“Fall 2019: CSCI 6522, Adv. ML-II Programming Assignment #3”.

Name : Priyanka Velamala ID: 2547756

SVM Function

Time taken to build model: 5.43 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances	6991	95.8853 %
Incorrectly Classified Instances	300	4.1147 %
Kappa statistic	0.9539	
Mean absolute error	0.1603	
Root mean squared error	0.2721	
Relative absolute error	89.8016 %	
Root relative squared error	91.0789 %	
Total Number of Instances	7291	

=== Detailed Accuracy By Class ===

	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
	0.989	0.006	0.970	0.989	0.979	0.975	0.998	0.982	0
	0.997	0.002	0.990	0.997	0.994	0.993	0.999	0.991	1
	0.938	0.008	0.930	0.938	0.934	0.927	0.985	0.917	2
	0.954	0.006	0.943	0.954	0.949	0.944	0.991	0.926	3
	0.956	0.006	0.940	0.956	0.948	0.942	0.989	0.914	4
	0.901	0.005	0.940	0.901	0.920	0.914	0.982	0.882	5
	0.959	0.003	0.967	0.959	0.963	0.959	0.996	0.951	6
	0.967	0.005	0.954	0.967	0.961	0.957	0.995	0.950	7
	0.908	0.004	0.953	0.908	0.930	0.925	0.986	0.896	8
	0.958	0.002	0.976	0.958	0.967	0.964	0.994	0.951	9
Weighted Avg.	0.959	0.005	0.959	0.959	0.959	0.954	0.992	0.943	

=== Confusion Matrix ===

a	b	c	d	e	f	g	h	i	j	<-- classified as
1181	0	6	1	2	2	2	0	0	0	a = 0
0	1002	0	0	0	1	0	1	1	0	b = 1
6	0	686	8	14	2	3	2	9	1	c = 2
1	1	9	628	0	10	0	3	6	0	d = 3
1	2	11	0	623	2	4	4	2	3	e = 4
12	0	3	18	4	501	13	0	4	1	f = 5
10	1	9	0	3	4	637	0	0	0	g = 6
1	0	1	1	7	0	0	624	2	9	h = 7
6	5	13	7	4	10	0	4	492	1	i = 8
0	1	0	3	6	1	0	16	0	617	j = 9

Simple logistic Function

Time taken to build model: 77.08 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances	6904	94.6921 %
Incorrectly Classified Instances	387	5.3079 %
Kappa statistic	0.9405	
Mean absolute error	0.0162	
Root mean squared error	0.0907	
Relative absolute error	9.0828 %	
Root relative squared error	30.3422 %	
Total Number of Instances	7291	

=== Detailed Accuracy By Class ===

	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
	0.983	0.006	0.972	0.983	0.978	0.973	0.999	0.997	0
	0.992	0.002	0.988	0.992	0.990	0.988	0.999	0.994	1
	0.930	0.005	0.950	0.930	0.940	0.933	0.991	0.975	2
	0.930	0.007	0.932	0.930	0.931	0.924	0.996	0.977	3
	0.931	0.009	0.906	0.931	0.918	0.910	0.992	0.946	4
	0.892	0.007	0.913	0.892	0.903	0.895	0.990	0.953	5
	0.950	0.005	0.950	0.950	0.950	0.945	0.998	0.986	6
	0.958	0.006	0.941	0.958	0.949	0.944	0.996	0.984	7
	0.889	0.006	0.918	0.889	0.903	0.896	0.992	0.950	8
	0.943	0.005	0.945	0.943	0.944	0.939	0.997	0.976	9
Weighted Avg.	0.947	0.006	0.947	0.947	0.947	0.941	0.996	0.977	

=== Confusion Matrix ===

a	b	c	d	e	f	g	h	i	j	<-- classified as
1174	0	2	3	4	5	4	0	2	0	a = 0
0	997	0	0	0	4	0	1	3	0	b = 1
2	1	680	12	13	1	5	7	9	1	c = 2
3	0	10	612	2	13	0	3	10	5	d = 3
3	4	6	0	607	5	10	4	4	9	e = 4
11	0	4	17	8	496	11	0	4	5	f = 5
7	2	5	0	8	6	631	1	4	0	g = 6
0	0	1	0	8	1	0	618	3	14	h = 7
7	1	8	11	12	12	3	5	482	1	i = 8
1	4	0	2	8	0	0	18	4	607	j = 9

Multilayer Perception Function

Time taken to build model: 2445.07 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances	7102	97.4078 %
Incorrectly Classified Instances	189	2.5922 %
Kappa statistic	0.971	
Mean absolute error	0.0069	
Root mean squared error	0.0674	
Relative absolute error	3.8735 %	
Root relative squared error	22.552 %	
Total Number of Instances	7291	

=== Detailed Accuracy By Class ===

	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
	0.989	0.004	0.982	0.989	0.985	0.983	1.000	0.998	0
	0.996	0.001	0.993	0.996	0.995	0.994	1.000	0.998	1
	0.963	0.003	0.975	0.963	0.969	0.966	0.996	0.987	2
	0.962	0.002	0.978	0.962	0.970	0.967	0.999	0.991	3
	0.965	0.004	0.959	0.965	0.962	0.958	0.997	0.977	4
	0.951	0.004	0.953	0.951	0.952	0.948	0.995	0.979	5
	0.977	0.003	0.970	0.977	0.974	0.971	0.999	0.995	6
	0.977	0.003	0.974	0.977	0.975	0.973	0.995	0.991	7
	0.958	0.003	0.961	0.958	0.959	0.956	0.996	0.983	8
	0.974	0.003	0.974	0.974	0.974	0.971	0.999	0.993	9
Weighted Avg.	0.974	0.003	0.974	0.974	0.974	0.971	0.998	0.991	

=== Confusion Matrix ===

a	b	c	d	e	f	g	h	i	j	<-- classified as
1181	0	2	2	2	4	3	0	0	0	a = 0
0	1001	0	0	0	2	0	1	1	0	b = 1
4	1	704	2	6	2	3	3	5	1	c = 2
1	0	3	633	1	9	0	3	7	1	d = 3
3	2	3	0	629	1	8	0	0	6	e = 4
7	0	3	5	2	529	5	0	3	2	f = 5
3	1	2	0	5	2	649	0	2	0	g = 6
0	0	2	0	2	1	0	630	3	7	h = 7
3	3	2	3	4	4	1	3	519	0	i = 8
1	0	1	2	5	1	0	7	0	627	j = 9

Naive Bayes Function

Time taken to build model: 0.4 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances	5699	78.1649 %
Incorrectly Classified Instances	1592	21.8351 %
Kappa statistic	0.7559	
Mean absolute error	0.0436	
Root mean squared error	0.2079	
Relative absolute error	24.4011 %	
Root relative squared error	69.5665 %	
Total Number of Instances	7291	

=== Detailed Accuracy By Class ===

	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
	0.853	0.006	0.964	0.853	0.905	0.890	0.991	0.963	0
	0.984	0.006	0.966	0.984	0.975	0.971	0.993	0.962	1
	0.808	0.015	0.859	0.808	0.833	0.815	0.973	0.857	2
	0.667	0.009	0.885	0.667	0.761	0.749	0.960	0.852	3
	0.334	0.011	0.741	0.334	0.461	0.468	0.920	0.565	4
	0.529	0.014	0.758	0.529	0.623	0.609	0.940	0.634	5
	0.917	0.025	0.784	0.917	0.845	0.832	0.973	0.793	6
	0.893	0.006	0.932	0.893	0.912	0.904	0.984	0.925	7
	0.729	0.063	0.481	0.729	0.580	0.552	0.911	0.540	8
	0.885	0.084	0.505	0.885	0.643	0.628	0.938	0.513	9
Weighted Avg.	0.782	0.021	0.815	0.782	0.781	0.770	0.964	0.792	

=== Confusion Matrix ===

a	b	c	d	e	f	g	h	i	j	<-- classified as
1018	0	10	6	7	9	16	1	126	1	a = 0
0	989	2	0	0	0	1	0	6	7	b = 1
5	1	591	19	9	10	27	3	65	1	c = 2
6	1	30	439	4	19	3	9	126	21	d = 3
0	6	6	0	218	18	15	1	9	379	e = 4
19	0	12	26	12	294	104	3	72	14	f = 5
6	5	20	0	1	14	609	0	9	0	g = 6
0	4	3	3	9	0	0	576	3	47	h = 7
2	9	14	3	4	20	2	4	395	89	i = 8
0	9	0	0	30	4	0	21	10	570	j = 9

Logistic Function

Time taken to build model: 148.5 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances	6460	88.6024 %
Incorrectly Classified Instances	831	11.3976 %
Kappa statistic	0.8723	
Mean absolute error	0.0227	
Root mean squared error	0.1485	
Relative absolute error	12.7388 %	
Root relative squared error	49.6942 %	
Total Number of Instances	7291	

=== Detailed Accuracy By Class ===

	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
	0.943	0.010	0.946	0.943	0.945	0.934	0.993	0.974	0
	0.986	0.009	0.945	0.986	0.965	0.960	0.998	0.989	1
	0.799	0.011	0.886	0.799	0.840	0.825	0.955	0.874	2
	0.842	0.010	0.894	0.842	0.867	0.855	0.982	0.912	3
	0.842	0.016	0.839	0.842	0.841	0.825	0.975	0.839	4
	0.818	0.016	0.810	0.818	0.814	0.799	0.976	0.850	5
	0.923	0.011	0.891	0.923	0.907	0.897	0.988	0.944	6
	0.907	0.015	0.857	0.907	0.881	0.870	0.984	0.902	7
	0.799	0.014	0.820	0.799	0.809	0.794	0.966	0.833	8
	0.885	0.013	0.866	0.885	0.876	0.863	0.988	0.901	9
Weighted Avg.	0.886	0.012	0.886	0.886	0.885	0.874	0.982	0.913	

=== Confusion Matrix ===

a	b	c	d	e	f	g	h	i	j	<-- classified as
1126	3	10	6	9	16	11	6	6	1	a = 0
0	991	3	0	3	0	1	5	2	0	b = 1
12	13	584	17	29	5	14	25	22	10	c = 2
4	2	15	554	2	36	4	16	19	6	d = 3
7	7	14	4	549	12	15	8	17	19	e = 4
13	5	4	19	10	455	24	6	11	9	f = 5
12	4	8	0	10	13	613	2	2	0	g = 6
1	7	4	3	7	2	2	585	8	26	h = 7
12	15	17	14	9	15	3	7	433	17	i = 8
3	2	0	3	26	8	1	23	8	570	j = 9

This programming assignment used 5 fold cross validations to train and measure the performance of a classifier.

The 5 different classifiers used to compare the performances are

1. SVM
2. Logistic
3. Simple Logistic
4. Multilayer Perception
5. Naive Bayes

1. SVM

A Support Vector Machine (SVM) is a discriminative classifier formally defined by a separating hyperplane. In other words, given labeled training data (*supervised learning*), the algorithm outputs an optimal hyperplane which categorizes new examples. In two dimentional space this hyperplane is a line dividing a plane in two parts where in each class lay in either side.

2.Logistic

The logistic regression is called the sigmoid function. It is S-shaped curve whose value is between 0 and 1 but never exactly at those limits.

It is a statistical method for analysing a data set in which there are one or more independent variables that determine an outcome. The outcome is measured with a dichotomous variable (in which there are only two possible outcomes). The goal of logistic regression is to find the best fitting model to describe the relationship between the dichotomous characteristic of interest (dependent variable = response or outcome variable) and a set of independent (predictor or explanatory) variables. This is better than other binary classification like nearest neighbor since it also explains quantitatively the factors that lead to classification.

3. Simple Logistic

Simple logistic regression is analogous to linear regresion , except that the dependent variable is nominal, not a measurement. One goal is to see whether the probability of getting a particular value of the nominal variable is associated with the measurement variable; the other goal is to predict the probability of getting a particular value of the nominal variable, given the measurement variable.

4. Multilayer Perception

A multilayer perceptron (MLP) is a class for feed forward artificial neural network which consists

of three layers of nodes. Except for the input nodes, each node is a neuron that uses a nonlinear activation function. MLP utilizes a supervised learning technique called backpropagation for training. Its multiple layers and non-linear activation distinguish MLP from a linear perceptron. It can distinguish data that is not a linearly separable.

5. Naive Bayes

It is a classification technique based on Bayes Theorem with an assumption of independence among predictors. In simple terms, a Naive Bayes classifier assumes that the presence of a particular feature in a class is unrelated to the presence of any other feature. Even if these features depend on each other or upon the existence of the other features, all of these properties independently contribute to the probability. Naive Bayes model is easy to build and particularly useful for very large data sets. Along with simplicity, Naive Bayes is known to outperform even highly sophisticated classification methods.

The parameters /hyper- parameters that have been chosen to train the classifiers:

FC	Dg	DCC	MI	NDP	RI	UCGD	EOP	HS	UA	UC	Other
SVM	100	False	-	2	1.0E-8	False	-	-	-	-	EP-1.0E-12 NP: [-1] BCB-False
LOGISTIC	100	False	-1	4	-	-	-	-	-	-	-
SIMPLE LOGISTIC	100	False	500	2	-	True	False	50	False	True	WTB: 0.0
NAIVE BAYE	100	False	-	2	-	-	-	-	-	-	DMIOF-False UKE-False USD-False
MULTILAYERPERCEPTION	100	False	-	2	-	-	-	-	-	-	GUI – False Auto Build – False Decay – False HL – False LR – False MOMT – False NTB – True NAT - True

											NMC - True RST – True Seed: 0 TT: 500 VSS: 0 VTH:20

Where,

FC – Function Classifiers
BS – Batch Size
Dg – Debug
DCC – Do not check capabilities
MI – Max Iterations
NDP – Number of Decimal Places
Ri – Ridge
UCGD – Use Conjugate Gradient Descent
NP – Num Folds
DMIOF – Display Model in Old Format
UKE – User Kernel Estimator
USD – Use Supervised Discretization
HL - Hidden Layers
LR – Learning Rate
EOP – Error on Probabilities
HS – Heuristic Stop
UA – Use AIC
UC – Use Cross Validation
WTB – Weight Trim Beta
BCB – Build Calibrator Model
Ep – Epsilon
MOMT – Momentum
NTB – Nominal to Binary
NAT – Normalize Attribute
NMC – Normalize Numeric Class
RST – Reset
TT – Training Time
Validation Set Size: 0
VTH: Validation Threshold

****

The terms used for measuring the performance of a classifier:

1**. TP Rate**: True Positive Rate is aso called as sensitivity, is the ratio of positive data points that are correctly predicted as positive, with respect to all positive data points.

True Positive Rate(TPR)= TP/(TP+FN)

2. **FP Rate** : False Positive Rate is the calculated as the ratio between the number of negative events wrongly categorized as positive(false positives) and the total number of actual negative events.

False Positive Rate(FPR)= FP/(TN+FP)

3. **Precision** : It is also known as the positive predictive value, is the number of correct positive results divided by the number of positive results predicted by the classifier. That is, precision is defined as the accuracy of the judgement.

Precision= TP/(TP+FP)

4.**Recall:** It is referred to as sensitivity which is a true possitive value rate. It is what fraction of those that are actually positive were predicted positive

5.**F-Measure:** It is also known as F-score , which is composed of precision and recall, both calculated as percentages and combined as harmonic mean to assign a single number, easy for comprehension. This is also known as F- measure or balanced F-score

F-score= 2((precision*recall)/(precision+recall))

6.**ROC Area**: Receiver Operating Characteristic Area is the graphical plot that illustrates the diagnostic ability of a binary classifier system as its discrimination threshold is varied

7**. Confusion Matrix**: It is the table that describes the performance of a classification model or classifier on a set of test data for which the true values are known.[1][2][3][4].

The performance of the classifiers in terms of correctly classified instances

Classifiers	Correctly classified instances
SVM	6991
LOGISTIC	6460
SIMPLE LOGISTIC	6904
NAIVE BAYES	5699
MULTILAYER PERCEPTION	7102

****

The performance of the classifiers in terms of time taken to build the model

Classifiers	Time taken to build the model
SVM	5.43 seconds
LOGISTIC	148.5 seconds
SIMPLE LOGISTIC	77.08 seconds
NAIVE BAYES	0.4 seconds
MULTILAYER PERCEPTION	2445.07 seconds

The performance of the classifiers in terms of TP Rate, FP Rate, Precision, Recall, F-Measure, ROC Area.

	SVM	LOGISTIC	SIMPLE LOGISTIC	NAIVE BAYES	MULTILAYER PERCEPTION
TP Rate	0.959	0.886	0.947	0.782	0.974
FP Rate	0.005	0.012	0.006	0.021	0.003
Precision	0.959	0.886	0.947	0.815	0.974
Recall	0.959	0.886	0.947	0.782	0.974
F-Measure	0.959	0.885	0.947	0.781	0.974
ROC Area	0.992	0.982	0.996	0.964	0.998

The performance of the classifiers in terms of confusion matrix

1. SVM

=== Confusion Matrix ===

```
   a    b    c    d    e    f    g    h    i    j   <-- classified as
1181    0    6    1    2    2    2    0    0    0 |    a = 0
   0 1002    0    0    0    1    0    1    1    0 |    b = 1
   6    0  686    8   14    2    3    2    9    1 |    c = 2
   1    1    9  628    0   10    0    3    6    0 |    d = 3
   1    2   11    0  623    2    4    4    2    3 |    e = 4
  12    0    3   18    4  501   13    0    4    1 |    f = 5
  10    1    9    0    3    4  637    0    0    0 |    g = 6
   1    0    1    1    7    0    0  624    2    9 |    h = 7
   6    5   13    7    4   10    0    4  492    1 |    i = 8
```

```
   0    1    0    3    6    1    0   16    0  617 |    j = 9
```

2. LOGISTIC

=== Confusion Matrix ===

```
   a    b    c    d    e    f    g    h    i    j   <-- classified as
1126    3   10    6    9   16   11    6    6    1 |    a = 0
   0  991    3    0    3    0    1    5    2    0 |    b = 1
  12   13  584   17   29    5   14   25   22   10 |    c = 2
   4    2   15  554    2   36    4   16   19    6 |    d = 3
   7    7   14    4  549   12   15    8   17   19 |    e = 4
  13    5    4   19   10  455   24    6   11    9 |    f = 5
  12    4    8    0   10   13  613    2    2    0 |    g = 6
   1    7    4    3    7    2    2  585    8   26 |    h = 7
  12   15   17   14    9   15    3    7  433   17 |    i = 8
   3    2    0    3   26    8    1   23    8  570 |    j = 9
```

3. SIMPLE LOGISTIC

=== Confusion Matrix ===

```
   a    b    c    d    e    f    g    h    i    j   <-- classified as
1174    0    2    3    4    5    4    0    2    0 |    a = 0
   0  997    0    0    0    4    0    1    3    0 |    b = 1
   2    1  680   12   13    1    5    7    9    1 |    c = 2
   3    0   10  612    2   13    0    3   10    5 |    d = 3
   3    4    6    0  607    5   10    4    4    9 |    e = 4
  11    0    4   17    8  496   11    0    4    5 |    f = 5
   7    2    5    0    8    6  631    1    4    0 |    g = 6
   0    0    1    0    8    1    0  618    3   14 |    h = 7
   7    1    8   11   12   12    3    5  482    1 |    i = 8
   1    4    0    2    8    0    0   18    4  607 |    j = 9
```

4. NAIVE BAYES

=== Confusion Matrix ===

```
a    b    c    d    e    f    g    h    i    j   <-- classified as
1018    0   10    6    7    9   16    1  126    1 |    a = 0
   0  989    2    0    0    0    1    0    6    7 |    b = 1
   5    1  591   19    9   10   27    3   65    1 |    c = 2
   6    1   30  439    4   19    3    9  126   21 |    d = 3
   0    6    6    0  218   18   15    1    9  379 |    e = 4
  19    0   12   26   12  294  104    3   72   14 |    f = 5
   6    5   20    0    1   14  609    0    9    0 |    g = 6
   0    4    3    3    9    0    0  576    3   47 |    h = 7
   2    9   14    3    4   20    2    4  395   89 |    i = 8
   0    9    0    0   30    4    0   21   10  570 |    j = 9
```

5. MULTILAYER PERCEPTION

=== Confusion Matrix ===

a	b	c	d	e	f	g	h	i	j	<-- classified as
1181	0	2	2	2	4	3	0	0	0	a = 0
0	1001	0	0	0	2	0	1	1	0	b = 1
4	1	704	2	6	2	3	3	5	1	c = 2
1	0	3	633	1	9	0	3	7	1	d = 3
3	2	3	0	629	1	8	0	0	6	e = 4
7	0	3	5	2	529	5	0	3	2	f = 5
3	1	2	0	5	2	649	0	2	0	g = 6
0	0	2	0	2	1	0	630	3	7	h = 7
3	3	2	3	4	4	1	3	519	0	i = 8
1	0	1	2	5	1	0	7	0	627	j = 9

****