Client Report - [Insert Project Title]

Course CSE 250 Jame Lule

Elevator pitch

paste your elevator pitch here

GRAND QUESTION 1

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

type your results and analysis here

TECHNICAL DETAILS

```
#paste chart code in this snippet box
alt.data_transformers.disable_max_rows()
gartype_chart = (alt.Chart(denver)
    .encode(
        x = 'gartype',
        y = alt.Y('yrbuilt',
                  scale = alt.Scale(zero = False),
                  axis = alt.Axis(format='d'))
    )
    .mark_boxplot(
        size = 50
    )
    .properties(
        width = 900
    ))

gartype_chart

# Example 2
arcstyle_chart = (alt.Chart(denver)
    .encode(
        x = 'arcstyle',
        y = alt.Y('yrbuilt',
                 # scale = alt.Scale(zero = False),
                  #axis = alt.Axis(format='d')
                  )
    )
    .mark_boxplot(
        size = 50
    )
    .properties(
        width = 900
    ))
```

```
arcstyle_chart

#%%
####
alt.Chart(denver).mark_bar().encode(
    alt.X('livearea', bin=True),
    y='count()',
    color = 'yrbuilt'
)

# Example 3
numbaths_chart = (alt.Chart(denver)
    .encode(
        x = 'numbaths',
        y = alt.Y('yrbuilt',
                  scale = alt.Scale(zero = False),
                  axis = alt.Axis(format='d'))
    )
    .mark_boxplot(
        size = 50
    )
    .properties(
        width = 900
    ))

numbaths_chart
```

insert your chart png here insert your chart png here

```
#paste your table code in this snippet box
```

replace the table below with your table

	animal
0	elk
1	pig
2	dog
3	quetzal

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)?

type your results and analysis here

TECHNICAL DETAILS

```
#paste chart code in this snippet box
```

insert your chart png here

GRAND QUESTION 3

COPY PASTE GRAND QUESTION 3Â FROM THE PROJECT HERE

0.9022474361771765

TECHNICAL DETAILS

```
#paste chart code in this snippet box
```

insert your chart png here

```
#paste your table code in this snippet box
```

replace the table below with your table

	animal
0	elk
1	pig
2	dog
3	quetzal

GRAND QUESTION 4

# Will you justify your classification model by detailing

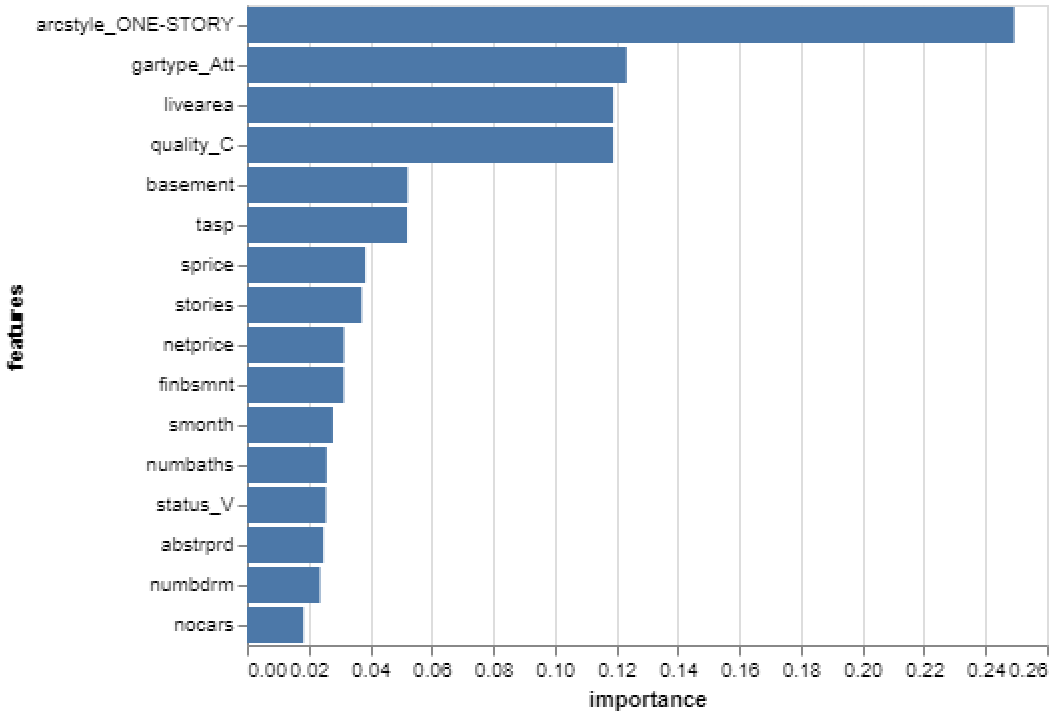
the most important features in your model (a chart and a description are a must)?

type your results and analysis here

TECHNICAL DETAILS

```
#paste chart code in this snippet box
```

insert your chart png here



GRAND QUESTION 5

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.

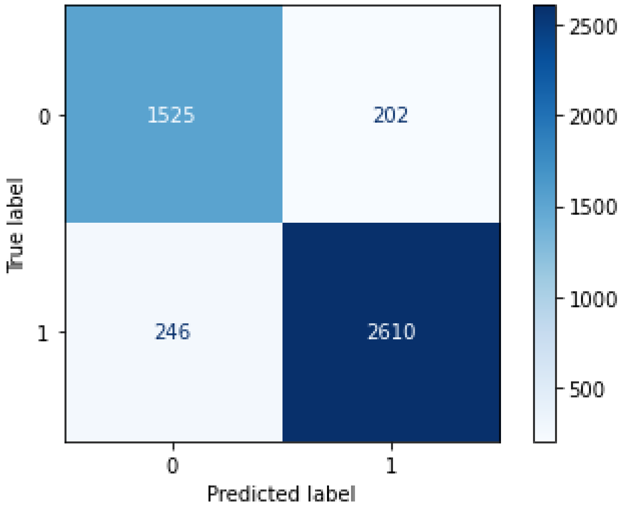
type your results and analysis here

```
precision    recall  f1-score   support

       0       0.86      0.88      0.87      1727
       1       0.93      0.91      0.92      2856

accuracy                           0.90      4583
```

macro avg 0.89 0.90 0.90 4583 weighted avg 0.90 0.90 0.90 4583

TECHNICAL DETAILS

```
#paste chart code in this snippet box
predictions = classifier.predict(x_test)
con_matrix = confusion_matrix(y_test, predictions)
plot_confusion_matrix(classifier, x_test, y_test, cmap = 'Blues')
```



insert your chart png here

~~~
## APPENDIX A (PYTHON CODE)
```python
#paste all your code from your python file (.py) here
#%%

import pandas as pd
import numpy as nps
import altair as alt
import json
from sklearn.model_selection import train_test_split
from sklearn import metrics
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import classification_report
from sklearn.naive_bayes import GaussianNB
from sklearn.ensemble import RandomForestClassifier
from sklearn.ensemble import GradientBoostingClassifier
from sklearn.metrics import confusion_matrix
from sklearn.metrics import plot_confusion_matrix

#%%


dwellings = 
pd.read_csv('https://raw.githubusercontent.com/byuidatascience/data4dwellings/master/data-raw/dwellings_ml/dwellings_ml.csv')
denver = 
pd.read_csv('https://raw.githubusercontent.com/byuidatascience/data4dwellings/master/data-raw/dwellings_denver/dwellings_denver.csv')
~~~

```
#%%
# Create 2-3 charts that evaluate potential relationships between the home
variables and before1980.
alt.data_transformers.disable_max_rows()
gartype_chart = (alt.Chart(denver)
    .encode(
        x = 'gartype',
        y = alt.Y('yrbuilt',
                  scale = alt.Scale(zero = False),
                  axis = alt.Axis(format='d'))
    )
    .mark_boxplot(
        size = 50
    )
    .properties(
        width = 900
    ))

gartype_chart

# Example 2
arcstyle_chart = (alt.Chart(denver)
    .encode(
        x = 'arcstyle',
        y = alt.Y('yrbuilt',
                 # scale = alt.Scale(zero = False),
                  #axis = alt.Axis(format='d')
                  )
    )
    .mark_boxplot(
        size = 50
    )
    .properties(
        width = 900
    ))

arcstyle_chart

#%%
####
alt.Chart(denver).mark_bar().encode(
    alt.X('livearea', bin=True),
    y='count()',
    color = 'yrbuilt'
)

# Example 3
numbaths_chart = (alt.Chart(denver)
    .encode(
        x = 'numbaths',
        y = alt.Y('yrbuilt',
                  scale = alt.Scale(zero = False),
                  axis = alt.Axis(format='d'))
    )
    .mark_boxplot(
        size = 50
    )
    .properties(
        width = 900
    ))

numbaths_chart
#%%
# 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)?
# Filtering the most important columns
x = dwellings.filter(['arcstyle_ONE-STORY', 'gartype_Att',
                      'quality_C', 'livearea', 'basement',
                      'tasp', 'stories', 'netprice', 'sprice',
                      'numbdrm', 'abstrprd', 'finbsmnt', 'numbaths',
                      'status_V', 'smonth', 'nocars'])
y = dwellings['before1980']

# Tuning Parameters
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size = 0.2,
random_state = 3450)

#create the model
classifier = DecisionTreeClassifier()

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

#make predictions
y_predictions = classifier.predict(x_test)

#test how accurate predictions are
metrics.accuracy_score(y_test, y_predictions)


#%%

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


# Feature importance
classifier.feature_importances_


feature_df = pd.DataFrame({'features':x.columns,
'importance':classifier.feature_importances_})
feature_df

chart = alt.Chart(feature_df).mark_bar().encode(
    x='importance:Q',
    y=alt.Y('features:N', sort='-x')
)
chart

# chart = (alt.Chart(x_train,
# title = 'Classification model'
# ).
# encode(
#     x = alt.X('f_names', title = 'f_names'),
#     y = alt.Y('f_values', title ='f_values')

# )
# .mark_bar()
# )
# chart

# %%

# 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.

# The confusion matrix

predictions = classifier.predict(x_test)
con_matrix = confusion_matrix(y_test, predictions)
plot_confusion_matrix(classifier, x_test, y_test, cmap = 'Blues')

# The table helping us understand the confusion matrix more deeply
print(metrics.classification_report(y_test, y_predictions))


# %%
```