly identifying True Positives. Thus, for all the patients who actually have heart disease, recall tells us how many we correctly identified.
  Both play a huge part to recognize our Model as viable and credible for the solutions we seek.

TECHNICAL DETAILS

```
from sklearn.model_selection import cross_val_score
from sklearn.linear_model import LogisticRegression
from sklearn.naive_bayes import GaussianNB
from sklearn.ensemble import RandomForestClassifier
from sklearn.ensemble import VotingClassifier
clf = RandomForestClassifier(n_estimators=50, random_state=1)
clf = clf.fit(X_train, y_train)
y_pred = clf.predict(X_test)
print(metrics.classification_report(y_test, y_pred))
```

```
    precision    recall

 0       0.90      0.89
 1       0.94      0.94
```

APPENDIX A (PYTHON CODE)

```
#%%
import pandas as pd
import numpy as np
import seaborn as sns
import altair as alt
#%%
from sklearn.model_selection import train_test_split
from sklearn import tree
from sklearn.naive_bayes import GaussianNB
from sklearn.ensemble import GradientBoostingClassifier
from sklearn import metrics
# %%
dwellings_denver = pd.read_csv("https://github.com/byuidatascience/data4dwellings/raw/master/dat
dwellings_ml = pd.read_csv("https://github.com/byuidatascience/data4dwellings/raw/master/data-ra
dwellings_neighborhoods_ml = pd.read_csv("https://github.com/byuidatascience/data4dwellings/raw/
#%%
h_subset = dwellings_ml.filter(['livearea', 'finbsmnt',
    'basement', 'yearbuilt', 'nocars', 'numbdrm', 'numbaths',
    'stories', 'yrbuilt', 'before1980']).sample(500)
sns.pairplot(h_subset, hue = 'before1980')
corr = h_subset.drop(columns = 'before1980').corr()
#%%
X_pred = dwellings_ml.drop(['before1980', 'yrbuilt', 'parcel'], axis=1)
y_pred = dwellings_ml.filter(["before1980"], axis = 1)
# Use of models...
X_train, X_test, y_train, y_test = train_test_split(
    X_pred,
    y_pred,
    test_size = .34,
    random_state = 76)
#%%
que_1 = dwellings_ml.filter(["livearea", "finbsmnt", "before1980"]).sample(500)
sns.pairplot(que_1, hue = "before1980")
#%%
que_2 = dwellings_ml.filter(["stories", "basement", "before1980"]).sample(500)
sns.pairplot(que_2, hue = "before1980")
#%%
que_3 = dwellings_ml.filter(["numbdrm", "livearea", "before1980"]).sample(500)
sns.pairplot(que_3, hue = "before1980")

# %%
# pregunta 2
from sklearn.model_selection import cross_val_score
from sklearn.linear_model import LogisticRegression
from sklearn.naive_bayes import GaussianNB
from sklearn.ensemble import RandomForestClassifier
from sklearn.ensemble import VotingClassifier
clf = RandomForestClassifier(n_estimators=50, random_state=1)
clf = clf.fit(X_train, y_train)
y_pred = clf.predict(X_test)
```

```
print(metrics.classification_report(y_test, y_pred))
#%%

#question 3
from sklearn.model_selection import cross_val_score
from sklearn.linear_model import LogisticRegression
from sklearn.naive_bayes import GaussianNB
from sklearn.ensemble import RandomForestClassifier
from sklearn.ensemble import VotingClassifier
clf = RandomForestClassifier(n_estimators=50, random_state=1)
clf = clf.fit(X_train, y_train)
y_pred = clf.predict(X_test)
metrics.plot_confusion_matrix(clf, X_test, y_test)
```