Dani DiTomasso
Professor Eleish
Data Analytics
15 November 2024

Data Analytics Lab 5 Written Document

Train SVM Linear Kernel (wine data)

```
> svm.mod0
Call:
svm(formula = Type ~ Alcohol + `Malic acid` + Ash + `Alcalinity of ash` +
    Magnesium + `Total phenols`, data = train, kernel = "linear")
Parameters:
   SVM-Type:  eps-regression
 SVM-Kernel:  linear
       cost:  1
      gamma:  0.1666667
    epsilon:  0.1
Number of Support Vectors:  114
```

Train SVM Polynomial Kernel (wine data)

```
> svm.mod1
Call:
svm(formula = Type ~ Alcohol + `Malic acid` + Ash + `Alcalinity of ash` +
    Magnesium + `Total phenols`, data = train, kernel = "polynomial")
Parameters:
   SVM-Type:  eps-regression
 SVM-Kernel:  polynomial
       cost:  1
     degree:  3
      gamma:  0.1666667
     coef.0:  0
    epsilon:  0.1
Number of Support Vectors:  110
```

Tuned SVM Polynomial Kernel (wine data)

```
> tuned.svm
Parameter tuning of ‘svm’:
- sampling method: 2-fold cross validation
- best parameters:
 gamma     cost
   0.3 0.015625
- best performance: 0.4079537
```

```
> svm.mod2
Call:
svm(formula = Type ~ Alcohol + `Malic acid` + Ash + `Alcalinity of ash` +
    Magnesium + `Total phenols`, data = train, kernel = "polynomial", gamma =
0.69,
    cost = 0.25)
Parameters:
   SVM-Type:  eps-regression
 SVM-Kernel:  polynomial
       cost:  0.25
     degree:  3
      gamma:  0.69
     coef.0:  0
    epsilon:  0.1
Number of Support Vectors:  114
```

CM for KNN (wine)

```
     Predicted
Actual  1  2  3
     1 18  2  1
     2  1 10  2
     3  2 10  8
```

Comparing SVM and KNN (wine data)

```
> print(all_metrics)
                     Model Precision      Recall        F1
1           SVM 0 (Linear) 0.3333333 0.009259259       NaN
2       SVM 1 (Polynomial) 0.3333333 0.009259259       NaN
3 SVM 2 (Tuned Polynomial) 0.3333333 0.009259259       NaN
4                KNN (k=2) 0.8527670 0.855131550 0.8538012
```

SVM Regression Model to predict Price based on Square Footage (NY-House data)

```
Call:
svm(formula = PRICE ~ PROPERTYSQFT, data = train_nyh, type = "eps-regression",
    kernel = "linear")
Parameters:
   SVM-Type:  eps-regression
 SVM-Kernel:  linear
       cost:  1
      gamma:  1
    epsilon:  0.1
Number of Support Vectors:  1583
```

Linear Model (NY-House data)

```
Call:
lm(formula = PRICE ~ PROPERTYSQFT, data = dataset_nyh)
Residuals:
      Min        1Q    Median        3Q       Max
-35520078  -1108628   -604214   -158340  58126608
Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  250170.17   71984.68   3.475 0.000515 ***
PROPERTYSQFT    743.16      22.39  33.197  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
Residual standard error: 3667000 on 4797 degrees of freedom
Multiple R-squared:  0.1868,   Adjusted R-squared:  0.1866
F-statistic:  1102 on 1 and 4797 DF,  p-value: < 2.2e-16
```

Compare the performance of Linear Model and SVM (NY-House data)

Based on the two plots returned for actual vs predicted price based on property square feet, the slope was steeper SVM Regression compared to the Linear Model. For this prediction, the closer the slope is to being 1 (aka centered from predicted price vs actual price) the more accurate the model would be - therefore, the Linear Model of the NY Housing dataset performed better than the SVM model.