

Technical Assessment

Contents

Instructions	1
Visualizing Data	2
Question 1	2
Question 2	2
Modeling Data	3
Question 3:	3
Question 4	3
Question 5	3
Question 6	4
Question 7	7
Manipulating Data	8
Question 8	8
Question 9	9
Question 10	9
Appendix	10
Explanation of Boston Housing Data	10

Instructions

This PDF contains a long form version of the questions in HackerRank in addition to a data appendix which is required to answer a few of the questions. Please refer to the HackerRank test to submit your answers. As you answer the questions in HackerRank, please keep a copy of your answers for your own reference. This will help you to discuss your answers during the technical interview. Responses should be your own; plagiarism will not be accepted.

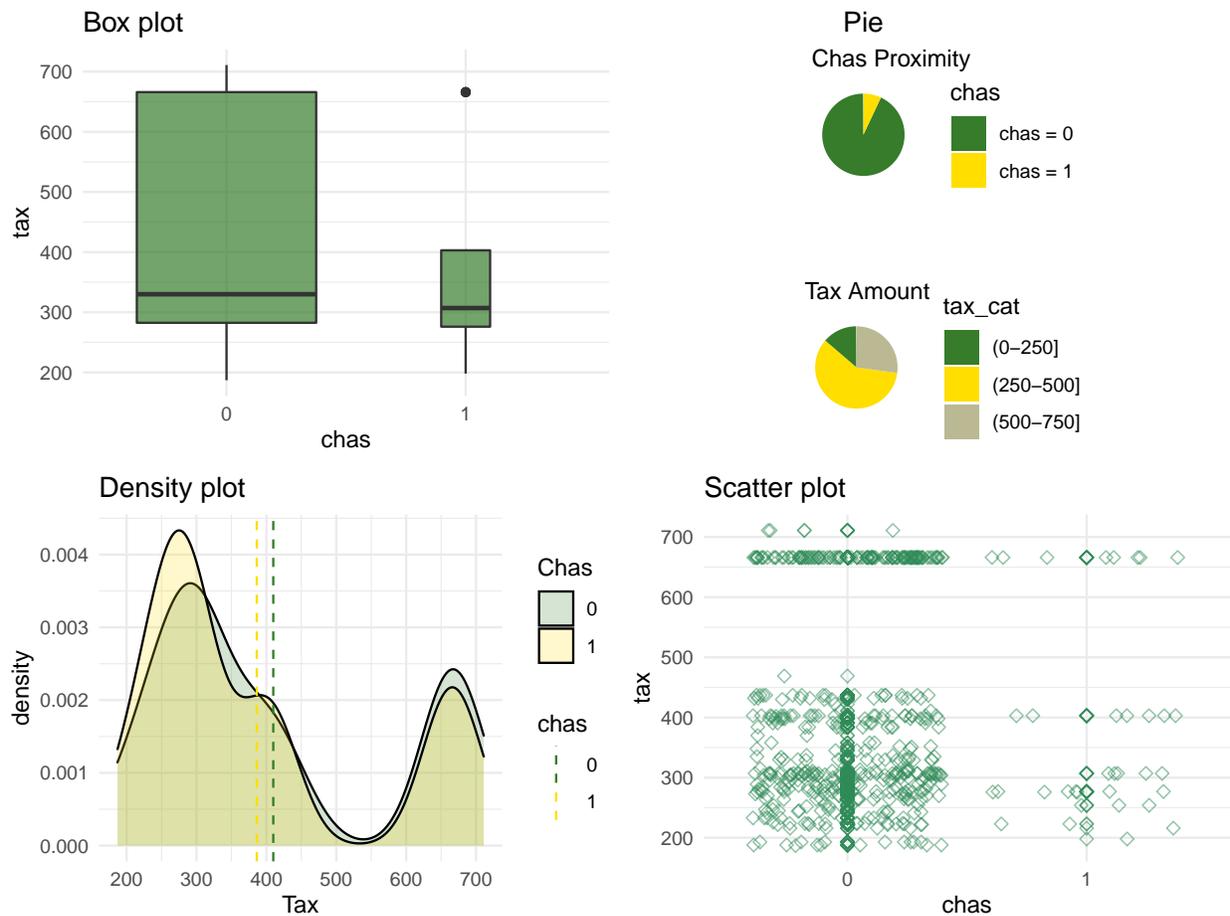
Visualizing Data

This section assesses the ability to interpret and communicate insights. The Boston Housing data, which is summarized in the appendix, was used to make the visuals in this section.

Question 1

Business leaders would like to understand the relationship between Charles River proximity and property tax rates, defined as the percent of tax paid in relation to home value. An analyst prepared the four visuals below.

- a) Do these visuals convey information about the property tax rates? If not, what changes would you make? (Remember to refer to the appendix for data details)
- b) Which of the chart types would you recommend including in the report and why?



Question 2

What cosmetic changes would you make to the visual you chose in the previous question to increase visual appeal and interpretability? In your report to business leaders, how would you describe the plot in one sentence?

Modeling Data

This section assesses the ability to think critically about variables and how they can be used to predict a desired outcome. Core competencies include understanding distributions in data, making appropriate data transformations, and selecting an appropriate model. The Boston Housing Data Exploration in the Appendix should be used for this section.

Question 3:

Predicting Tax:

- a) A **linear regression** model to predict **tax** was fit using **medv**, **rad**, **age** and **zn**. Based on the data summary in the appendix, what additional feature transformations or feature engineering would you consider to better prepare the data for a **linear regression** model? Consider only the **tax**, **rad**, **age** and **zn** variables.
- b) A co-worker is thinking about adding **cmdev** as an additional regressor to the model? Do you think this a good idea? Support your answer.

Question 4

Predicting River Proximity: Using the Boston Housing data, you want to predict which tracts are adjacent to the Charles River (as denoted by the **chas** variable). Propose an interpretable model to investigate the relationship between the covariates (x variables) and **chas** variable (y variable or target). Explain how you would use the model and its output to provide evidence of the strength and confidence in the relationship.

Question 5

A realtor thinks that if all other variables are held equal, a tract on the Charles River increases median home values by \$6,500. To test the realtor's hypothesis, you created a linear regression model with **chas** as a covariate (Note that **medv** is in \$1,000's of dollars). The coefficient associated with **chas** was 5.01 with a standard error of 0.84. Is there evidence to reject the realtor's claim at the 95% confidence level? Additionally, how would you defend this analysis with the realtor (who has no knowledge of linear models)?

Question 6

Predicting Property Tax Amounts: A local realtor wants to use the Boston Housing data to provide tax estimates for clients. A Data Scientist used the code below to fit a random forest model using the Boston Housing data.

- What percentage of the data is being used to train the model?
- After seeing the results the model developer is surprised to see that the error metric for Approach 3 is much higher than Approach 1 and Approach 2. Is the error value for Approach 3 correct, if not, how would you fix the value?
- A co-worker reviews the code and suggests that the error metrics in the results for Approach 1 and Approach 2 are optimistic (better than they actually are). Why might the co-worker have this belief?
- Approach 1 and 2 seem to have the same error? Is this result surprising? Support your answer.

(Note: there is no need to actually run the code shown below)

Results

approach_1	approach_2	approach_3
37.02471	37.02461	401.2936

```
# This is R language
# Ranger is a popular R package for building Random forest models
library(ranger)
library(dplyr)

df <- BostonHousing

# Ensure that results are reproducible
set.seed(1)
x_var <- c('chas', 'crim', 'zn', 'indus', 'nox', 'rm', 'rad', 'age', 'medv')
y_var <- 'tax'

# Selects Only x_var and y_var columns: y_var will be predicted with those in x_var
df <- df[, c(x_var, y_var)]

# `runif` produces a random number between 0 and 1
train_rows_boolean <- runif(nrow(df)) < .9

# Copy Data for different approaches
df_1 <- df
df_2 <- df
df_3 <- df

#####
# Model Training
#####

##### APPROACH 1 #####
# Note that `~ .` means predict tax with everything else in the data
model_1 <- ranger::ranger(tax ~ .,
                          data = df_1[train_rows_boolean,], seed = 1, num.trees=300)
```

```
##### APPROACH 2 #####
vars_to_normalize = c('crim', 'zn', 'indus', 'nox', 'rm', 'rad', 'age')

# For each variable observation, subtract the mean and divide by the standard deviation
normalize = function(x){
  values = (x - mean(x, na.rm=TRUE))/sd(x, na.rm=TRUE)
  return (values)
}

# Normalize All Columns in "vars_to_normalize"
for (var in vars_to_normalize){
  df_2[[var]] = normalize(df_2[[var]])
}

model_2 <- ranger::ranger(tax ~ .,
                          data = df_2[train_rows_boolean,], seed = 1, num.trees=300)

##### APPROACH 3 #####
df_3[y_var] <- log(df_3[y_var]) + 1
model_3 <- ranger::ranger(tax ~ .,
                          data = df_3[train_rows_boolean,], seed = 1, num.trees=300)

#####
# Predict On Data
#####
actuals <- BostonHousing[, c('tract', x_var, y_var)]

actuals[['model_1_predictions']] = predict(model_1, df_1)[['predictions']]
actuals[['model_2_predictions']] = predict(model_2, df_2)[['predictions']]
actuals[['model_3_predictions']] = predict(model_3, df_3)[['predictions']]

get_error = function(estimate, truth){
  return(mean(abs(truth - estimate)))
}

#####
# Calculate Performance
#####
results <- actuals %>%
summarise(approach_1 = get_error(model_1_predictions, actuals[[y_var]]),
          approach_2 = get_error(model_2_predictions, actuals[[y_var]]),
          approach_3 = get_error(model_3_predictions, actuals[[y_var]])
)
```

Python Option

```
# This is Python language
import numpy as np
import pandas as pd
from sklearn.ensemble import RandomForestRegressor

df = BostonHousing
x_var = ['chas', 'crim', 'zn', 'indus', 'nox', 'rm', 'rad', 'age', 'medv']
```

```

y_var = ['tax']

# Selects Only X and Y Columns
df = df[y_var + x_var]

# `np.random.uniform` produces a random number between 0 and 1
train_rows_boolean = [ x < .9 for x in np.random.uniform(size = df.shape[0])]

# Copy Dataframe for different approaches
df_1 = df.copy()
df_2 = df.copy()
df_3 = df.copy()

#####
# Model Training
#####
regr_1 = RandomForestRegressor(random_state=0, n_estimators = 300)
regr_2 = RandomForestRegressor(random_state=0, n_estimators = 300)
regr_3 = RandomForestRegressor(random_state=0, n_estimators = 300)

##### APPROACH 1 #####
model_1 = regr_1.fit(df_1.loc[train_rows_boolean, x_var],
                    y = np.squeeze(df_1.loc[train_rows_boolean, y_var]))

##### APPROACH 2 #####
vars_to_normalize = ['crim', 'zn', 'indus', 'nox', 'rm', 'rad', 'age']

# For each variable observation, subtract the mean and divide by the standard deviation
def normalize(x):
    values = (x - np.mean(x))/np.std(x)
    return(values)

# Normalize All Columns in "vars_to_normalize"
df_2[vars_to_normalize] = df_2[vars_to_normalize].apply(normalize, axis = 0)
model_2 = regr_2.fit(df_2.loc[train_rows_boolean, x_var],
                    y = np.squeeze(df_2.loc[train_rows_boolean, y_var]))

##### APPROACH 3 #####
df_3[y_var] = np.log(df_3[y_var]) + 1
model_3 = regr_3.fit(df_3.loc[train_rows_boolean, x_var],
                    y = np.squeeze(df_3.loc[train_rows_boolean, y_var]))

#####
# Predict On Data
#####
actuals = BostonHousing.loc[:, ['tract'] + y_var ]

actuals['model_1_predictions'] = model_1.predict(df_1[x_var])
actuals['model_2_predictions'] = model_2.predict(df_2[x_var])
actuals['model_3_predictions'] = model_3.predict(df_3[x_var])

```

```
def get_error(estimate, truth):
    return(np.mean(abs(truth - estimate)))

#####
# Calculate Performance
#####
results = {}
results["approach_1"] = get_error(actuals["model_1_predictions"], actuals[y_var[0]])
results["approach_2"] = get_error(actuals["model_2_predictions"], actuals[y_var[0]])
results["approach_3"] = get_error(actuals["model_3_predictions"], actuals[y_var[0]])

results = pd.DataFrame([results])
```

Question 7

Using Python or R, please simulate 365 hypothetical max daily temperatures for Boston and plot a 2-week rolling average of the temperatures. The temperatures do not have to exactly match historical values but should follow reasonable seasonal trends. Feel free to use a resource such as:

- <https://www.wunderground.com/>
- <https://weatherspark.com/y/26197/Average-Weather-in-Boston-Massachusetts-United-States-Year-Round>

Manipulating Data

This section assesses the ability to understand and improve upon existing code, with an emphasis on communication and efficiency. Effective code should accomplish its desired purpose in a clear and efficient way that is reproducible between coworkers.

Question 8

Please describe the code in a way that a non-technical co-worker can understand the purpose and outcome of the code. It may help to look at each step individually before providing a wholistic summary.

A preview of the dataframe “df” after step 1 is shown below:

uuid
7d2842f9-1a6c-4da2-9dcc-f0193cef1521
9169fe04-6eba-45fd-8183-3a8ef6424564
8f542817-19e4-4696-a251-6fd321a458a6
57c780a3-e65e-4227-ba5d-154b09a2562c
7bcaacee-011f-4288-af26-204e51c84121

```
import pandas as pd
from uuid import uuid4

# Step 1
df = pd.DataFrame([str(uuid4()) for _ in range(100_000)], columns=["uuid"])

print(df.head(), "\n\n")

# Step 2
for i, row in df.iterrows():
    current_uuid = row["uuid"]
    hex_group = 1
    hex_group_2 = ''
    for s in current_uuid:
        if s == '-':
            hex_group += 1
        else:
            if hex_group == 2:
                hex_group_2 += s

# Step 3
df.loc[i, "number"] = int(hex_group_2, 16)

# Step 4
df = df.sort_values("uuid").head(100)

# Step 5
total = 0
for i, row in df.iterrows():
    total += row["number"]
```

Question 9

Please propose your own R or Python code to more efficiently accomplish the task performed by the code in Question 8. Note: Changing the length of the input DataFrame (100k records) is not allowed!

Question 10

You are researching past home sales in particular neighborhoods using the data shown below. Using Python, SQL, or R, please write and submit code to answer parts a through d. Run the code using the provided data that is shown below and include the output values in your answer.

- A table/dataframe with columns that contain `house_id`, `address`, `neighborhood_name`. Results should be ordered by address ascending and limited to the top 10 results.
- A table/dataframe with a column that contains the total number of house sales in 1970.
- A table/dataframe with a column that contains the average sale price for houses with walkability score greater than 50, rounded to the nearest integer.
- A table/dataframe that lists the difference between the highest and lowest sale price for each neighborhood. Columns should include `neighborhood_name` and `price_difference`, with results ordered by `neighborhood_name` ascending.

Before submitting, please restart the kernel and run the notebook from beginning to end to ensure your code runs without error.

Table 3: houses

house_id	address	neighborhood_id
0	1	4862 Corbin Branch Road
1	2	1030 Mayo Street
2	3	2819 Duke Lane
3	4	1042 New Creek Road
4	5	3449 Tuesday Lane

Table 4: house_sales

house_id	date	sale_price
1	11/2/1978	81000
1	7/24/1975	80000
1	12/1/1971	74000
2	4/4/1970	102000
2	2/28/1975	112000

Table 5: neighborhoods

neighborhood_id	neighborhood_name	walkability_score
1	Oak Ridge	85
2	Meadowview	92
3	Lakewood	5
4	Park Place	44
5	Glenstone	18

Appendix

A preview of the Boston Housing data is shown below.

town	lon	lat	tract	age	chas	crim	indus	tax	nox	rm	rad	medv	cmedv	zn
Arlington	-71.0870	42.2416	3561	88.5	0	0.13914	4.05	296	0.51	5.572	5	23.1	23.1	0
Arlington	-71.0855	42.2450	3562	84.1	0	0.09178	4.05	296	0.51	6.416	5	23.6	23.6	0
Arlington	-71.0833	42.2475	3563	68.7	0	0.08447	4.05	296	0.51	5.859	5	22.6	22.6	0
Arlington	-71.0940	42.2575	3564	33.1	0	0.06664	4.05	296	0.51	6.546	5	29.4	29.4	0
Arlington	-71.1125	42.2550	3565	47.2	0	0.07022	4.05	296	0.51	6.020	5	23.2	23.2	0

A correlation matrix of all non categorical variables in the Boston Housing data is shown below.

	lon	lat	tract	age	crim	indus	tax	nox	rm	rad	medv	cmedv	zn
lon	1												
lat	0.14	1											
tract	-0.22	-0.23	1										
age	0.2	0.08	-0.49	1									
crim	0.07	-0.08	-0.55	0.35	1								
indus	0.06	-0.04	-0.58	0.64	0.41	1							
tax	0.05	-0.17	-0.79	0.51	0.58	0.72	1						
nox	0.16	-0.07	-0.57	0.73	0.42	0.76	0.67	1					
rm	-0.26	-0.07	0.31	-0.24	-0.22	-0.39	-0.29	-0.3	1				
rad	0.03	-0.21	-0.83	0.46	0.63	0.6	0.91	0.61	-0.21	1			
medv	-0.32	0.01	0.43	-0.38	-0.39	-0.48	-0.47	-0.43	0.7	-0.38	1		
cmedv	-0.32	0.01	0.43	-0.38	-0.39	-0.48	-0.47	-0.43	0.7	-0.38	1	1	
zn	-0.22	-0.13	0.37	-0.57	-0.2	-0.53	-0.31	-0.52	0.31	-0.31	0.36	0.36	1

Explanation of Boston Housing Data

The Boston Housing data used in this assessment is a selection of the original Boston Housing data and contains information about 506 Boston census tracts from the 1970 census.

variable	type	description
town	string	Name of town
lon	numeric	Longitude of census tract
lat	numeric	Latitude of census tract
tract	integer	Census tract
age	numeric	Proportion of owner-occupied units built prior to 1940
chas	factor/category	1 if tract bounds river, 0 otherwise)
crim	numeric	Per capita crime rate by town
indus	numeric	Proportion of non-retail business acres per town
tax	integer	Full-value property-tax rate per USD 10,000
nox	numeric	Nitric oxides concentration (parts per 10 million)
rm	numeric	Average number of rooms per dwelling
rad	integer	Index of accessibility to radial highways
medv	numeric	Median value of owner-occupied homes in USD 1000's
cmedv	numeric	Corrected median value of owner-occupied homes in USD 1000's
zn	numeric	Proportion of residential land zoned for lots over 25,000 sq.ft

Variable Histograms

