<pre>import num import mat import mat from sklea from sklea from sklea from sklea</pre>	Odal.ca
<pre>from sklea</pre>	born as sns
from sklea import skf import ten from tenso from tenso from tenso from tenso [1] Prepar Explain year	rn.decomposition import PCA uzzy as fuzz sorflow as tf rflow.keras.models import Sequential rflow.keras.utils import to_categorical rflow.keras.layers import Conv2D, MaxPooling2D, Flatten, Dense rflow.keras.optimizers import Adam re the dataset by applying data transformation. Which method of transformation did you choose? cour rationality behind it.
<pre>testing_da training_d Out[2]:</pre>	ata = pd.read_csv('data/sign_mnist_train.csv') ta = pd.read_csv('data/sign_mnist_test.csv') ata.head() pixel1 pixel2 pixel3 pixel4 pixel5 pixel6 pixel7 pixel8 pixel9 pixel775 pixel776 pixel777 pixel778 pixel7 107
training_d	164 167 170 172 176 179 180 184 185 92 105 105 108 5 columns b[1;31m'+'Check for NaN values in data :'+'\x1b[0m') ata.isna().sum() laN values in data : 0 0
pixel2 pixel3 pixel4 pixel780 pixel781 pixel782 pixel783 pixel784 Length: 785	0 0 0 0 0 0 0 0 0 0 0 0 0 0 type: int64 b[1;31m'+'Total Records :'+'\x1b[0m', len(training_data))
print('\x1 Total Record Duplicate F In [5]: label_coun label_coun label_coun print(labe	b[1;31m'+'Duplicate Records :'+'\x1b[0m', len(training_data[training_data.duplicated()])) rds : 27455 Records : 0 ts = training_data['label'].value_counts().reset_index().sort_values(by='label') ts.reset_index(drop=True, inplace=True) ts['Alphabet'] = label_counts['label'].apply(lambda x: chr(x + ord('A'))) ts.columns = ['Labels', 'Count', 'Alphabet']
<pre>sns.barplo plt.title(plt.show()</pre>	<pre>(figsize=(10,8)) t(data=label_counts, x="Alphabet", y="Count") 'Frequency of Alphabets/Labels in Training data') Count Alphabet 1126</pre>
7 7 8 8 9 10 10 11 11 12 12 13 13 14 15 15 16 16 17 17 18 18 19 19 20	1013 H 1162 I 1114 K 1241 L 1055 M 1151 N 1196 O 1088 P 1279 Q 1294 R 1199 S 1186 T 1161 U
20 21 21 22 22 23 23 24 Visual Anal	1082 V 1225 W 1164 X 1118 Y Lysis of data training data: Frequency of Alphabets/Labels in Training data
1200 - 1000 -	
Count Count - 008	
200 - 0 -	A B C D E F G H I K L M N O P Q R S T U V W X Y Alphabet
<pre># Set up t plt.figure # Iterate for i, lab if i > i # Find</pre>	raining_data.drop('label', axis=1).values / 255.0 he figure for displaying images (figsize=(12, 12)) through labels to display corresponding images el in enumerate(label_counts['Alphabet']): = 9: = i+1 the index of the first occurrence of the current label training_data[training_data['label'] == i].index[0]
<pre># Crea plt.su plt.ti plt.im plt.ax</pre>	act the image corresponding to the label images[idx].reshape(28, 28) te subplots to display images bplot(4, 7, i + 1) tle(label) show(img, cmap='gray') is('off') layout() # Adjust layout for better spacing B C D E F G
-	
8	
V	
<pre>print('\n\ scaler = M training_d</pre>	ng the pixel values to range from 0 to 1, aiding in faster convergence during model training n\x1b[1;31m'+'Normalizing the pixel values to range from 0 to 1:'+'\x1b[0m\n\n') inMaxScaler() ata.iloc[:, 1:] = scaler.fit_transform(training_data.iloc[:, 1:])
training_d Normalizing Out[77]: label 0 3 0	ta.iloc[:, 1:] = scaler.fit_transform(testing_data.iloc[:,1:]) ata.head() g the pixel values to range from 0 to 1 : pixel1 pixel2 pixel3 pixel4 pixel5 pixel6 pixel7 pixel8 pixel9 pixel775 pixel776 pixel 0.419608 0.462745 0.498039 0.525490 0.545098 0.560784 0.572549 0.588235 0.600000 0.811765 0.811765 0.811765 0.607843 0.615686 0.611765 0.611765 0.611765 0.615686 0.611765 0.619608 0.619608 0.270588 0.584314 0.500
3 2 0 4 13 0 5 rows × 78! Reason	0.733333 0.737255 0.737255 0.733333 0.733333 0.729412 0.733333 0.737255 0.733333 0.792157 0.788235 0.7840 0.827451 0.827451 0.831373 0.831373 0.827451 0.823529 0.827451 0.823529 0.823529 0.921569 0.917647 0.9130 0.643137 0.654902 0.666667 0.674510 0.690196 0.701961 0.705882 0.721569 0.725490 0.360784 0.411765 0.4130 0.654902 0.666667 0.674510 0.690196 0.701961 0.705882 0.721569 0.725490 0.360784 0.411765 0.4130 0.654902 0.666667 0.674510 0.690196 0.701961 0.705882 0.721569 0.725490 0.360784 0.411765 0.4130 0.654902 0.666667 0.674510 0.690196 0.701961 0.705882 0.721569 0.725490 0.360784 0.411765 0.41300 0.6643137 0.654902 0.666667 0.674510 0.690196 0.701961 0.705882 0.721569 0.725490 0.360784 0.411765 0.41300 0.666667 0.674510 0.690196 0.701961 0.705882 0.721569 0.725490 0.360784 0.411765 0.41300 0.666667 0.674510 0.690196 0.701961 0.705882 0.721569 0.725490 0.360784 0.411765 0.41300 0.666667 0.674510 0.690196 0.701961 0.705882 0.721569 0.725490 0.360784 0.411765 0.41300 0.666667 0.674510 0.690196 0.701961 0.705882 0.721569 0.725490 0.360784 0.411765 0.41300 0.666667 0.674510 0.690196 0.701961 0.705882 0.721569 0.725490 0.360784 0.411765 0.41300 0.666667 0.674510 0.690196 0.701961 0.705882 0.721569 0.725490 0.360784 0.411765 0.41300 0.666667 0.674510 0.690196 0.701961 0.705882 0.721569 0.725490 0.360784 0.411765 0.41300 0.666667 0.674510 0.690196 0.701961 0.705882 0.721569 0.725490 0.725490 0.360784 0.411765 0.41300 0.701961 0.7
 Ne No 2. Faster No Th 3. Gradie State co 	eural networks often perform better when input features are within a similar range. commalizing pixel values between 0 and 1 helps in creating a consistent scale for the input features. Convergence: commalization assists in stabilizing the training process by preventing overly large or small input values. is aids in faster convergence during gradient descent optimization while training the model. Int Descent Efficiency: andardizing the range of input features helps in avoiding potential issues such as vanishing or exploding gradients, which uld hinder the optimization process during training.
4. Improve • Note that the set up to the se	ormalization can contribute to better generalization by preventing the dominance of certain features due to their larger agnitude compared to others. In \x1b[1;31m'+'Visual Analysis of data training data after Normalization :'+'\x1b[0m\n\n') The figure for displaying images (figsize=(12, 12)) Through Labels to display corresponding images
<pre>for i, lab if i > i # Find idx = # Extr img = # Crea plt.su plt.ti</pre>	<pre>el in enumerate(label_counts['Alphabet']): = 9: = i+1 the index of the first occurrence of the current label training_data[training_data['label'] == i].index[0] act the image corresponding to the label images[idx].reshape(28, 28) te subplots to display images bplot(4, 7, i + 1) tle(label)</pre>
plt.ax plt.tight_ plt.show()	show(img, cmap='gray') is('off') layout() Lysis of data training data after Normalization : B C D E F G
Н	
V	
<pre>y_train = X_test = t</pre>	training_data.drop('label', axis=1).values training_data['label'].values esting_data.drop('label', axis=1).values esting_data['label'].values
<pre># Split th X_train, X pca = PCA(X_train, X I have used setting n_cc</pre>	e data into training and validation sets (80% training, 20% testing) _val, y_train, y_val = train_test_split(X_train, y_train, test_size=0.2) n_components = 2) _test, X_val = pca.fit_transform(X_train), pca.fit_transform(X_test), pca.fit_transform(X_val) PCA to reduce the number of features (or dimensions) while retaining most of the important information. In this case, by emponents to 2, PCA is reducing the high-dimensional data to a two-dimensional space. This reduction is beneficial for
[2] Apply a. Change algorithm In [26]:	the k-means algorithm to Sign Language MNIST dataset. It the number of clusters from 10 to 200 with the step size of 10. Show the performance of the based on accuracy and the objective function value for each cluster number
<pre>inertia_li silhouette # Range of for i in r kmeans kmeans</pre>	<pre>ist_kmeans = [] st_kmeans = [] _avg_kmeans = [] clusters from 10 to 200 with step size 10 ange(10, 201, 10):</pre>
<pre>inerti # Appe accura inerti silhou In [27]: j = 0 for i in r</pre>	<pre>cy = accuracy_score(y_val, y_pred) a = kmeans.inertia_ nd accuracy and inertia to the lists cy_list_kmeans.append(accuracy) a_list_kmeans.append(inertia) ette_avg_kmeans.append(silhouette_score(X_val, y_pred)) ange(10, 201, 10):</pre>
<pre>print(print(print(print(j+=1 Clusters : Accuracy : Objective F</pre>	<pre>'\n\x1b[1;31m'+'Clusters :'+'\x1b[0m', i) '\x1b[1;31m'+'Accuracy :'+'\x1b[0m', accuracy_list_kmeans[j]*100, '%') '\x1b[1;31m'+'Objective Function :'+'\x1b[0m', inertia_list_kmeans[j]) '\x1b[1;31m'+'Silhoutte Average :'+'\x1b[0m', silhouette_avg_kmeans[j])</pre>
Objective F Silhoutte F Clusters: Accuracy: Objective F Silhoutte F Clusters: Accuracy:	5.026406847568749 % Function: 15852.660861665832 30 3.64232380258605 % Function: 10766.456262085001 Average: 0.31561292562607624
Clusters : Accuracy : Objective F Silhoutte / Clusters : Accuracy : Objective F	2.1671826625387 % Function: 6528.9708099390555 Average: 0.31807936517529123 60 1.2930249499180477 % Function: 5514.246403473899 Average: 0.32104072578113174
Objective F Silhoutte F Clusters: Accuracy: Objective F Silhoutte F Clusters: Accuracy: Objective F	1.2019668548533966 % Sunction: 4134.935486018572 Average: 0.3203439862576019
Clusters : Accuracy : Objective F Silhoutte / Clusters : Accuracy : Objective F	100 1.0380622837370241 % Function: 3295.2780113445415 Everage: 0.32153563882277497 110 0.9652158076853032 % Function: 2986.264214924753 Everage: 0.3175819859016003
Objective F Silhoutte F Clusters: Accuracy: Objective F Silhoutte F Clusters: Accuracy: Objective F	1.0016390457111637 % Function: 2564.0212323312358 Average: 0.31135576793382463 140 0.874157712620652 % Function: 2356.2113417410624
Clusters : Accuracy : Objective F Silhoutte F Clusters : Accuracy : Objective F Silhoutte F	0.47350209433618645 % Function: 2187.189003559469 Average: 0.3155046440913891 160 0.5463485703879074 % Function: 2048.4871403782754 Average: 0.31956330240023356
Objective F Silhoutte F Clusters: Accuracy: Objective F Silhoutte F Clusters: Accuracy:	0.509925332362047 % Function: 1926.7904113543414 Average: 0.3191625593266038 180 0.49171371334911673 % Function: 1816.9048499887808 Average: 0.31505814543116906
Clusters: Accuracy: Objective F Silhoutte F b. What is In [28]: print('\n\	200 0.291385904206884 % Function: 1627.5496091272528 Everage: 0.31706015481453703
<pre>plt.xlabel plt.ylabel plt.title(plt.show()</pre>	s the optimal number of clusters? Justify your answer. n\x1b[1;31m'+'Silhoutte Analysis For Optimal k :'+'\x1b[@m\n\n')
0.322 0.320 0.318	s the optimal number of clusters? Justify your answer.
0.314 0.312	s the optimal number of clusters? Justify your answer. n\x1b[1;31m'+'Silhoutte Analysis For Optimal k :'+'\x1b[0m\n\n') ange(10, 201, 10), silhouette_avg_kmeans, marker='o') ('Values of k') ('Silhoutte Score') 'Silhoutte Analysis For Optimal k') analysis For Optimal k :
<pre>plt.plot(r plt.xlabel plt.ylabel</pre>	s the optimal number of clusters? Justify your answer. n\x1b[1;31m'+'Silhoutte Analysis For Optimal k :'+'\x1b[0m\n\n') ange(10, 201, 10), silhouette_avg_kmeans, marker='o') ('Values of k') ('Silhoutte Score') 'Silhoutte Analysis For Optimal k') analysis For Optimal k :
<pre>plt.title(plt.show()</pre>	s the optimal number of clusters? Justify your answer. N\xib[1;31m'+'Silhoutte Analysis For Optimal k:'+'\xib[@m\n\n') ange(10, 201, 10), silhoutte_avg_kmeans, marker='o') ('Values of k') ('Silhoutte Score') ('Silhoutte Analysis For Optimal k') Silhoutte Analysis For Optimal k Silhoutte
plt.title(plt.show()) Elbow Method 30000	s the optimal number of clusters? Justify your answer. Invals[1;31m*'Silhoutte Analysis For Optimal k:'*'\x1b[@m\n\n') mage(18, 281, 18), 5ilhoutte_avg_kmeans, earker='o') (Values of k') (Silhoutte Analysis For Optimal k') Silhoutte Analysis For Optimal k Silhoutte Analysis For Optimal k Silhoutte Analysis For Optimal k **Type Toptimal k :** Type Toptimal
plt.title(plt.show()) Elbow Methor 30000	sthe optimal number of clusters? Justify your answer. Analysis For Optimal
plt.title(plt.show()) Elbow Method 30000 25000 25000 10000 5000 In [92]: num_cluste kmeans = K kmeans.fit y_pred_kme	s the optimal number of clusters? Justify your answer. Invisit jum - Silboutte Analysis for Optimal k : "+"vxtel@mvnn") Insilboutte Analysis for Optimal k') Silboutte Analysis for Optimal k': Silboutte Analysis for Optimal k Silboutte Anal
plt.title(plt.show()) Elbow Method 30000 25000 25000 20000 In [92]: num_clusted kmeans = K kmeans.fit y_pred_kmed # Calculated accuracy = inertia = # Append a print('\n\ print('\x1 print('\x1 print('\x1 clusters: Accuracy:	st the optimal number of clusters? Justify your answer. (vish); 3in 41st hourte Asalysis Son (petral & 114 / vish(petrus)) (visians of \$\frac{1}{2}\$) (visin of spance distrance/facetiae) (visin of spance distrance/facetiae) (visin of spance distrance/facetiae) (visin of spance distrance/facetiae) (visians of spance distrance/facetiae) (visin of spance
plt.title(plt.show()) Elbow Method 30000 25000 25000 25000 15000 In [92]: num_cluste kmeans.fit y_pred_kme # Calculat accuracy = inertia = # Append a print('\n\print('\x1 print('\x1 print	st the optimal number of clusters? Justify your answer. Problems of St. 190, 190, 1910 stillouting analysis for destinal k 11 (Mainteen St. 190, 190), 1910 stillouting analysis for destinal k 11 (Mainteen St. 190, 190), 1910 stillouting analysis for Optimal k 1 Silhoutte Analysis
plt.title(plt.show()) Elbow Method 30000 25000 25000 25000 15000 5000 In [92]: num_cluste kmeans = K kmeans.fit y_pred_kme # Calculat accuracy = inertia = # Append aprint('\n\print('\x1) print('\x1)	st the optimal number of clusters? Austify your answer. **PARTITION OF THE CONTROL OF THE CONTR
In [92]: num_cluste kmeans = K kmeans.fit y_pred_kme # Calculat accuracy = inertia = # Append a print('\n\print('\x1 print('\x1	subscriptional number of clusters? Justify your answer. Applicant in Subscription or applied in 11 Subscription (Clusters) Medical (Lab., Subscription or applied in 11 Subscription (Clusters) S
In [92]: num_clustekmeans = Kkmeans.fit y_pred_kme # Calculat accuracy = inertia = # Append a print('\n\print('\x1 print('\x1 print(st the optimal number of clasters? Justify your answer. Including 1970 (1986)
plt.title(plt.show()) Elbow Method 30000 25000 25000 25000 15000 10000 In [92]: num_cluste kmeans = K kmeans.fit y_pred_kme # Calculat accuracy = inertia = # # Append a print('\n\print('\x1 print('\x1 print('	And the optimal number of clusters? Audity year amoves. Site of place of the clusters are proposed as a second of clusters and a second of cluste
plt.title(plt.show()) Elbow Method 30000 Elbow Method 30000 250000 250000 Soloo Soloo Soloo Soloo Soloo In [92]: num_cluste kmeans = K kmeans = K kmeans = K kmeans = K the Calculate accuracy = inertia = # # Append a print('\n\print('\x1) print('\x1) pr	the optimal number of clusters / Justing your amoves. another in Charles Analysis on Ottaria is a registrative of the cluster
plt.title(plt.show()) Elbow Method 30000 in 25000 25000 25000 25000 15000 15000 100000 100000 100000 100000 100000 100000 10000	the optimal number of clusters 2 Justiny your amover. Interesting of Schools Analysis of Analysis of Analysis of Analysis (1964) Colons of 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
plt.title(plt.show() Elbow Method 30000 25000 25000 25000 25000 25000 In [92]: num_cluste kmeans.fit y_pred_kme # Calculat accuracy = inertia = # Append a print('\x1 print('\x	the applical number of dustant Justify your answer. The application and property is a friendly provided by the friendly by the friendly provided by the friendly b
plt.title(plt.show()) Elbow Method 30000 in [92]: num_cluste kmeans = K kmeans.fit y_pred_kme # Calculat accuracy = inertia = # # Append a print('\x') pri	The continued number of columns / Junify your entown. Secretary 10 - 20 - 20 - 20 - 20 - 20 - 20 - 20 -
plt.title(plt.show() 30000 Source plt.show() 30000 Source plt.show() 10000 In [92]: num_cluste kmeans = K kmeans.fit y_pred_kme # Calculat accuracy = inertia = # # Append a print('\x1 plt.scat plt.subplo colors = pfor i,c in plt.scat plt.tigle(plt.xlabel plt.tishe(plt.xlabel plt.tigle(plt.xlabel plt.xlabel plt.tigle(plt.xlabel plt.xlab	Similar Control number of claracy 1 Audity year amove. Similar Control Name Analysis for Optimal R Similar Co
plt.title(plt.stitle(plt.show()) Elbow Method 30000 Elbow Method 40000 Elbow Method 4	The optional number of discrete (butter) year around: Similarly 1, 15 - Similarly
plt.title(plt.stitle(plt.show()) Elbow Method 30000 Elbow Method 30000 Elbow Method 30000 Soloto 10000 In [92]: In getting In g	Since the equivalent number of closery of boding your money. Since the special content of
plt-title(plt-show()) Flbow Method 30000 Flbow Method 40000 Flbow M	Silved Analysis of Column Column State Colum
The most st a couracy = in pit. state of pit	Service of the control and the
The most sc a slight elegative sc in plt. scard plt. state sc in print scard plt. state sc in print sc in print sc in print sc in print scard plt. state sc in print sc in pri	And the first and the continue of the continue
plt.title(plt.show()) Elbow Method 30000 Elbow Method 40000 Elbow	And the continued and of a state
plt.title(plt.show()) 30000 in [92]: plt.title(plt.show()) 30000 in [92]: plt.title(plt.show()) 5000 in [92]: num_cluste kmeans = K km	Silmurate Analysis for Optional K Silmurate Analys

	25	50	75 100 Values (125 of k	150 175 200
<pre>In [64]: # Initialize li accuracy_list = objective_funct silhouette_avg num_clusters=20 # Varying the f # The value can for m_value in</pre>	sts to sto [] ion_list = = [] fuzzifier v not be 1 a [1.1, 2, 3 eans clust d, jm, p .T, num_cl	re perform [] alue from s it throw, 4, 5]: ering with, fpc = fu usters, m=	ance metrics 1.1 to 5 s DivideByZer varying fuzz zz.cluster.cm _value, erro	ro error fo zifier valu means(or=0.005, m	ccuracy and the objective function value by changing or starting from 1.1 in this case ue maxiter=1000, init=None
<pre>u_test, u0_</pre>	test, d_te , cntr, m= .argmax(u_ ng accurace accuracy_s st.append(function_li e score	st, jm_tes m_value, e test, axis y core(y_val accuracy) value st.append(rror=0.005, m =0) , y_pred)	oc_test = 1	
for i in [1.1, print('\n\x print('\x1b print('\x1b print('\x1b j+=1 Fuzzier Value : Accuracy : 2.699 Objective Function Fuzzier Value : Accuracy : 4.261 Objective Function Fuzzier Value : Collinoutte Average Fuzzier Value : Fuzzier Value :	1b[1;31m'+'A [1;31m'+'A [1;31m'+'O [1;31m'+'S 1.1 53196139136 ion: 15881 ge: 0.3189 2 15188490256 ion: 5729 ge: 0.2952	'Fuzzier V ccuracy :' bjective F ilhoutte A 577 % 1.284565646 98606997622	unction :'+'\ verage :'+'\x 0146 2175	accuracy_li x1b[0m', o	<pre>ist[j]*100, '%') objective_function_list[j]) ilhouette_avg[j])</pre>
Accuracy: 4.552 Objective Functi Silhoutte Average Fuzzier Value: Accuracy: 2.956 Objective Functi Silhoutte Average Fuzzier Value: Accuracy: 3.715 Objective Functi Silhoutte Average In [90]: num_clusters=20 m_value=3 # Fuzzy K-means	don: 452.6 ge: 0.1964 4 02822800947 don: 25.34 ge: 0.1177 5 51702786377 don: 1.326 ge: 0.0507	52804182381 11514614316 7004 % 12097591595 72764484877 7707 % 52874835196 70670601849	5962 7916 5343 9837		
<pre># Predicting cl u_test, u0_test X_test.T, c) y_pred_fcm = np # Calculating a accuracy = accu print('\x1b[1;3 print('\x1b[1;3</pre>	<pre>uster Labe , d_test, ntr, m_val .argmax(u_ ccuracy racy_score 1m'+'Accur 1m'+'Objec 1m'+'Silho 03290574456 ion : 448.5</pre>	<pre>ls for the jm_test, p ue, error= test, axis (y_test, y acy :'+'\x tive Funct utte Avera 5215 % 38083682407</pre>	<pre>validation s _test, fpc_te 0.005, maxite =0) _pred_fcm) 1b[0m', accur ion :'+'\x1b[0] ge :'+'\x1b[0]</pre>	est = fuzz er=1000	
<pre>plt.legend(font plt.title('Befo plt.xlabel('X-a plt.ylabel('Y-a # Plot after cl plt.subplot(1, colors = plt.cm for i,c in enum plt.scatter(X</pre>	<pre>lustering 2, 1) .rainbow(n erate(colo _test[y_te size=5, ti re K-means xis') xis') ustering 2, 2) .rainbow(n erate(colo _test[y_pr</pre>	<pre>p.linspace rs): st==i, 0], tle_fontsi Clusterin p.linspace rs): ed_fcm==i,</pre>	<pre>X_test[y_tes ze=5) g') (0, 1, 20)) 0], X_test[y</pre>		<pre>marker='o',color = c, label=i, alpha =0.7) ==i, 1], marker='o',color = c, label=i, alpha =0.7)</pre>
plt.legend(font plt.title('Afte plt.xlabel('X-a plt.ylabel('Y-a plt.tight_layou plt.show()	r K-means xis') xis') t()	_ Clustering	•	0 0 1 2 3 3 4 4 5 5 7 8 9 9 10 11 12 12 12 12 13 14 15 16 16 17 18 18 18 18 18 18 18 18 18 18 18 18 18	2 - 10 10 10 10 10 10 10 10 10 10 10 10 10
	1m'+'Resul display tonfusion_m n Matrix:"	X-axis d FCM ba ts for K-m he confusi atrix(y_te)	sed on the reans :'+'\x1b	D[0m')	-2 -4 -4 -7.5 -5.0 -2.5 0.0 2.5 5.0 7.5 10.0 X-axis
<pre>sns.heatmap(conf_matrix</pre>	<pre>, annot = ap=sns.cub display a racy_score y: {accura display p cision_sco _score(y_t _test, y_p on: {preci {recall:.</pre>	True, ehelix_pal ccuracy (y_test, y cy:.2f}") recision, re(y_test, est, y_pre red_kmeans sion:.2f}"	recall, and F y_pred_kmean d_kmeans, ave , average='mi	F1 score ns, average erage='micr	
0 -0 0 9 5 0 0 0 8 5 2 11 5 18 0 2 1 1 1 1 1 1 1 1 2 5 5 7 2 5 6 1 4 2 5 5 7 2 5 6 1 4 2 1 1 1 1 1 1 1 2 1 1 1 1 1 2 1 1 1 1	cation Repation_reports	ort:") rt(y_test, 01589008 10155214624 3 0180023 62 2471114 02 1410 038 0 7890 014 1012458293 10166274314 0161441015 0 0 0 0 0 0 0 1252160 036 0 0211506 129426 718 010 82 3 2 0 4170 023 5947023630	y_pred_kmean 10000000 100000000 1700000000000000000	- 100 - 80 - 60	
87 -2 12014 831 4 014 6130 07 -4 03 145 810 37 43 112 911 07 -4 113 86208 2 1101413 711 14 -4 5511 6523 0 2 4 Accuracy: 0.04 Precision: 0.04 Recall: 0.04 F1 Score: 0.04 Classification F	0 924 144 3221918312 15171416011 2 4 5161 0 2 1510512 11816111010 6 8 10 Report: recision 0.09 0.03 0.02 0.03	recall f	210 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	pport 331 432 310 245	
4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21	0.04 0.07 0.07 0.03 0.00 0.01 0.10 0.01 0.02 0.02 0.18 0.03 0.01 0.05 0.05 0.00 0.00 0.00	0.04 0.10 0.06 0.02 0.05 0.00 0.01 0.10 0.00 0.03 0.07 0.07 0.02 0.03 0.09 0.04 0.00 0.00	0.04 0.08 0.06 0.02 0.03 0.00 0.01 0.10 0.00 0.03 0.04 0.10 0.03 0.02 0.06 0.05 0.00 0.00 0.00	498 247 348 436 288 0 331 209 394 291 246 347 164 144 246 248 266 346 206	
23 24 accuracy macro avg weighted avg In [166]: print('\x1b[1;3 # Calculate and conf_matrix = c print("Confusio plt.figure(figs sns.heatmap(0.00 0.00 0.04 0.04 1m'+'Resul display tonfusion_m n Matrix:" ize=(5,4))	0.00 0.00 0.04 0.04 ts for FCM he confusi atrix(y_te)	0.00 0.00 0.04 0.03 0.04 :'+'\x1b[0m'	267 332 7172 7172 7172	
<pre># Calculate and accuracy = accu print(f"Accurac # Calculate and precision = pre</pre>	racy_score y: {accura display p cision_sco _score(y_t _test, y_p on: {preci {recall:.e: {f1:.2f lassificat cation Rep	<pre>(y_test, y cy:.2f}") recision, re(y_test, est, y_pre red_fcm, a sion:.2f}" 2f}") }") ion report ort:")</pre>	recall, and F y_pred_fcm, d_fcm, averag verage='weigh)	average='vge='vge='vge='vge='veighten nted', zero	
7 -0370 1440 9 7235 814 4 -74511231 224152517 9 -1871162763 21826432-02 8 -1167 9721 0 0 0 0 0 0 0 -524631315 1100 41703 21 1681918124 417140 0 6 91 -305 012168 8 -144215123 7150 45141 0 -6 5 415128 10132251883	0:2992123892491629017 130470270 130470270 126 228622 169524 266 138239142 1551715112 2262281313 103 434114 00 0 0 0 0 0 0 928234622 130241210 202682215 1181520145 13052 0224 73706 055 324314019 4 0 5 8132 1361120820 160181211 180144190 1891528915	365 52 162 84 2476 036 0 24 9151912 2072 10810 10842 2330 7155 148 58 1 12 22 32 86 1 7165 1918 48 0 0 0 0 0 0 25 349 8100 81871 104 10 22 78 1925 21 100 183 05 725 1 04 1 0 10 0 16 3 181 01 28 14 0	205 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	- 140 - 120 - 100 - 80 - 60 - 40 - 20	
2 91451125 2 23827112084 0 2 4 Accuracy: 0.02 Precision: 0.03 Recall: 0.02 F1 Score: 0.02 Classification F	834112898 141510121617 6 8 10	30107 31518 2110013132:	32180 0 0 0 0 18140 0 0 0 0 18 20 22 24	- 0	
8 9 10 11 12 13 14 15 16 17 18 19 20 21 22	0.01 0.00 0.02 0.00 0.06 0.02 0.00 0.00 0.00	0.01 0.00 0.02 0.00 0.06 0.03 0.00 0.00 0.00 0.02 0.02 0.02 0.02	0.01 0.00 0.02 0.00 0.06 0.02 0.00 0.00 0.00	288 0 331 209 394 291 246 347 164 144 246 248 266 346	
23 24	0.00 0.00	0.00 0.00	0.00	206 267 332	
accuracy macro avg weighted avg Comparisor Accuracy: K- samples. Precision, Re Precision ind proportion of low scores su Analysis:	0.00 0.02 0.03 T: -means outple call, and Ficates the part of actual postuggest poor	0.00 0.02 0.02 0.02 0.fore: Boroportion or itive instance performance	0.00 0.02 0.02 0.02 M in terms of a th K-means and f correctly ident tes that were co	267 332 7172 7172 7172 d FCM exhil tified instar orrectly ider true positiv	dicating that K-means achieved a higher ratio of correctly classified bit similar performance across these metrics, with very low scores. Inces among the instances predicted as positive; Recall measures the natified; F1 Score is the harmonic mean of Precision and Recall. The res and minimizing false positives/negatives for both algorithms.
accuracy macro avg weighted avg Comparisor Accuracy: K- samples. Precision, Re Precision ind proportion of low scores so Analysis: Interpretation correctly class with the grow Limitations: algorithms for Conclusion:	0.00 0.02 0.03 1: -means outper secall, and Finance in post poor in the given in	0.00 0.02 0.02 0.02 0.02 1 Score: Bo roportion or itive instance reperformance conditions and the conditions are performance of data distributions.	0.00 0.02 0.02 M in terms of a th K-means and f correctly ident tes that were co te in capturing values of Precis into distinct clu could be due to oution, or inade	267 332 7172 7172 7172 accuracy, incomo def FCM exhilatified instart prectly ider true positives ion, Recall, justers or classion, accuracy factorious fac	bit similar performance across these metrics, with very low scores. nces among the instances predicted as positive; Recall measures the ntified; F1 Score is the harmonic mean of Precision and Recall. The
accuracy macro avg weighted avg Comparisor Accuracy: K- samples. Precision, Re Precision ind proportion of low scores so Analysis: Interpretation correctly class with the grow Limitations: algorithms for Conclusion: Both K-meat low performation using the train Sign-MNIST d In [138]: def fix_label_g if(1>=9): return else: return X_train = train y_train = to_ca X_test = testin y_test = testin	0.00 0.02 0.03 1: -means outper conditions and FCN conditions and FCN conditions and FCN conditions are across and acceptance across and test ataset, the conditions and test ataset, the conditions are across are acros	0.00 0.02 0.02 0.03 0	0.00 0.02 0.02 0.02 M in terms of a th K-means and forectly identes that were concern to a the control of t	267 332 7172 7172 7172 7172 accuracy, incomo def CM exhilitified instant prectly identified instant prectly identified instant prectly identified in effective for and train pedition of the color of th	bit similar performance across these metrics, with very low scores. Incres among the instances predicted as positive; Recall measures the notified; F1 Score is the harmonic mean of Precision and Recall. The res and minimizing false positives/negatives for both algorithms. and F1 Score indicate that both algorithms might struggle in sees. The clusters formed by both algorithms might not align well stors such as noisy data, inappropriate choice of clustering processing of data before clustering. By capturing the underlying structure in the data, as reflected by the at the network. This network will be trained and tested by the instance into 1 of 24 classes.
accuracy macro avg weighted avg Comparisor Accuracy: K- samples. Precision, Re Precision ind proportion of low scores st Analysis: Interpretation correctly class with the grow Limitations: algorithms for Conclusion: Both K-meat low performs [4] Implement using the train Sign-MNIST d In [138]: def fix_label_g if(1>=9): return else: return x_train = train y_train = to_cat # Reshape the d X_train = train y_train = to_cat # Reshape the d X_train = X_train	o.00 o.02 o.03 i: means output ecall, and F icates the p f actual post aggest poor on of Result sifying the output truth late The low perior the given ms and FCN ance across a feedfor and test ataset, th ap(1): (1-1) ling_data_dro g_data_['lategorical(y ata (assum in_reshape(- imple Con pation fun architectu ial() Layers O(32, kern o(1)(64, kern	0.00 0.02 0.02 0.03 0	0.00 0.02 0.02 0.02 M in terms of a th K-means and forectly identes that were concerned in capturing values of Precision distinct clusted and the concerned in capturing values of Precision distinct clusted and the concerned in capturing values of Precision distinct clusted in capturing values of Precision distinct clusted in capturing value in capturi	267 332 7172 7172 7172 7172 accuracy, income defend the state of the s	bit similar performance across these metrics, with very low scores, access among the instances predicted as positive; Recall measures the titified; F1 Score is the harmonic mean of Precision and Recall. The res and minimizing false positives/negatives for both algorithms. and F1 Score indicate that both algorithms might struggle in sses. The clusters formed by both algorithms might not align well stors such as noisy data, inappropriate choice of clustering rocessing of data before clustering. by capturing the underlying structure in the data, as reflected by the another network. This network will be trained and tested and the network. This network will be trained and tested and input image (28 x 28 = 784 pixels) from the assify the image into 1 of 24 classes. The maximum 10 hidden layers composed of sign and build your model. Specify kernel sizes, number ion, and loss functions of your model.
accuracy macro avg weighted avg Comparisor Accuracy: Kasamples. Precision, Rance Precision inderity proportion of low scores states and proportion of low performs and	o.00 o.02 o.03 ceall, and Ficates the properties of actual postuggest poor and truth late and	o.00 o.02 o.02 o.02 oerforms FC 1 Score: Bo roportion or itive instance performance ts: The low of data points bels. rformance of data distrib I seem to ha multiple me rward net sets of th e network rop('label' label'].appl _test, num ing grayso (-1, 28, 28, 1, 28, 28, 1, 28, 28, re el_size=(3, 2))) _tion from vation='re ation='sof dam(learni orical_cro ccuracy']) in, y_trai test) (np.argmax	0.02 0.02 0.02 0.02 M in terms of a th K-means and forrectly identes that were contained by the contained b	7172 7172 7172 7172 7172 Accuracy, income of FCM exhibitified instant prectly identified instant prectly identified instant prectly identified in effective of and train pecifically in effective of and train pecifically in effective of an exhibition of the company of the comp	bit similar performance across these metrics, with very low scores, coes among the instances predicted as positive; Recall measures the ntified; F1 Score is the harmonic mean of Precision and Recall. The less and minimizing false positives/negatives for both algorithms. and F1 Score indicate that both algorithms might struggle in sses. The clusters formed by both algorithms might not align well stors such as noisy data, inappropriate choice of clustering rocessing of data before clustering. by capturing the underlying structure in the data, as reflected by the control of the network. This network will be trained and tested the properties of the sisting the image into 1 of 24 classes. by the image into 1 of 24 classes. ch maximum 10 hidden layers composed of sign and build your model. Specify kernel sizes, number ion, and loss functions of your model. ch, input_shape=(28, 28, 1))) connected Layers with 24 classes
accuracy macro avg weighted avg **Comparisor* • Accuracy: K- samples. • Precision, Re Precision ind proportion of low scores so Analysis: • Interpretation correctly class with the grout • Limitations: algorithms for **Conclusion: • Both K-mean low performation using the train Sign-MNIST d In [138]: def fix_label_g	0.00 0.02 0.03 1: means output the actual posting the actual posting the actual posting the actual truth late and truth late actual the low people of the given the given the given actual to a truth actual to	o.00 o.02 o.02 derforms FC 1 Score: Bo roportion or itive instance performance ts: The lowedata points bels. rformance of data distrib I seem to ha multiple me rward neusets of the e network rop('label' label'].appl _test, num ing grayso (-1, 28, 1, 28, 28, nvolutions and fully octions, lead re el_size=(3, 2))) el_size=(3, 2))) tion from vation='re ation='sof dam(learni orical_cro ccuracy']) tion from vation='re ation='sof dam(learni orical_cro ccuracy']) in, y_trai test) (np.argmax acy:'+'\x ry['accura ory'['accura o	0.00 0.02 0.02 0.02 M in terms of a th K-means and f correctly iden es that were co ce in capturing values of Precis into distinct clu could be due to ution, or inade ave limitations etrics. ural network the dataset. So twill be train ', axis=1).valu y(fix_label_g classes=24) axis=1).valu y(fix_label_g classes=24) ale images) ale	7172 7172 7172 7172 7172 Accuracy, income a FCM exhibitified instant prectly identified instant prectly identified instant prectly identified in effective and train pecifically in effective and train pecifically in effective and to class a system of the company of the compan	bit similar performance across these metrics, with very low scores. Inces among the instances predicted as positive; Recall measures the natified; F1 Score is the harmonic mean of Precision and Recall. The less and minimizing false positives/negatives for both algorithms. and F1 Score indicate that both algorithms might struggle in sees. The clusters formed by both algorithms might not align well stors such as noisy data, inappropriate choice of clustering focessing of data before clustering. by capturing the underlying structure in the data, as reflected by the line of the network. This network will be trained and tested the national build your model. Specify kernel sizes, number long, and build your model. Specify kernel sizes, number long, and loss functions of your model. connected layers and build your model. Specify kernel sizes, number long, and loss functions of your model. connected layers and build your model. Specify kernel sizes, number long, and loss functions of your model. connected layers and F1 Score indicate that both algorithms might struggle in series and the latest properties of the series of the series of the latest properties of the latest propert
accuracy macro avg weighted avg Comparisor Accuracy: K- samples. Precision, R- Precision ind proportion of low scores st Analysis: Interpretation correctly class with the grow Limitations: algorithms for Conclusion: Both K-mea low performs: If Implement using the train Sign-MNIST d In [138]: def fix_label_g if(l>=9): return else: return X_train = train y_train = train y_train = to_cat X_test = testin y_test = testin y_test = testin y_test = to_cat # Reshape the d X_train = X_tra X_test = X_test a. Develop a s convolutional of filters, activ In [154]: # Define model model.add(Conv2 mo	0.00 0.02 0.03 1: means output ecall, and F icates the particular post aggest poor on of Result sifying the output truth lail The low perior the given and FCN ance across that a feed for a fee	0.00 0.02 0.03	0.00 0.02 0.02 0.02 0.02 M in terms of a th K-means and for correctly idented and the correctly idented are in capturing and always are limitations are into distinct clusted at a set in capturing are limitations are l	267 332 7172 7172 7172 7172 7172 7172 Accuracy, income defended instant precipitation of the composition of	bit similar performance across these metrics, with very low scores, ices among the instances predicted as positive; Recall measures the utilied; F1 Score is the harmonic mean of Precision and Recall. The rese and minimizing false positives/negatives for both algorithms. and F1 Score indicate that both algorithms might struggle in sees. The clusters formed by both algorithms might not align well tors such as noisy data, inappropriate choice of clustering rocessing of data before clustering. by capturing the underlying structure in the data, as reflected by the notwork. This network will be trained and tested by given an input image (28 x 28 = 784 pixels) from the saffy the image into 1 of 24 classes. chapter image into 1 of 24 classes. consecuted (ayers) connected (ayers) conn
accuracy macro avg weighted avg **Comparisor* * Accuracy: K- samples. * Precision, R- Precision ind proportion of low scores stands. * Analysis: * Interpretation correctly class with the grounds. * Limitations: algorithms for * Conclusion: * Future * Future * Future * Future * Convolutional * Actast = testin * Y_train = train * Y_train = train	0.02 0.03 1: means output ecall, and F icates the period of Results sifying the country of the given on of Results sifying the country of the given on and FCN ance across to a feed for and test ataset, th ap(1): (1-1) a	o.00 o.02 o.02 o.02 o.o2 o.o3	0.00 0.02 0.02 0.02 0.02 M in terms of a th K-means and forectly identifies that were concerned as the control of the contro	267 332 7172 7172 7172 7172 accuracy, income of FCM exhilitified instant precipitive of the contract of the c	bit similar performance across these metrics, with very low scores, toes among the instances predicted as positive. Recall measures the tified, F1 Score is the harmonic mean of Precision and Recall. The research minimizing false positives/negatives for both algorithms. and F1 Score indicate that both algorithms might struggle in sees. The clusters formed by both algorithms might not align well tons such as noisy data, inappropriate choice of clustering rocessing of data before clustering. by capturing the underlying structure in the data, as reflected by the article of the individual processing of data before clustering. cluster in a such as a reflected by the sign and build your model. Specify kernel sizes, number for, and loss functions of your model. cluster in a such as a such
accuracy macro avg weighted avg Comparisor Accuracy: K. samples. Precision, R. Precision ind proportion of low scores st Analysis: Interpretatic correctly class with the grou Limitations: algorithms for Conclusion: Both K-mea low performs In [138]: def fix_label_g if(1=9): return else: return X_train = train y_train = train y_train = to_ca X_test = testin y_test = to_cat # Reshape the d X_train = X_trai X_test = X_test A. Develop a s convolutional model. add(Conv2 mod	0.00 0.02 0.03 1: means output call, and F icates the p factual possing the factu	o.00 o.02 o.02 o.02 oerforms FC 1 Score: Bo roportion or itive instance or performance or data distribution and the performance or data distribution and th	0.00 0.02 0.02 0.02 0.02 M in terms of a th K-means and forectly identifies that were concerned as the control of the contro	267 332 7172 7172 7172 7172 7172 Accuracy, income of FCM exhibit diffed instant precipition of the comment of	bit similar performance across these metrics, with very low scores, toes among the instances predicted as positive. Recall measures the tified, F1 Score is the harmonic mean of Precision and Recall. The research minimizing false positives/negatives for both algorithms. and F1 Score indicate that both algorithms might struggle in sees. The clusters formed by both algorithms might not align well tons such as noisy data, inappropriate choice of clustering rocessing of data before clustering. by capturing the underlying structure in the data, as reflected by the article of the individual processing of data before clustering. cluster in a such as a reflected by the sign and build your model. Specify kernel sizes, number for, and loss functions of your model. cluster in a such as a such
accuracy macro avg macro avg macro avg meighted avg Comparisor Accuracy: K. samples. Precision, R. Precision ind proportion of low scores st Analysis: Interpretatic correctly clast with the grout Limitations: algorithms for Conclusion: Both K-mea low performs Idin [138]: def fix_label_g if(1>=9): return using the train Sign-MNIST d if [138]: def fix_label_g if(1>=9): return vtrain = train y_train = x_train y_train	0.00 0.02 0.03 1: means output ecall, and F icates the p factual poss aggest poor on of Resul sifying the independence of the given and truth lai The low pee or the given ance across a feedfo an	0.002 0.02 0.02 0.02 0.02 0.02 0.02 0.0	## 1000 ## 1002 ##	267 332 7172 7172 7172 7172 7172 7172 7172	this inflat performance across these metrics, with very low scores, cost among the instances predicted as positive. Recall measures the indiced TS store is the harmone mean of Personan and Recall The reason of Personan and Recall The research minimizing false positives/negatives for both algorithms. and F1 Score indicate that both algorithms might struggle in some scores. The clusters formed by both algorithms might not align well than such as noisy data. Inappropriate choice of clustering occasing of data before clustering. by capturing the underlying structure in the data, as reflected by the content of the property of the property of the property of the image into 1 of 24 classes. by the network. This network will be trained and tested to given an input image (28 x 28 = 784 pixels) from the softly the image into 1 of 24 classes. by softly the image into 1 of 24 clas
accuracy macro avg weighted avg Comparisor Accuracy: K. samples. Precision, R. Precision ind proportion of low scores st Analysis: Interpretatic correctly clas with the grou Limitations: algorithms for Conclusion: Both K-meal low performs Conclusion: Both K-meal low performs In [138]: def fix_label_g if(label): return X_train = train y_train = to_ca X_train = train y_train = to_ca X_train = X_train y_train = to_ca X_train = X_train y_train = to_ca X_train = X_train y_train = X	0.00 0.02 0.03 1: means outposed in the proposed in the given of th	erforms FC 1 Score: Bo roportion or itive instance performance ts: The low of data points bels. rformance of data distrib 1 seem to he multiple me rward neusets of the e network rope('label').ap y_train, n p('label').ap y_train ding graysc (-1, 28, 28, n volutions, lea re el_size=(3, 2))) el_size=(3, 2)) in, y_trai test) dam(learni orcical_cro orcuracy']) in, y_trai test) ry['accuracy (np. argmax ary: '+'x ry['accuracy (ny. argmax ary: '+'x ry['accuracy (ny. argmax ary: '-'y '-'o', tre-font, sti dation Ac ss, 'g-o', y-re-in, sti dation Ac atrix and ts for Neu the confusi idation Ac ss, 'g-o', y-re-in, sti dation Ac ss, 'g-o', y-re-in, sti dation Ac atrix and ts for Neu the confusi idation Ac ss, 'g-o', y-re-in, sti dation Ac ss, 'g-o', y-re-in, sti dation Ac atrix and ts for Neu the confusi atrix and	## O.00 ## O.02 ## M in terms of a ## th K-means and ## fcorrectly iden ## creation in the content of the ## creation in the content of the ## of the ## of the content of the ## of the	267 332 7172 7172 7172 7172 7172 7172 7172	this imitial performance across these metrics, with very too scores, ocea among the instances predicted as positive Receil research to the individence of Special Market Profession and Recoil. The search minimaling faste positive Area Profession and Recoil. The search minimaling faste positive Area Profession and Recoil. The search minimaling faste positive Area Profession and Recoil. The search minimaling faste positive Area Profession and Recoil. The search minimaling faste positive Area Profession and Recoil. The search minimaling faste positive Area Profession and Recoil Area Pr
accuracy macro avg metro a	0.00 0.02 0.03 1: Immeans outpressed of the process of the proce	erforms FC 1 Score: Bo reportion or itive instance performance ts: The low or data points bels. rformance or data distrib 1 seem to ha multiple me rward nets ests of the enetwork rop('label label'].ap y_train, n p('label', ap y_train, n p('label', ap y_train, n p('label', ap y_train, n p('label', ap y_train, n p('label').ap ing ry sc (-1, 28, 28, n validation ction='re ation='re ation=	## O 00 ## O 02 ## O 03 ##	alues agap). values alues alues agap). values alues alues	to similar performance across these metrics, with very low scores, ros among the instances predicted all possible Recall innections the Medich of Screen in the methods of Screen in the Medich of Screen in the members of the Medich of Screen in the Screen in the Casteria ordered by both algorithms might strugget in the Casteria ordered by the Medich of Casteria ordered by the Medich of Screen in the Casteria ordered by the Medich of Screen in the Casteria ordered by the Indian in Indian Indian ordered by the Medich of Screen in the Casteria ordered by the Indian of Screen in the Casteria ordered by the Indian of Screen in the Casteria ordered by the Indian of Screen in the Casteria ordered by the Indian of Screen in the Casteria ordered by the Indian of Screen in the Casteria ordered by the Indian ordered by the I
accuracy macro ave generators ave ge	0.00 0.02 0.03 1: Imeans outp ecall, and F icates the p factual pos uggest poor on of Resul sifying the und truth la The low pe or the given Ins and FCN ance across It a feedfo and ataset, th ap(1): (1-1) 1 ing_data.d ing_data.[' tegorical(y ata (assum in.reshape(- oreshape(- o	erforms FC 1 Score: Bo roportion or itive instance reperformance ts: The low of data points bels. rformance or data distrib 1 seem to ha multiple me rward neusets of the enetwork rop('label label'].ap y_train, n p('label'].ap y_train, n p('label'].ap y_train, n p('label').ap y_train ingraysc (-1, 28, 28, 28, 28, 28, 28, 28, 28, 28, 28	0.00 0.02 0.02 0.02 0.02 0.02 Minterms of a th K-means and formetty iden formetty iden certic and were come in capturing values of Precis into distinct clu ould be due to ution, or inade ave limitations etrics. Iral network is diases. Si cwill be train ', axis=1).val y(fix_label_g _clases=24) axis=1).val y(fix_label_g _clases=24) ale images) al Neural Ne connected la irning rate, of all Neural Ne connected	267 332 37172 7717	and initial professionates across those motions, with very low stores, see among the initiances proditional appointment Recall measures the fallouff 1 Store is all memorisement of President and Recall Televisian and Reca
accuracy machos avg meished avg Comparisor Accuracy: K- samples. Precision ind proportion of low scores si Analysis: Interpretatic correctly clas with the grou Limitations: algorithms for Conclusion: Both K-mea low perform low scores si Analysis: Interpretatic correctly clas with the grou Limitations: algorithms for Conclusion: Both K-mea low perform low scores si Analysis: Interpretatic correctly clas with the grou Limitations: algorithms for Conclusion: Accuracy: Limitations: algorithms for Limitations: algorith	0.00 0.02 0.03 1: means outpost of the state of the properties of the given of t	0.00 0.02 0.03	## 10.00 ## 10.	267 332 7172 7172 7172 7172 7172 7172 7172	the maximum to hidden layers composed of states the state of the state
accuracy macroal avg weighted avg weighted avg Comparisor Accuracy: Precision, Racuracy: Precision, Racuracy: Precision, Racuracy: Interpretation of low scores store the content of low scores store store the content of low scores store the content of low scores store store the content of low scores store store store store the content of low scores store store store the content of low scores store store store the content of low scores store store store store the content of low scores store	0.00 0.02 0.03 1: means output ceall, and Ficates the post of actual post output post	0.00 0.02 0.02 0.02 0.02 0.02 0.02 0.02 0.02 0.02 0.02 0.02 0.02 0.02 0.03	## 1	accuracy, income and accuracy income and accuracy income accur	and infore performance correct three matters, with very low shores, test among the instances predicted as positives Recall resumes the filled IT Scare is the Permission report of these and risk matter the earth risk matter than a continuous production and Recall The search risk matter than a continuous report of the footh algorithms may be really a continuous three distributions of the production of the production of the production of the search of the production of the produ

Silhoutte Analysis For Optimal k:

Silhoutte Analysis For Optimal k