Name:

Hanliang Jiang

Netid:

hj33

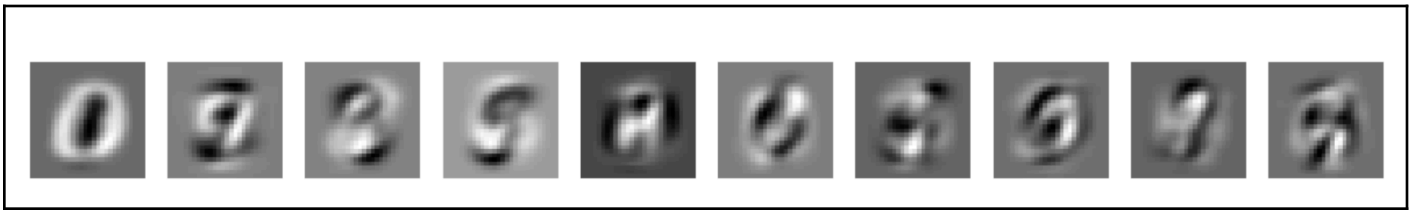
CS 441 - HW2: PCA and Linear Models

Complete the sections below. You do not need to fill out the checklist.

Total Points Available	**[ ] / 160**
1. PCA on MNIST	
a. Display 10 principal component vectors	[ ] / 5
b. Display scatterplot	[ ] / 5
c. Plot cumulative explained variance	[ ] / 5
d. Compression and 1-NN experiment	[ ] / 15
2. MNIST Classification with Linear Models	
a. LLR / SVM error vs training size	[ ] / 20
b. Error visualization	[ ] / 10
c. Parameter selection experiments	[ ] / 15
3. Temperature Regression	
a. Linear regression test	[ ] / 10
b. Feature selection results	[ ] / 15
4. Stretch Goals	
a. PR and ROC curves	[ ] / 10
b. Visualize weights	[ ] / 10
c. Other embeddings	[ ] / 15
d. One city is all you need	[ ] / 15
e. SVM with RBF kernel	[ ] / 10

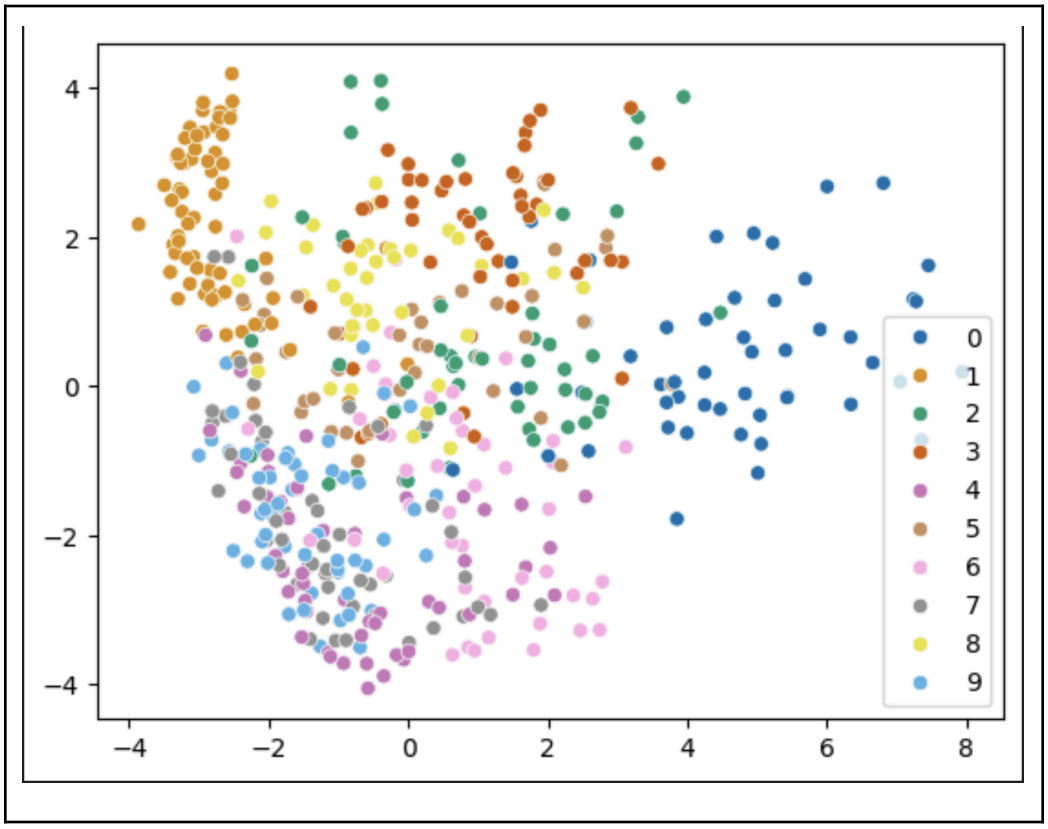
1. PCA on MNIST

a. Display 10 principal component vectors

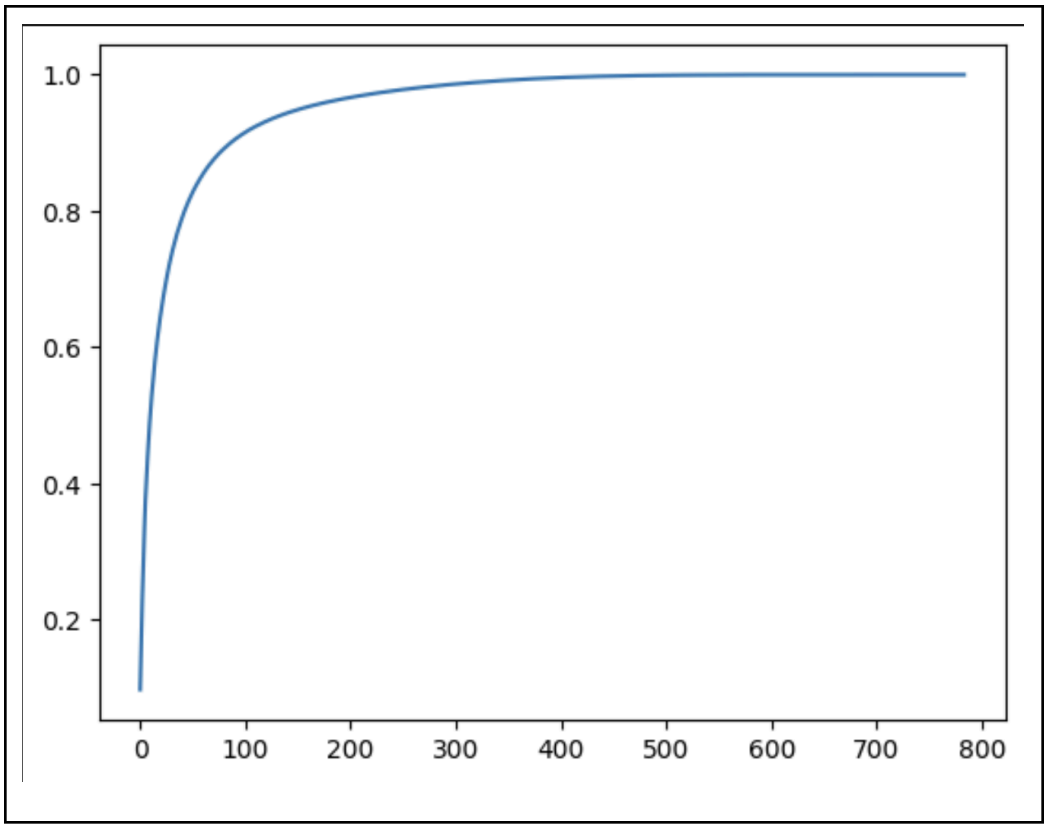


b. Display scatterplot

Scatterplot `x_train[:500]` for the first two PCA dimensions. Show a different color for each label.



c. Plot cumulative explained variance



d. Compression and 1-NN experiment

Number of components selected

	Total Time (s)	Test Error (%)	Dimensions
Brute Force (PCA)	3.038s	0.38%	86
Brute Force	8.268s	0.44%	784

2. MNIST Classification with Linear Models

a. LLR / SVM error vs training size

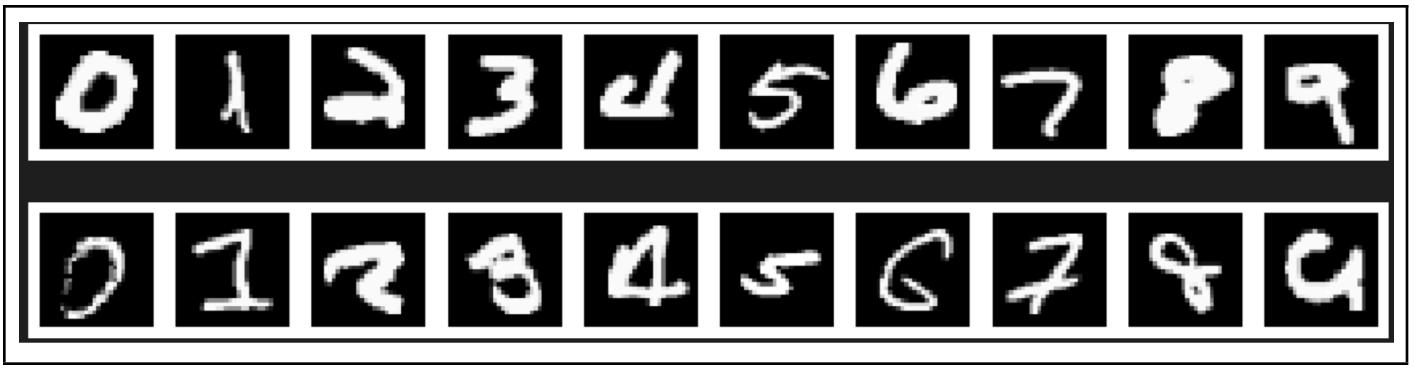
Test error (%)

# training samples	LLR	SVM
100	32.5%	32.4%
1,000	13.6%	16.1%

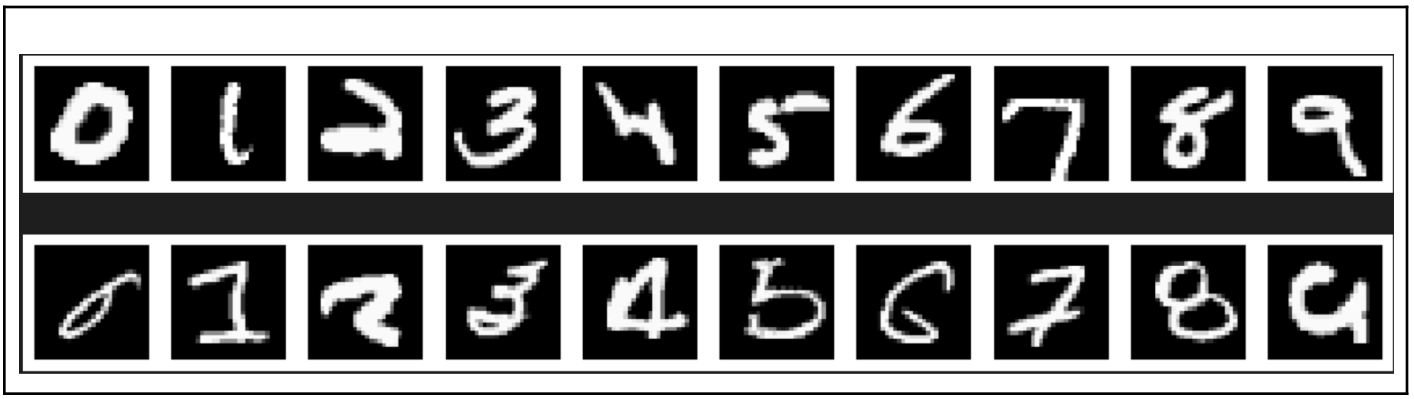
10,000	9.5%	11.1%
60,000	7.37%	8.2%

b. Error visualization

LLR



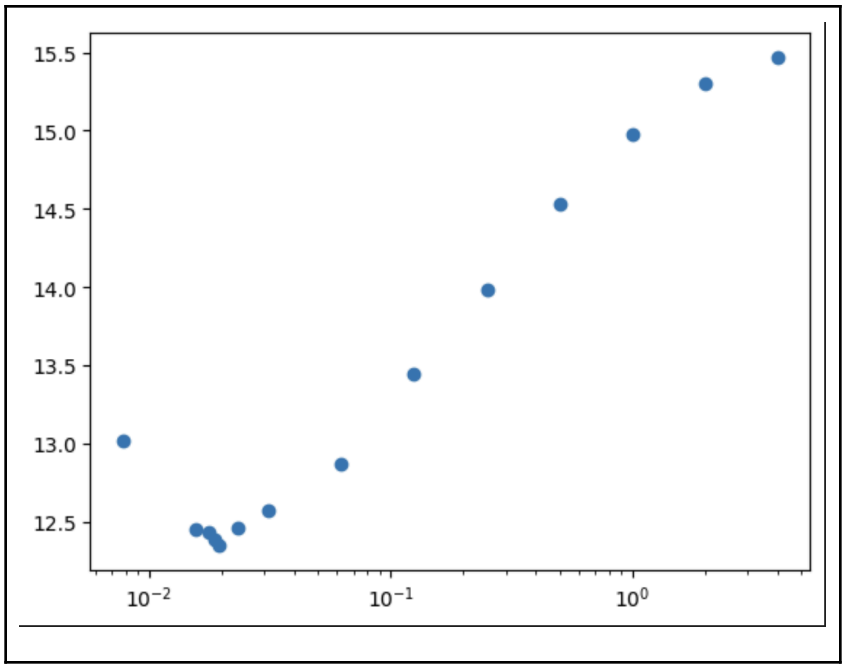
SVM



c. Parameter selection experiments

	SVM
Best C value	0.01855
Validation error (%)	12.39%
Test error (%)	13.62%

Plot C value vs validation error for values tested



3. Temperature Regression

a. Linear regression test

Test RMSE

	Linear regression
Original features	2.16
Normalized features	2.16

Why might normalizing features in this way not be as helpful as it is for KNN?

Linear regression trains model parameters according to training data distribution, so scales of the input features do NOT really matter for linear regression, but matters to KNN, so normalizing features is not helpful for linear regression.

b. Feature selection results

Feature Rank	Feature number	City	Day
1	334	Chicago	-1
2	347	Minneapolis	-1
3	405	Grand Rapids	-1
4	366	Kansas City	-1
5	361	Cleveland	-1
6	307	Omaha	-2
7	367	Indianapolis	-1
8	264	Minneapolis	-2
9	9	Boston	-5
10	236	Springfield	-3

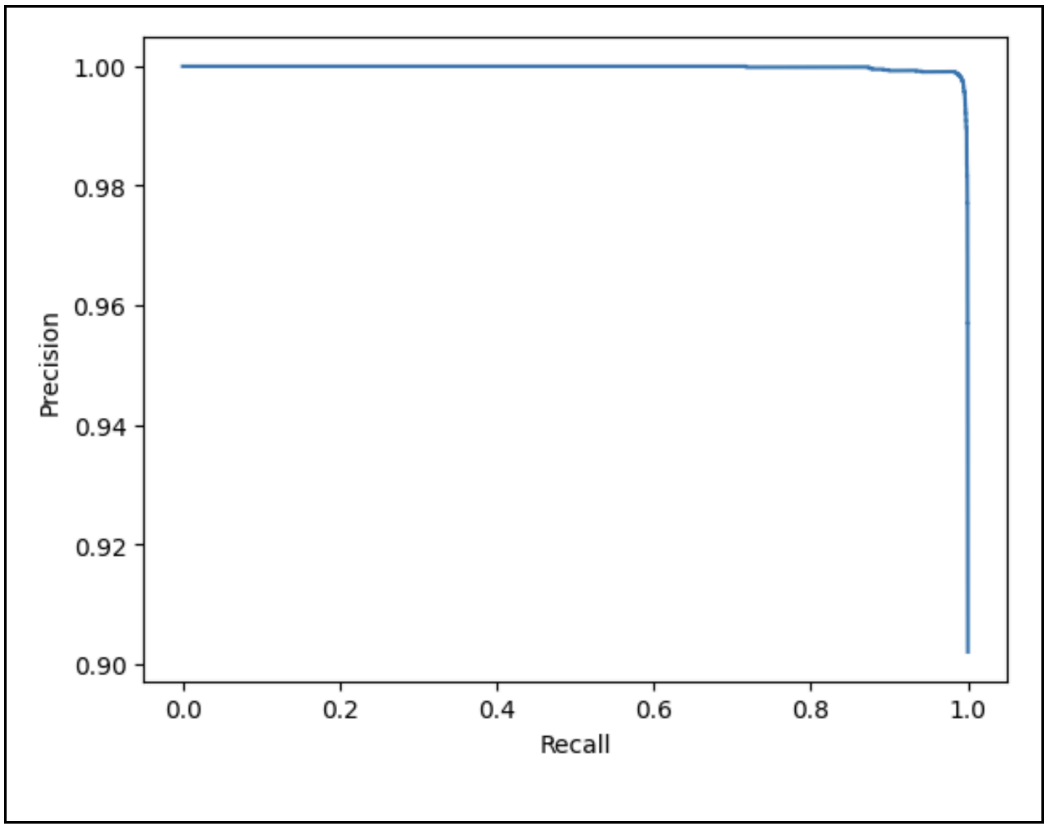
Test error using only the 10 most important features for regression

	Linear Regression
RMS Error	2.06

4. Stretch Goals

a. PR and ROC curves

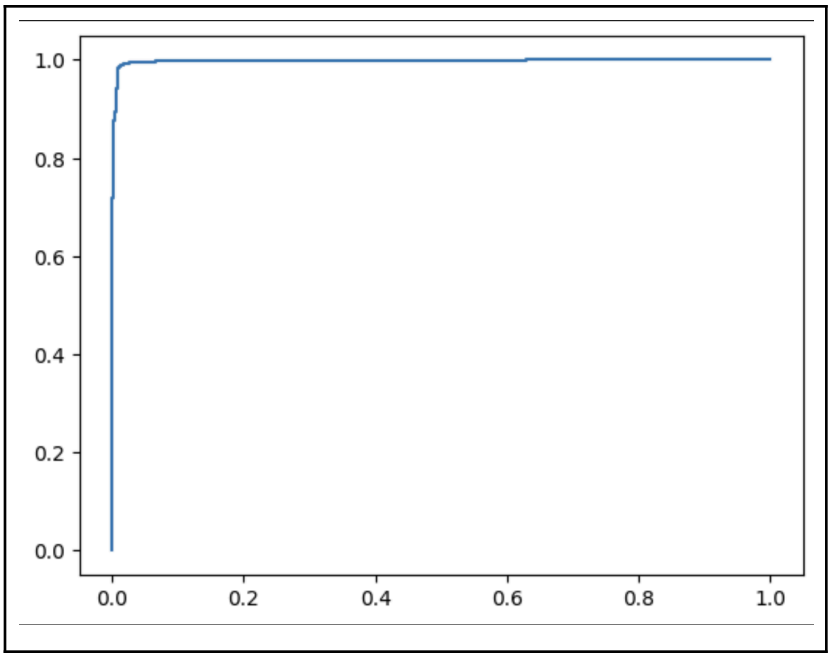
PR plot



Average Precision

99.98%

ROC plot

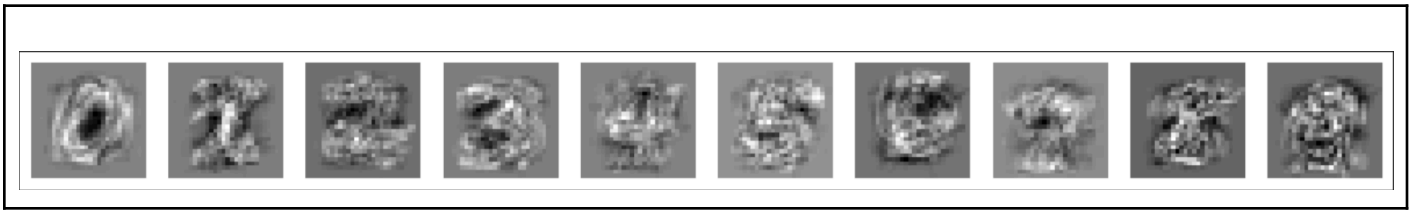


Area under the curve (AUC)

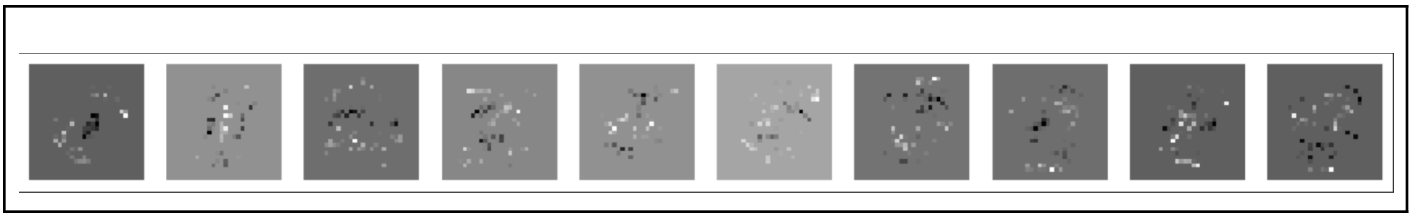
99.83

b. Visualize weights

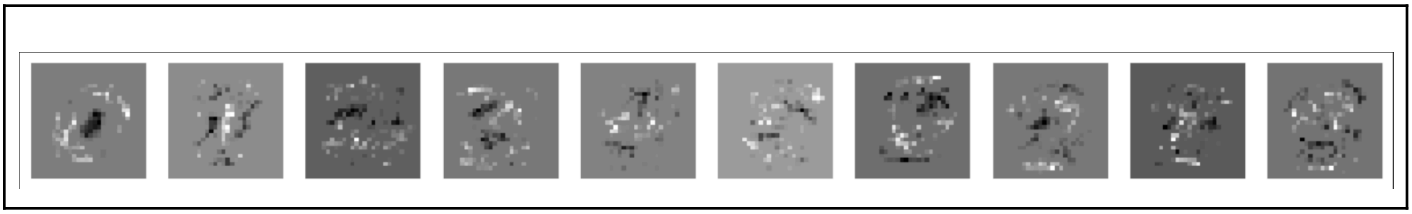
LLR - L2



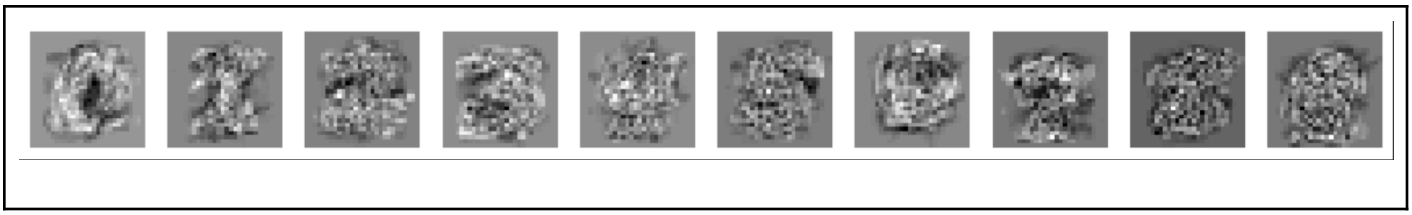
LLR - L1



LLR - elastic



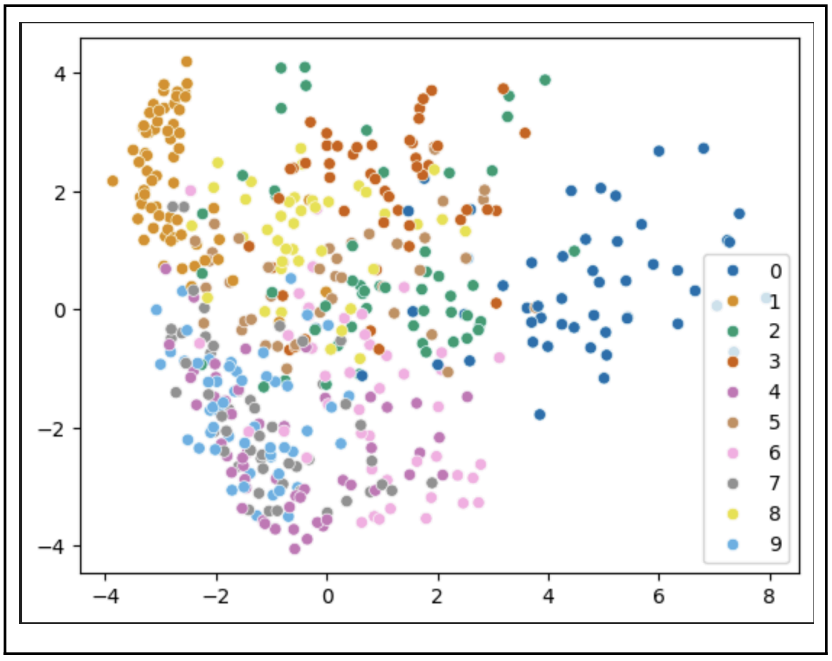
SVM



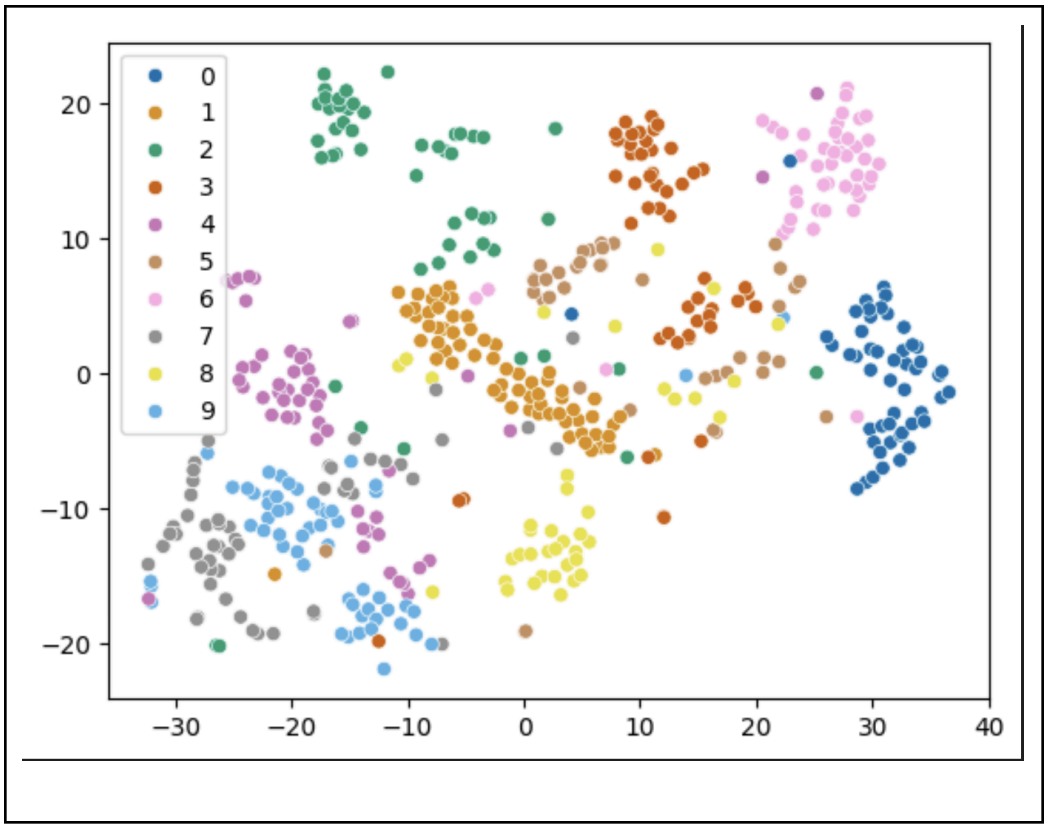
c. Other embeddings

Display 2+ plots for TSNE, MDA, and/or LDA, and copy PCA plot from 1b here.

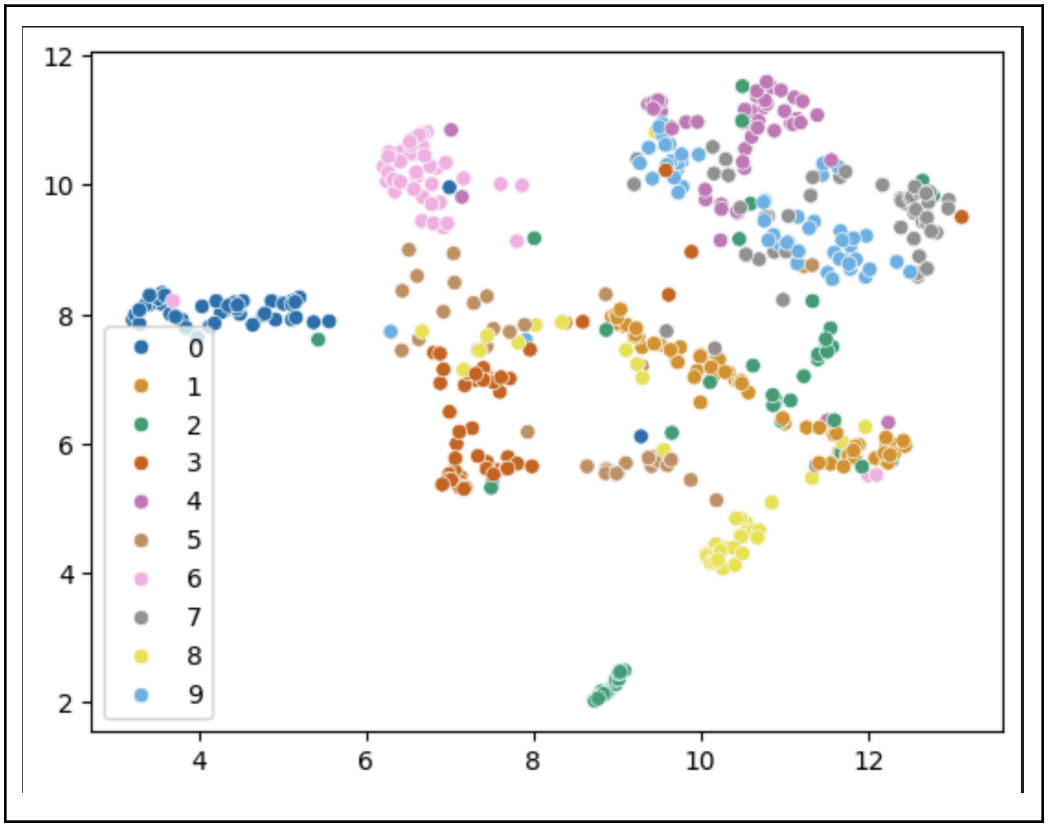
PCA



t-SNE


0
1
2
3
4
5
6
7
8
9
20
10
0
−10
−20
−30
−20
−10
0
10
20
30
40

UMAP


0
1
2
3
4
5
6
7
8
9
12
10
8
6
4
2
4
6
8
10
12

d. One city is all you need

City

'St. Louis'

Test error using features only from that city

3.13

Explain your process (in words):

For each city of feature_to_city, I first select all the features of that particular city of x_train and x_val, then use ridge(L2) regression method to get rmse. Finally I found 'St. Louis' give the smallest rmse.
Then I train with only 'St. Louis' and test on the test set to get the rmse.

e. Compare linear SVM and SVM with RBF kernel

Test accuracy (%)

# training samples	SVM-Linear	SVM-RBF
100	32.4%	34.4%
1,000	16.1%	9.2%
10,000	11.1%	4.1%
60,000	8.2%	2.1%

Acknowledgments / Attribution

For part of stretch goal problems, I used gpt to look up function syntax.