Module 2.3 – COMP 642-901 – Christian Ruiz, cr72

4. Download the data for the California Census and answer the following

a. What are the attributes for each district?

It seems that most of the districts contain float64 data on longitude, latitude, median age, total rooms, total bedrooms, population, households, median income, and median house value while ocean proximity is the only categorical feature.

b. What attributes are confusing to you?

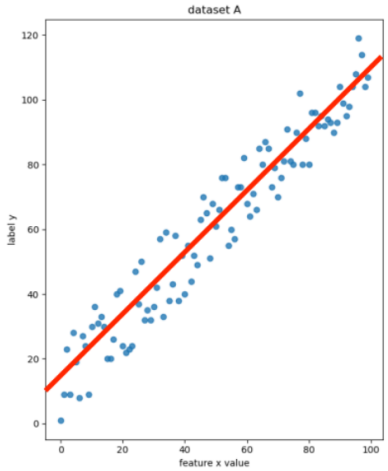
It seems that the names of some of the attributes may be confusing such as the median age. Also, ocean proximity may be hard to interoperate looking at <1H OCEAN and NEAR OCEAN. The two definitions seem difficult to differentiate.

c. Without graphing tools, what observations can you make about the data?

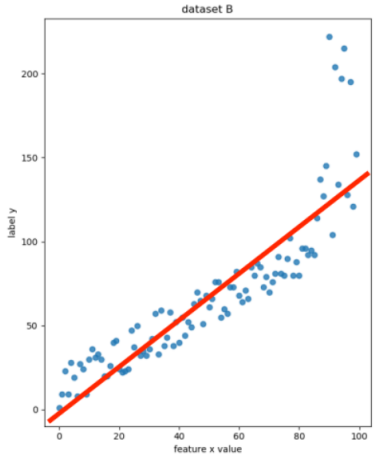
- There are some missing values for total bedrooms
- There are only 5 ISLAND homes and the largest count attribute is <1H OCEAN at 9,136 observations
- ISLAND homes have the greatest median age with INLAND homes having the lowest median age
- ISLAND homes have the lowest number of total rooms and total bedrooms compared to the other attributes
- ISLAND has the lowest number of households (unsure what this feature means)
- ISLAND has the lowest median income but also has the lowest std of median income compared to the other homes
- INLAND has the lowest median house values

5. Suppose you have two datasets A and B, they have exactly the same structure that each row contains a single feature x and its corresponding label y and you want to train a linear regression model to fit two datasets.

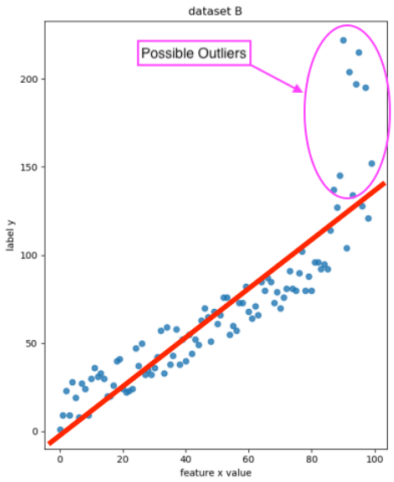
a. Draw the possible predicted regression line for dataset A. (You can draw by hand)



b. Draw the possible predicted regression line for dataset B. (You can draw by hand)

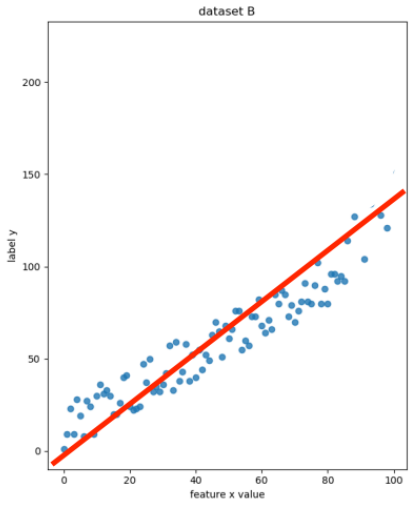


c. Which data points in dataset B are possible outliers, can you draw a circle around them?



d. Draw the possible predicted regression line for dataset B after removing the outliers.

(You can draw by hand)



e. Suppose we use the mean square error to evaluate the model on dataset B, how does removing possible outlier affect the mean square error?

The mean square error, compared to the root mean squared error, considers outliers more in it's evaluation by squaring simply the mean error calculation. Removing the outliers would essential lower the error value.

$$\sum_{i=1}^{N}(predicated_i - actual_i)^2 \frac{1}{N}$$

6. Run the notebook of housing price prediction and answer the following questions.

a. Question 1: Frame the Problem

Given a dataset like this, how can it be framed as a machine learning problem (try to frame it in different ways other than predicting housing price)? Is the problem you want to solve supervised learning or unsupervised learning? Classification problem or regression problem?

ANSWER:

There are a few predictions and/or approaches that we could make on this dataset.

1. We can try to use a supervised learning regression approach to predict the median income based on the other features of the data
2. We can also create unsupervised classification mini models of each ocean proximity to determine if there are underlying segments of data present within this categorical data.

b. Question 2 : Sign of Coefficients for Linear Regression

Given the equation of linear regression measure, the predicted dependent variable / target equals to the weighted sum of each independent variables / feature plus a bias / noise term, and the equation to predict the house price with all of the given features is shown below:

$$median\_house\_value = \beta_0 * longitude + \beta_1 * latitude + \beta_2 * housing\_median\_age + \beta_3 * total\_rooms +$$

$$\beta_4 * total\_bedrooms + \beta_5 * population + \beta_6 * households + \beta_7 * median\_income + \epsilon$$

Each feature weight is known as coefficient. The interpretation of a regression coefficient is that it represents the mean change in the dependent variable for each 1 unit change in an independent variable when you hold all of the other independent variables constant. And the machine learning is all about building algorithm to learn these coefficients and use the learned coefficients to predict future unseen data. Given the correlation matrix above. What can you concluded about the sign of each coefficient in this linear regression model?

```
median_house_value    1.000000
median_income         0.688075
total_rooms           0.134153
housing_median_age    0.105623
households            0.065843
total_bedrooms        0.049686
population           -0.024650
longitude            -0.045967
latitude             -0.144160
Name: median_house_value, dtype: float64
```

ANSWER:

The sign of each coefficient is directly related to the positive or negative correlation of that value (median_house_value) to the other features in the data. If there is positive correlation then we can assume that an increase of one unit in the independent variables will cause an increase change in the dependent variable. For negatively correlated features, an increase of one unit in the independent variable would cause a decrease change in the dependent variable.

c. Question 3: Try StratifiedShuffleSplit by Yourself

"median_income" was categorized into 5 groups and we use StratifiedShuffleSplit to make sure that the ratio of each group is exactly the same for training and test set. Apply the same method on feature "housing_median_age" to preserve the percentage of samples for training and test set. (Hint: You have to choose the number of categories and the split thresholds carefully to avoid generating skewed data, you can make the decision based on the output of describe() method, this method can show some important statistics for each feature)

```
housing['housing_median_age'].describe()
```

```
count    20640.000000
mean        28.639486
std         12.585558
min          1.000000
25%         18.000000
50%         29.000000
75%         37.000000
max         52.000000
Name: housing_median_age, dtype: float64
```

```
housing["age_cat"] = pd.cut(housing["housing_median_age"],
                            bins=[0., 17, 28, 37, 45, np.inf], # TODO :: split the
feature into different categories, expect 1 line of code
                            labels=[1, 2, 3, 4, 5]) # TODO :: label for each
category, expect 1 line of code)

# TODO :: use StratifiedShuffleSplit to split the dataset into training set and test
set while preserving
# the percentage of samples for each category.
# expect 4 lines of code
split = StratifiedShuffleSplit(n_splits=1, test_size=0.2, random_state=42)
for train_index, test_index in split.split(housing, housing["age_cat"]):
    strat_train_set = housing.iloc[train_index]
    strat_test_set = housing.iloc[test_index]

strat_test_set['age_cat'].value_counts() / len(strat_test_set)
```

```
3    0.265746
2    0.264293
1    0.230378
4    0.132510
5    0.107074
Name: age_cat, dtype: float64
```

```
housing['age_cat'].value_counts() / len(housing)
```

```
3    0.265843
2    0.264147
1    0.230426
4    0.132461
5    0.107122
Name: age_cat, dtype: float64
```

d. Question 4 : Pros and Cons of One-Hot-Encoding

One hot encoding is a way to transform categorical feature into the format that the model can take as input. One hot encoding has the advantage that the result is binary rather than ordinal and that everything sits in an orthogonal vector space. Can it be used for 'closest_city' feature? Why or why not?

ANSWER:

If there was a new feature called "closest_city", it would depend on how the data for that feature is represented. For example, if the data is city names then there may be too many city names represented and would cause too many columns. However, if closest_city was measured as a range distance categorical variable (i.e. 250 - 500)which is representative of a set of radial range distances from a central distance point, then it may be something we can measure.

e. Question 5 : Is Decision Tree Regressor a Great Model ?
The decision tree regressor has 0 error on the training set, is that a good model? why or why not ?

ANSWER:

The decision tree containing an error of 0 means that the data is overfit to the training data and will not predict the testing data appropriately.

f. Question 6 : Try Out SVR Model

Try a support vector machine regressor with various hyperparameters such as kernel = "linear" (with various values for the C hyperparameter) or kernel = "rbf" (with various values for the C and gamma hyperparameter). Don't worry about what these parameters mean for now. How does the best SVR predictor perform? You can refer to https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVR.html for more description.

```
from sklearn.svm import SVR
# attempt using linear

c_gamma_parameter_random_list = [(3, 5), (2, 2), (5, 3), (6, 4), (1.0, 1.0), (6, 1)]

for c, gamma in c_gamma_parameter_random_list:
    svm_reg = SVR(kernel="linear", C=c, gamma=gamma)
    svm_reg.fit(housing_prepared, housing_labels)
    housing_predictions = svm_reg.predict(housing_prepared)
    svm_mse = mean_squared_error(housing_labels, housing_predictions)
    svm_rmse = np.sqrt(svm_mse)
    print(f'c={c}\ngamma={gamma}\nrmse={svm_rmse}\n')
```

```
c=3
gamma=5
rmse=91903.71907928665

c=2
gamma=2
rmse=97510.23236584605

c=5
gamma=3
```

```
rmse=85364.97053253266

c=6
gamma=4
rmse=83280.80237421073

c=1.0
gamma=1.0
rmse=106016.94936243235

c=6
gamma=1
rmse=83280.80237421073
```

```python
# attempt using rbf

c_gamma_parameter_random_list = [(3, 5), (2, 2), (5, 3), (6, 4), (1.0, 1.0)]

for c, gamma in c_gamma_parameter_random_list:
    svm_reg = SVR(kernel="rbf", C=c, gamma=gamma)
    svm_reg.fit(housing_prepared, housing_labels)
    housing_predictions = svm_reg.predict(housing_prepared)
    svm_mse = mean_squared_error(housing_labels, housing_predictions)
    svm_rmse = np.sqrt(svm_mse)
    print(f'c={c}\ngamma={gamma}\nrmse={svm_rmse}\n')
```

```
c=3
gamma=5
rmse=118294.20739891082

c=2
gamma=2
rmse=118286.61155901091

c=5
gamma=3
rmse=118279.6432111982

c=6
gamma=4
rmse=118287.55143394705

c=1.0
gamma=1.0
rmse=118270.06577781877
```

ANSWER:

When our kernal is set to linear and C is set to larger values, we seem to get a better rmse at 83280 while changing gamma does not yield very significant changes. Although results are still better than the default settings, they are still not the best. When we set kernal to rbf our rmse results, essentially, do not change.