Phang Teng Fone 1003296
50.039 - Hw 1

Regularization Tuning:

By applying 3 different types of kernel (Linear, Polynomial, RBF) and tuning the hyper parameters of the regularization constant, c. My results are shown using the validation set of data and using the classwise average accuracy performance measure:

Kernel	Regularization Constant						
	0.01	0.1	$0.1^{0.5}$	1	$10^{0.5}$	10	100
Linear	0.489	0.404	0.413	0.433	0.413	0.413	0.413
Polynomial	0.333	0.466	0.481	0.508	0.510	0.476	0.488
RBF	0.333	0.456	0.475	0.487	0.514	0.485	0.485

For the linear kernel, the best C value is when c = 0.01. From the table shown, it can be seen that the performance is hovering between the higher dimensional kernels, this means that the data is non-linear as the linearity performance is worse than that of the non-linear kernels. Therefore, based on the data from the table, using the polynomial or RBF would be a better choice to allow data to be solved in a higher dimensional space where it then can be linearly separable.

Training on train + validation:

By using the best C value of each kernel shown from the regularization tuning above, the results are shown using the test set and measured by the classwise average accuracy:

Kernel	Regularization Constant (c)	Performance
Linear	0.01	0.3903
Polynomial	$10^{0.5}$	0.4777
RBF	$10^{0.5}$	0.4492

Two performance measures on validation and test sets

Using the linear kernel and the best C value of 0.01, when trained on both the train plus validation gave the following results using Vanilla accuracy and classwise average accuracy:

Vanilla Accuracy

	Train	Train + Val
Test Performance	0.7991	0.7817
Validation Performance	0.7849	0.7442

Classwise Average Accuracy

	Train	Train + Val
Test Performance	0.4333	0.3903
Validation Performance	0.4891	0.4134

Both accuracy scores give us a different analysis of the performance of the data and classifier.By using vanilla accuracy, it only measures the correct number of predicted labels to the true label. By using classwise average accuracy it measures the average of true positive labels of each individual class. This benefits this particular experiment as the sample size of each class is different, using this average class accuracy ensures that the results are not skewed to the major class.

Theoretical, the train + val set performance should increase as compared to the train result as the model has more correctly labelled data to learn from, it should perform better. However in this particular case, the train + val performance dropped, perhaps it might be due to the reason that there might be overfitting on how the preprocessing might be done.

Validation score increases on the classwise average accuracy as the model was trained on the validation set as well. By performing prediction on a previously trained data, the model will be able to predict the same input better.

Code Output

----------Running linear kernel----------
C value: 0.01 , using classwise Accuracy: 0.4891
C value: 0.1 , using classwise Accuracy: 0.4037
C value: 0.31622776601683794 , using classwise Accuracy: 0.4134
C value: 1 , using classwise Accuracy: 0.4327
C value: 3.1622776601683795 , using classwise Accuracy: 0.4134
C value: 10 , using classwise Accuracy: 0.4134
C value: 100 , using classwise Accuracy: 0.4134
VALIDATION: {0.01: 0.48911111111111105, 0.1: 0.40370370370373, 0.31622776601683794: 0.41338271604938276, 1: 0.4327407407407408, 3.1622776601683795: 0.41338271604938276, 10: 0.41338271604938276, 100: 0.41338271604938276}
TEST: {0.01: 0.43332725836826436, 0.1: 0.37956381750804935, 0.31622776601683794: 0.39031650568009235, 1: 0.39031650568009235, 3.1622776601683795: 0.39031650568009235, 10: 0.39031650568009235, 100: 0.39031650568009235}

----Report-----
Best performance of C in class wise accuracy(Test score) is when c = 0.01 of value 0.4333 for train set only
Best performance of C in class wise accuracy(Val score) is when c = 0.01 of value 0.4891 for train set only
Using C = 0.01, applying the Vanilla Accuracy(Test score) will get you: 0.7991
Using C = 0.01, applying the Vanilla Accuracy(Val score) will get you: 0.7849

---Retraining SVM Model with best C value of 0.01 and train + validation set---
New Class Wise Accuracy(TEST score): 0.3903
New Vanilla Accuracy(TEST score): 0.7817

New Class Wise Accuracy(VAL score): 0.4134
New Vanilla Accuracy(VAL score): 0.7442

----------Running poly kernel----------
C value: 0.01 , using classwise Accuracy: 0.3333
C value: 0.1 , using classwise Accuracy: 0.4661
C value: 0.31622776601683794 , using classwise Accuracy: 0.4811
C value: 1 , using classwise Accuracy: 0.5085
C value: 3.1622776601683795 , using classwise Accuracy: 0.5101
C value: 10 , using classwise Accuracy: 0.4758
C value: 100 , using classwise Accuracy: 0.4881
VALIDATION: {0.01: 0.3333333333333333, 0.1: 0.46607407407407403, 0.31622776601683794: 0.48111111111111104, 1: 0.5084691358024691,

3.1622776601683795: 0.5101234567901235, 10: 0.4757777777777777, 100: 0.48812345679012337}
TEST: {0.01: 0.3333333333333333, 0.1: 0.4707230076284204, 0.31622776601683794: 0.49733135463042516, 1: 0.46882241141399145, 3.1622776601683795: 0.48145833875740934, 10: 0.4776918604146598, 100: 0.4776918604146598}

----Report-----
Best performance of C in class wise accuracy(Test score) is when c = 3.1622776601683795 of value 0.4815 for train set only
Best performance of C in class wise accuracy(Val score) is when c = 3.1622776601683795 of value 0.5101 for train set only
Using C = 3.1622776601683795, applying the Vanilla Accuracy(Test score) will get you: 0.8166
Using C = 3.1622776601683795, applying the Vanilla Accuracy(Val score) will get you: 0.7791

---Retraining SVM Model with best C value of 3.1622776601683795 and train + validation set---
New Class Wise Accuracy(TEST score): 0.4777
New Vanilla Accuracy(TEST score): 0.8079

New Class Wise Accuracy(VAL score): 0.4881
New Vanilla Accuracy(VAL score): 0.6802

------------------------------Running rbf kernel------------------------------
C value: 0.01 , using classwise Accuracy: 0.3333
C value: 0.1 , using classwise Accuracy: 0.4564
C value: 0.31622776601683794 , using classwise Accuracy: 0.4747
C value: 1 , using classwise Accuracy: 0.4871
C value: 3.1622776601683795 , using classwise Accuracy: 0.5144
C value: 10 , using classwise Accuracy: 0.4854
C value: 100 , using classwise Accuracy: 0.4854
VALIDATION: {0.01: 0.3333333333333333, 0.1: 0.45641975308641974, 0.31622776601683794: 0.47474074074074074, 1: 0.4870864197530864, 3.1622776601683795: 0.5144444444444444, 10: 0.4854320987654321, 100: 0.4854320987654321}
TEST: {0.01: 0.3333333333333333, 0.1: 0.45245471981393254, 0.31622776601683794: 0.4365817039409166, 1: 0.49546547250210454, 3.1622776601683795: 0.47582597828633916, 10: 0.44731703506990544, 100: 0.44920027424128023}

----Report-----
Best performance of C in class wise accuracy(Test score) is when c = 3.1622776601683795 of value 0.4758 for train set only
Best performance of C in class wise accuracy(Val score) is when c = 3.1622776601683795 of value 0.5144 for train set only
Using C = 3.1622776601683795, applying the Vanilla Accuracy(Test score) will get you: 0.8122
Using C = 3.1622776601683795, applying the Vanilla Accuracy(Val score) will get you: 0.7791

---Retraining SVM Model with best C value of 3.1622776601683795 and train + validation set---
New Class Wise Accuracy(TEST score): 0.4492
New Vanilla Accuracy(TEST score): 0.8035

New Class Wise Accuracy(VAL score): 0.4854
New Vanilla Accuracy(VAL score): 0.6919

Process finished with exit code 0