Machine Learning, Fall 2019: Project 2

Zhibo Sheng

**Github**

https://github.com/zhibosheng/machinelearning/tree/master/project2

**Header**:I use Python, and use pandas to read csv file, use numpy to manipulate the matrix, and use matplotlib to do data visualizations.

The dataset contains house sale prices for King County, which includes Seattle. It includes homes sold between May 2014 and May 2015. It’s a great dataset for evaluating simple regression models.

**Programming Task**: For the dataset, I use a linear regression model that uses the training data to train , and evaluate and report on the linear regression model performance.

**Dataset details**:

The dataset has 21613 simples, it has 20 feature vector and 1 label. The meaning of the feature vector is shown as below:

|  |  |
| --- | --- |
| **Variable** | **Description** |
| id | Unique id for each home sold |
| date | Date of the home sale |
| bedrooms | Number of bedrooms |
| bathroom | Number of bathrooms |
| Sqft\_living | Square footage of the apartments |
| Sqft\_lot | Square footage of the land space |
| floors | Number of floors |
| waterfront | Whether the apartment was overlooking the waterfront |
| view | The view of the property |
| condition | The condition of the apartment |
| Grade | The quality level of construction and design |
| Sqft\_above | The square footage of the interior housing space that is above ground level |
| Sqft\_basement | The square footage of the interior housing space that is below ground level |
| Yr\_built | The year the house war initially built |
| Yr\_renovated | The year of the house’s last renovation |
| Zipcode | What zipcode area the house is in |
| Lat | Latitude |
| Long | Longitude |
| Sqft\_living15 | The square footage of interior housing living space for the nearest 15 neighbors |
| Sqft\_lot15 | The square footage of the land lots of the nearest 15 neighbors |

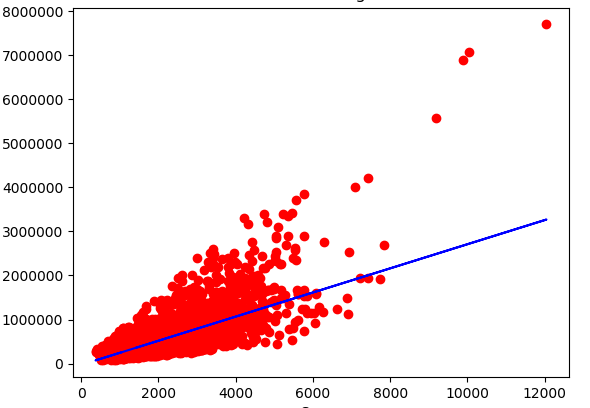
**Algorithm Description**

For the linear regression, I just use a single feature for my regression model. That is sqft\_living, which indicate the square footage of the apartments interior living space. And it does not has missing value, and I do not do data preprocessing and data normalization. And I split 80% data of dataset as training data and 20% data of dataset as test data.

**Algorithm Results:**

The result of the linear regression is that the w is 273 and b is -29039, and the average abs error is 172649.

The linear regression result and the test data distribution is shown as below



**Runtime**: Describe the run-time of your algorithm and also share the actual ”wall-clock” time that it took to compute your results

The time compliexity is O(n). For the linear regression, it take 0.0812s to finish the task.

**Decision Trees**

**2.1 ID3**

1. What is the sample entropy H(Y ) for this training data (with logarithms base 2)?

H(Y) = H(4/7,3/7) = -12/21\*lg(12/21)-9/21\*lg(9/21) = 0.4613456697472024+ 0.5238824662870492 = 0.9852281360342516

2.What are the information gains IG(X1) ≡ H(Y ) − H(Y |X1) and IG(X2) ≡ H(Y ) − H(Y |X2) for this sample of training data?

IG(X1) ≡ H(Y ) − H(Y |X1)

=0.9852281360342516–(8/21)\*(-7/8\*lg(7/8)-1/8\*lg(1/8))-(13/21)\*(-5/13\*lg(5/13)-8/13\*lg(8/13))

=0.9852281360342516–(8/21)\*(0.16856444319959643+0.375)–(13/21)\*(0.5301967781745115+0.4310398265483644)

= 0.9852281360342516-(8/21)\*(0.5435644431995964)-(13/21)(0.961236604722876)

= 0.9852281360342516- 0.20707216883794147-0.5950512314951137

= 0.18310473570119645

IG(X2) ≡ H(Y ) − H(Y |X2)

=0.9852281360342516-(10/21)\*(-7/10\*lg(7/10)-3/10\*lg(3/10))-(11/21)\*(-5/11\*lg(5/11)-6/11\*lg(6/11))

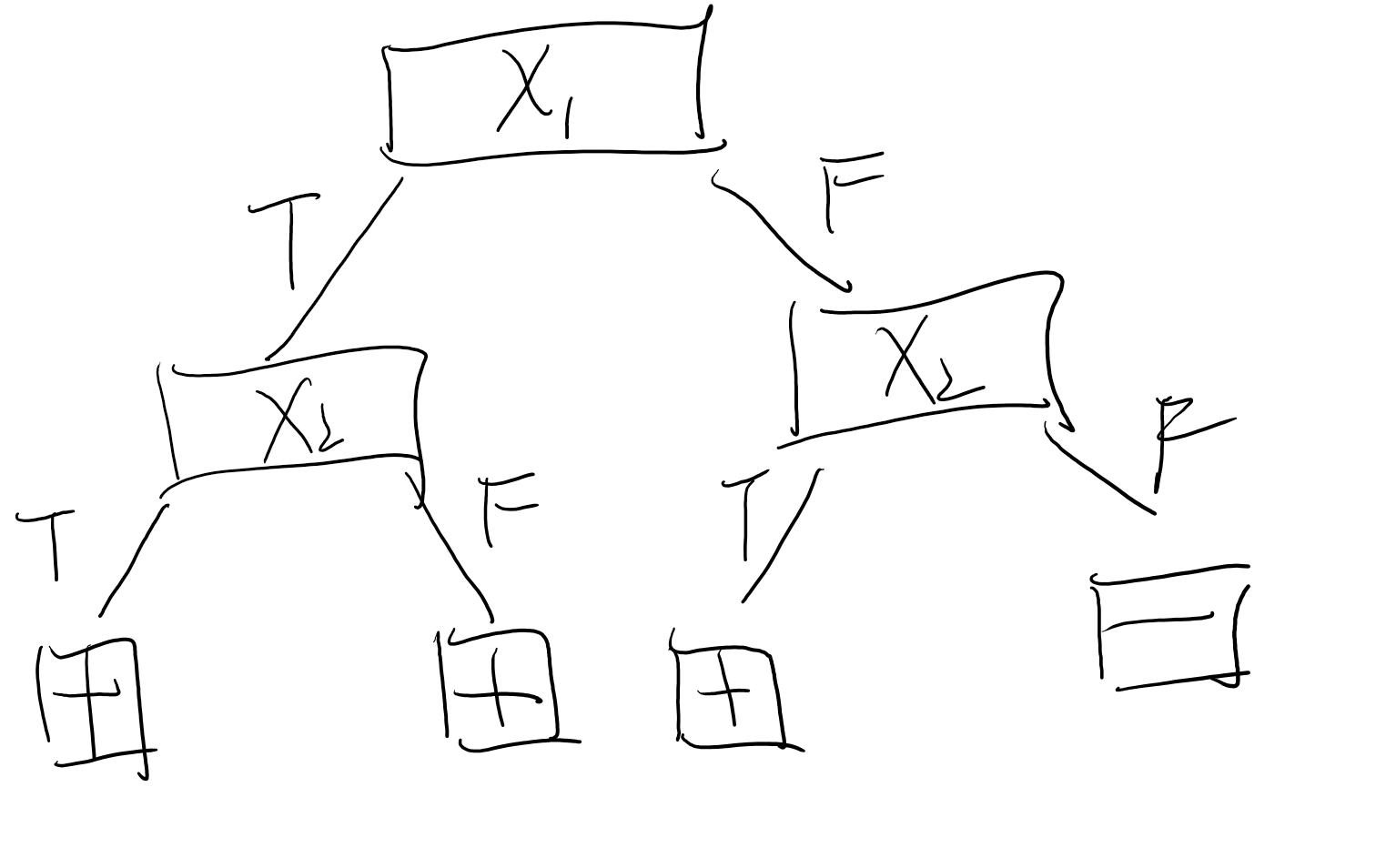
=0.9852281360342516–(10/21)\*(0.36020122098083085+0.5210896782498619)–(11/21)\*( 0.5170470562499704+0.47698315522698614)

=0.9852281360342516–(10/21)\*(0.8812908992306927)-(11/21)\*( 0.9940302114769566)

=0.9852281360342516 -0.4196623329669965-0.5206824917260249

=0.04488331134123025

3.Draw the decision tree that would be learned by ID3 (without postpruning) from this sample of training data



**Header**:I use Python, and use pandas to read csv file, use numpy to manipulate the matrix,

**Datasets**: The Wisconsin Diagnostic Breast Cancer (WDBC)is the dataset, which used for breast tumor diagnosis.

**Programming Task**: For the dataset, I use a SVM model that uses the training data to train , and evaluate and report on the SVM model performance

**Dataset details**:

The dataset has 569 simples, it has 10 feature vector and 1 label. The feature vector is radius,texture, perimeter, area, smoothness, compactness, concavity,concave point, symmetry, fractal dimension.

**Algorithm Description**

For the SVM, I first use PCA to reduce the 10 feature\_vector to 2. And set the label to 1 if the diagnosis is malignant, and set the label to -1 if the diagnosis is -1. And I split 2/3 data of dataset as training data and 1/3 data of dataset as test data. Finally, I train the SVM with 1000 epochs to make sure the weight is converge. For SVM, I use hinge loss as the loss fuction with l2 regulazation.

**Algorithm Results:**

The result of the SVM is that the weight is [0.3652532889155876, -0.0907289505878032]

For the test set, the TP is 58, the FP is 10, the FN is 10, the TN is 112, the total accuracy is 0.89.



**Runtime**: Describe the run-time of your algorithm and also share the actual ”wall-clock” time that it took to compute your results

The time compliexity is O(MN). N is the number of the data, and M is the epochs For the SVM, it take 2.81s to finish the task.