[3]: [ [4]: [	<pre>x_train,y x_test,y_ clf = Dec</pre>	<pre>pd.read v_train = test = t cisionTre</pre>	d_csv('D:/ = train_df cest_df.il	BITS/DL/As  iloc[:,1: cc[:,1:].v	].values,	/Assignment1/mn Assignment1/mni train_df.iloc[: t_df.iloc[:,0]. y',min_samples_	st_test.csv') ,0].values values	t_fraction_lead	E=0.0001
[5]: [ [5]: [6]: [ [7]: [	<pre>clf = DecisionTreeClassifier(criterion='entropy', min_samples_leaf=2, min_weight_fraction_leaf=0.0001     _depth=15, min_samples_split=4)  clf.fit(x_train, y_train)  DecisionTreeClassifier(ccp_alpha=0.0, class_weight=None, criterion='entropy',</pre>								
[7]: [8]:	acc = accuracy_score(y_test,prediction_test)  acc = accuracy_score(y_test,prediction_test)  acc  0.8874  importances = clf.feature_importances_ ft = clf.feature_importances_ ft = tr*25500 plt.imshow(ft.reshape((28,28))) plt.title("Feature Importances learnt via Decision Tree\ncriterion: Entropy,\nTest Accuracy={}%".fo (acc*100))  Text(0.5, 1.0, 'Feature Importances learnt via Decision Tree\ncriterion: Entropy,\nTest Accuracy=88 4%')  Feature Importances learnt via Decision Tree criterion: Entropy, Test Accuracy=88.74%  0  5- 10- 15- 20-								
	sns.heatm plt.ylabe plt.xlabe plt.title	el("Corre el("Predi e("Confus 1.0, 'C	ect Label" icted Labe sion Matri Confusion	') el") .x for Deci Matrix for Decision Tre	sion Tree\ Decision		ues_r',fmt='',ar ={}%".format(acc uracy=88.74%')		False)
	Confusion Matrix for Decision Tree  Model Accuracy=88.74%  O - 915								
	print("Clt)))	assification Re	Predicted  ation Report for ecision  0.92 0.96 0.86	Decision T recall f	ree: 1-score 0.93 0.97 0.87	980 1135 1032	lassification_re	port(y_test,prediction	
	accur macro weighted	5 6 7 8 9 acy avg avg	0.85 0.87 0.83 0.91 0.92 0.86 0.88	0.85 0.89 0.85 0.89 0.90 0.82 0.86	0.85 0.88 0.84 0.90 0.91 0.84 0.87	1010 982 892 958 1028 974 1009			
11]:	indices =	np.args	sort(impor	rtances)[::	-1]				

