University of Nebraska Lincoln

Practical Machine Learning
ECEN 478/878

Assignment 1

Fall 2024

Data Exploration and A Study of the k-Nearest Neighbors Model

---

ECEN 478: 100 points
ECEN 878: 120 points

Last Name: Swanson

First Name: Zachary

NUID: 88795124

Obtained Score:

Introduction

This experiment explores the use of the k-nearest neighbor (kNN) model for binary and multi-class classification problems using the UCI Adult Income and the CIFAR-10 datasets, respectively. The kNN model is an example of an analogy-based learning algorithm. Analogy-based implies that the model makes predictions based on similarities (i.e. analogies) to other instances that the model has seen before.

The kNN model may be considered a simple model as it does not require any explicit training process or parameter estimation. Instead, it makes predictions by directly comparing new instances to the stored training data. When a new data point is introduced, the kNN algorithm calculates the distance between this point and all the other data points in the training set, typically using a metric like Euclidean distance. Based on the closest neighbors, it assigns a class label (for classification tasks) or predicts a value (for regression tasks). This "lazy" approach to learning means that all computation occurs at prediction time, making it straightforward but computationally expensive with large datasets. Additionally, the model's performance is highly dependent on the choice of the distance metric and the number of neighbors (k).

This experiment also explores general topics related to machine learning. These topics include hyperparameter tuning, feature selection, classification performance metrics, and the curse of dimensionality.

Methodology

The experiment was conducted in two parts, binary and multiclass classification, each with additional sub-experiments. To perform binary classification on the UCI Adult Income dataset, the data was first preprocessed by loading it into a Pandas DataFrame and exploring its structure. Categorical features were one-hot encoded, and missing values were handled by replacing them with the median value for each feature. After preparing the data, it was split into training and test sets and converted into NumPy arrays for further analysis.

Following preprocessing, several k-NN classification experiments were conducted. In **Experiment 1**, a k-NN model was trained without feature standardization, and key performance metrics such as accuracy, precision, recall, F1 score, and the confusion matrix were recorded. **Experiment 2** applied standardization to the features before training, with the same metrics reported. **Experiment 3** involved generating ROC and precision-recall (PR) curves, identifying the optimal threshold from the PR curve, and evaluating performance based on this threshold.

In **Experiment 4**, Sequential Feature Selection was used to identify the 10 most significant features, which were then used to train a k-NN model. The model's performance was evaluated using the same metrics as in previous experiments. For **Experiment 5** (graduate-level), four different feature subsets were selected based on correlation with the target, and k-NN models were trained on each subset. The dataset was standardized, and hyperparameters were tuned to optimize performance. Metrics such as accuracy, precision, recall, F1 score, and the confusion matrix were reported for each experiment, with hyperparameters (n_neighbors, p, and weights) adjusted to improve the model's results.

To perform multi-class classification on the CIFAR-10 dataset, the data was first preprocessed. The CIFAR-10 dataset consists of 60,000 32x32 color images across 10 classes, with 50,000 images used for training and 10,000 for testing. The images were flattened from their original dimensions (32x32x3) to a 1D array of 3072 features using the NumPy reshape function. The labels, initially loaded as 1D column vectors, were also converted into 1D arrays using the ravel function. Finally, the feature data was scaled using min-max normalization by dividing each value by 255.0.

In **Experiment 6**, a k-NN classifier was trained on the preprocessed dataset to perform multi-class classification. The model's training accuracy, test accuracy, and confusion matrix were reported. While hyperparameter tuning using grid search was suggested, due to the high dimensionality of the data and the computational expense, predefined values of n_neighbors=5 and p=1 were used, with other hyperparameters set to their default values.

Results

Table 1 shows the accuracy, precision, recall and F1 scores for experiments one through six. Additionally, the accuracy is presented for both the training and test sets. This helps provide some indication of how well the model generalizes to new data. Optimal thresholds were also provided for experiments three and four, where optimal is relative to the F1 score. The optimal thresholds are indicated on the precision-recall curves provided for experiment three and four in Figure 3 and Figure 4 of the Appendix, respectively. These figures illustrate how the optimal F1 score is achieved at the crossover point of the precision and recall curves. The receiver operating characteristic (ROC) curve for experiment two is also provided in Figure 2 of the Appendix.

Table 1 - Classification performance metrics for experiments one through six.

		Train Accuracy	Test Accuracy	Test Precision	Test Recall	Test F1
Experiment 1		1.000	0.768	0.523	0.444	0.480
Experiment 2		1.000	0.836	0.701	0.561	0.623
Experiment 3	Opt thresh = 0.43	1.000	0.836	0.661	0.660	0.660
Experiment 4	Non-opt thresh	0.856	0.852	0.807	0.509	0.625
	Opt thresh = 0.35	0.854	0.852	0.799	0.518	0.628
Experiment 5	Subset 1 features:	0.855	0.844	0.714	0.593	0.648
	Subset 2 features:	0.857	0.851	0.731	0.604	0.662
	Subset 3 features:	0.788	0.785	0.549	0.610	0.578
	Subset 4 features:	0.854	0.841	0.712	0.575	0.636
Experiment 6		0.535	0.38	0.46	0.38	0.37

Table 2 provides the optimal hyperparameters `n_neighbors`, `p` (power of the Minkowski distance metric), and `weights` (the neighbor weighting scheme). These hyperparameters were determined using Scikit-Learn's GridSearchCV function which leverages cross-validation (CV) to achieve the grid search.

Table 2 - Optimal hyperparameters determined via hyperparameter tuning for experiments one through six.

		n_neighbors	p	weights
Experiment 1		3	1	distance
Experiment 2		51	1	distance
Experiment 3	Opt thresh = 0.43	51	1	distance
Experiment 4	Opt thresh = 0.35	87	1	distance
Experiment 5	Subset 1 features:	17	1	uniform
	Subset 2 features:	21	2	uniform
	Subset 3 features:	83	2	uniform
	Subset 4 features:	17	50	uniform
Experiment 6		5	1	uniform

Experiments four and five explored feature selection. Experiment four utilized the Scikit-Learn `SequentialFeatureSelector` class to determine optimal features. Given the training sets and the available features running on an eight-core CPU, the automated feature selection took roughly one hour and fifteen minutes to determine the top ten features. The selected features are presented in Table 3.

Table 3 - Top ten selected features for experiment four.

Index	Feature Name	Index	Feature Name
3	capital-gain	29	marital-status_Married-civ-spouse
4	capital-loss	36	occupation_Exec-managerial
10	workclass_Self-emp-not-inc	45	occupation_Tech-support
22	education_Doctorate	59	native-country_Columbia
26	education_Prof-school	74	native-country_India

Regarding Question 1 of the assignment and referencing Table 1, feature selection improved the test accuracy and precision by 2% and 22%, respectively. However, this improvement was also at the cost of reduced recall and F1 scores. For this dataset, the benefits of feature selection are not obvious because the dataset does not have high-dimensionality (when compared to that of the CIFAR-10 image data). Therefore, the computational cost of determining the optimal features seems excessive compared to the marginal gain in accuracy and the tradeoff of precision and recall. However, as the experiment progresses towards the high-dimensional CIFAR-10 dataset and the prediction times grow considerably large with sub-par performance, the value of feature selection and dimensionality reduction become more attractive.

Experiment five required manual feature selection based on Pearson correlation coefficients between the features and the target class. Four subsets were selected as listed in Table 5 through Table 8 of the Appendix. The justifications for these four subsets are provided below:

1. The top-10 positively correlated features were selected because greater correlation between the target should indicate greater discriminability.

2. The top-10 positively correlated features with redundancy removed. Specifically, education-num provides a numerical representation of the other education labels. Hence, education_Bachelors, education_Masters, etc. were removed.
3. The top-10 negatively correlated features were selected to explore how well the kNN model could perform if the features indicate the opposite class.
4. The top-10 absolute correlated features were selected to explore how well the kNN model could discriminate positively and negatively correlated features to predict the target.

The performance metrics for experiment six are listed in Table 1 and the resulting confusion matrix is presented in Table 4. Note that in general the diagonal possesses the largest value for the corresponding row and column. However, there are some classes that feature a considerable number of off diagonal predictions (e.g. see column 2).

Table 4 - Confusion matrix for experiment six.

Predicted	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**All**
True											
0	582	9	101	10	49	7	25	7	195	15	1000
1	139	288	89	50	130	40	44	17	168	35	1000
2	145	5	456	54	206	30	55	13	34	2	1000
3	82	11	215	246	162	109	101	14	52	8	1000
4	92	4	259	40	489	18	43	14	40	1	1000
5	72	4	214	151	166	266	64	14	43	6	1000
6	36	4	259	74	285	27	288	1	25	1	1000
7	116	10	155	50	259	58	38	267	37	10	1000
8	154	20	47	33	43	17	10	6	662	8	1000
9	166	90	71	40	91	30	46	27	213	226	1000
All	1584	445	1866	748	1880	602	714	380	1469	312	10000

The following statements address Question 2 of the assignment. As described in the methodology section, the CIFAR-10 images have 3072 features (32x32 pixels each with three color channels) which constitutes high dimensionality. In high-dimensional spaces, the distance between points becomes less meaningful due to the curse of dimensionality. In other words, as dimensionality increases, data points tend to be equidistant from each other, making it difficult for k-NN to effectively distinguish between similar and dissimilar instances. Furthermore, Figure 1 shows ten examples of the CIFAR-10 data set. Looking at the two truck examples in the top row, there are several factors that would cause the kNN model to poorly identify these images as "neighbors". The scale, pose, and color of the trucks are different. The background and lighting conditions are different. The differences in geometry are also different due to the elevated truck bed in the right example. All the drastic differences are superimposed on top of each other to

further hinder a kNN model that is already facing poor performance due to the curse of dimensionality.

Figure 1 - Example CIFAR-10 images.

In contrast, the provided MNIST example notebook demonstrates that the kNN model can achieve high performance on image data. However, there are some key differences with this data. First, the images are grey-scale, i.e. the dimensionality is already reduced by a factor of three without the three color channels. Second, all the MNIST images have a consistent background (black or pixel value zero). This is in sharp contrast to the highly variable backgrounds illustrated in Figure 1. It may also be argued that the digits present a simpler pattern and exhibit less intra-class variability as was demonstrated by the significantly different geometries of the trucks in Figure 1. Lastly, the MNIST digits have been cropped and scaled such that all digits have a relatively similar scale and position in the image. Hence, the model can ignore more of the edge pixels which could further reduce the dimensionality.

It seems that higher performance could be achieved using kNN with other computer vision techniques. Semantic segmentation techniques could be used to extract the object pixels and set all background pixels to black. Furthermore, scale invariant feature transform (SIFT) and histogram of oriented gradients (HOG) could be used to extract key-points from the image and the kNN could use these key-points to make predictions. The dimensionality of the key-points would be much lower than the raw pixel values. Lastly, a pretrained convolution neural network (CNN) could be truncated and used to extract low-level features from the images and the kNN model could try to predict neighbors in a lower-dimensional feature space. For example, if a convolutional block halved the spatial dimensions and doubled the channels (32x32x3=3,072 to 16x16x6=1,536) the kNN model would be handling half the dimensions and the extracted features may be more discriminative than the raw pixels.

Conclusion

This report investigated the application of the k-nearest neighbor (k-NN) model for both binary and multi-class classification tasks using the UCI Adult Income and CIFAR-10 datasets. Through a series of carefully designed experiments the model's performance was examined under various preprocessing techniques, feature selection methods, and hyperparameter tuning strategies.

The findings reveal that while the k-NN model demonstrates strong performance on the UCI Adult Income dataset, achieving high accuracy and reliable metrics through effective preprocessing and feature selection, it faces significant challenges when applied to the CIFAR-10 dataset. The curse of dimensionality profoundly impacts the model's ability to distinguish between similar instances in high-dimensional spaces, leading to lower test accuracy and precision. In contrast, the simpler MNIST dataset serves as a testament to the model's strengths, as the lower dimensionality and consistent patterns enable k-NN to achieve high classification performance.

Moreover, this exploration into feature selection highlighted the potential benefits in reducing computational costs and improving model performance, particularly in high-dimensional datasets. However, the marginal gains observed in lower-dimensional cases, such as the UCI dataset, emphasize the need for careful consideration of feature selection strategies and their applicability.

In conclusion, while the k-NN model is a straightforward and intuitive approach to classification, its effectiveness is contingent upon the nature of the dataset and the preprocessing methods employed. Future work could explore advanced techniques, such as using feature extraction methods or integrating deep learning models to enhance the k-NN framework, particularly for complex datasets like CIFAR-10. This research underscores the importance of understanding the limitations and capabilities of different models in the broader landscape of machine learning.

Appendix

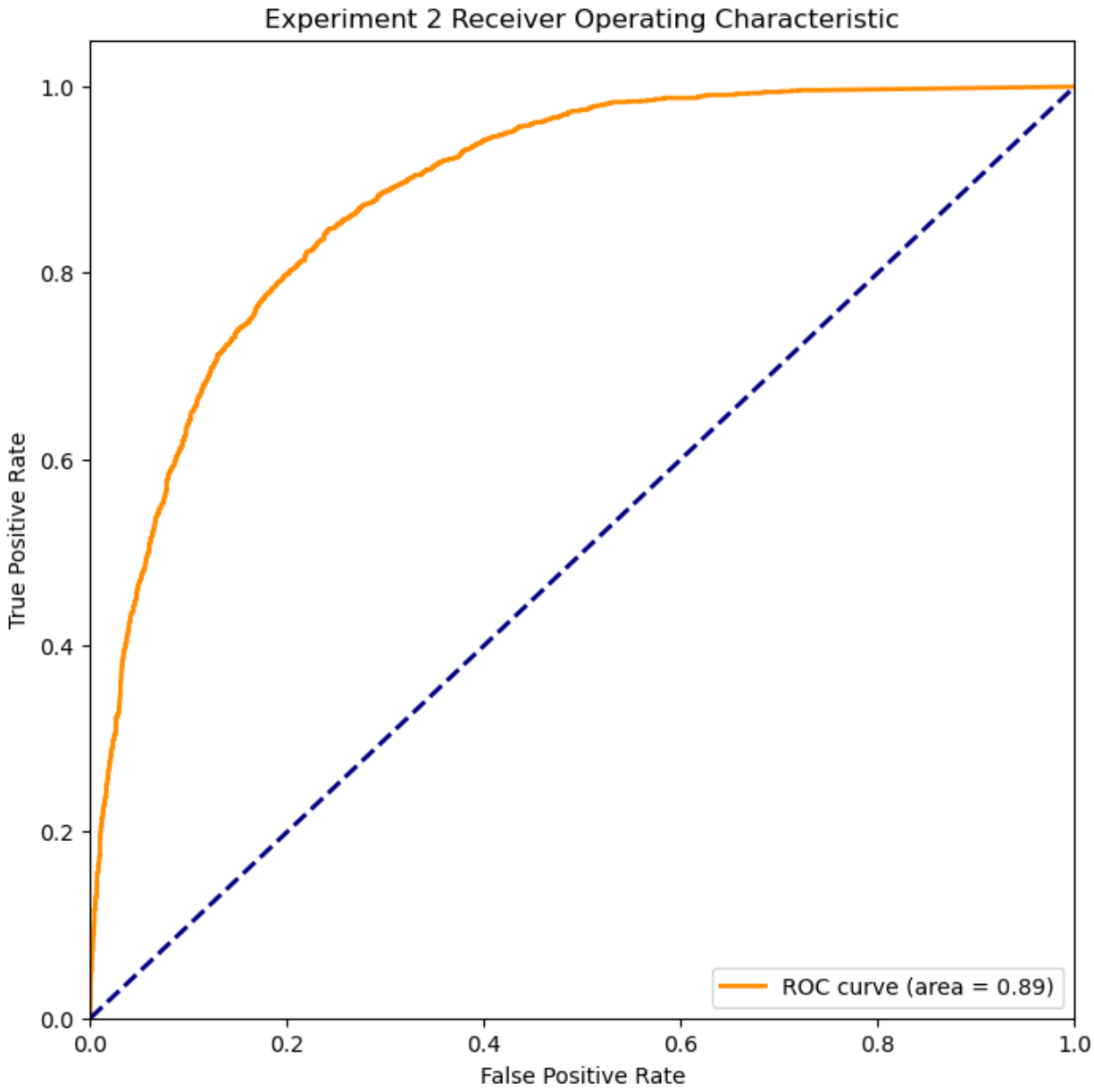

Figure 2 - Receiver Operating Characteristic (ROC) curve for experiment two. Note, area under the curve (AUC) is provided in the legend.

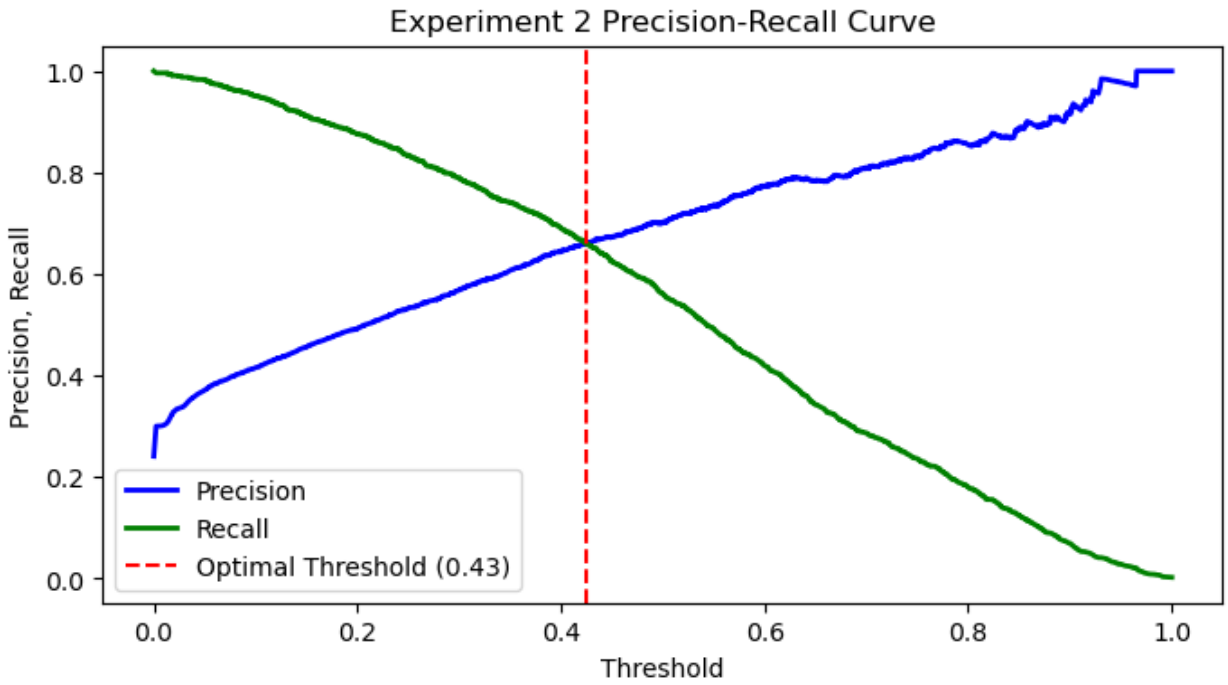

Figure 3 - Precision-recall for experiment two with optimal threshold indicated.

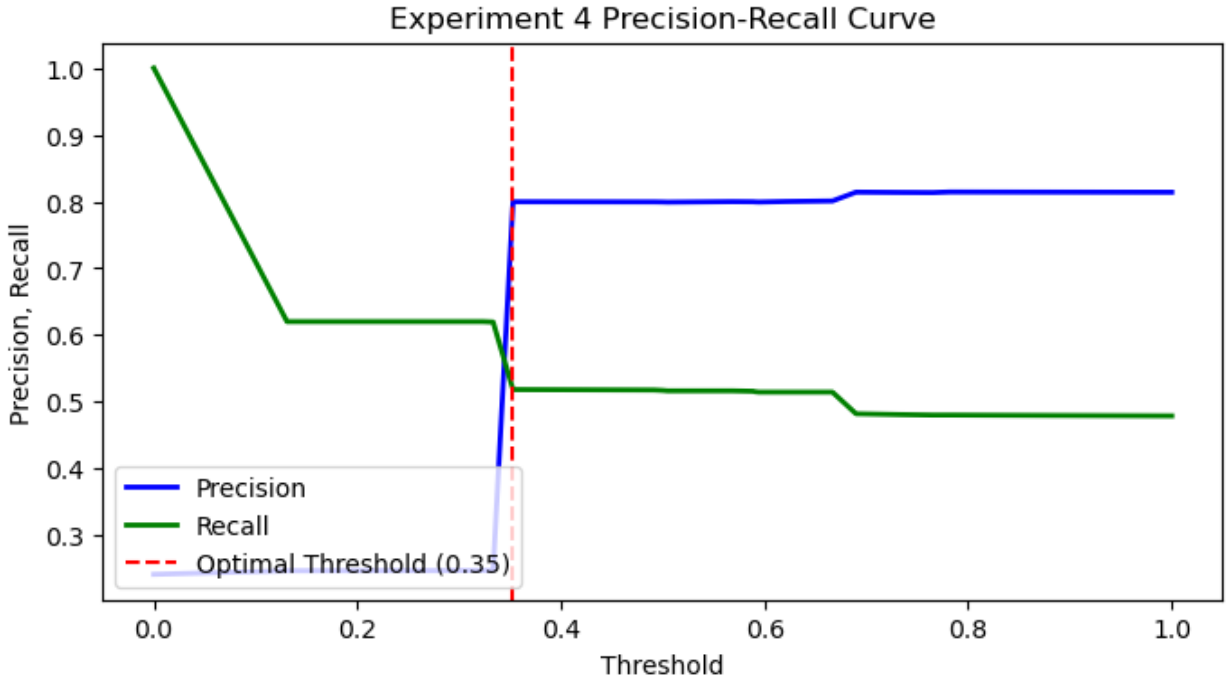

Figure 4 - Precision-recall for experiment four with optimal threshold indicated.

Table 5 – Top ten positively correlated (Pearson) features with target (income_>50K) used for experiment five.

Feature Name	ρ	Feature Name	ρ
marital-status_Married-civ-spouse	0.44	sex_Male	0.21
education-num	0.34	occupation_Exec-managerial	0.21
age	0.23	education_Bachelors	0.18
hours-per-week	0.23	education_Masters	0.17
capital-gain	0.22	education_Prof-school	0.15

Table 6 - Same experiment five features from Table 5 with redundant categorical education features removed.

Feature Name	ρ	Feature Name	ρ
marital-status_Married-civ-spouse	0.44	sex_Male	0.21
education-num	0.34	occupation_Exec-managerial	0.21
age	0.23	capital-loss	0.15
hours-per-week	0.23	workclass_Self-emp-inc	0.14
capital-gain	0.22	relatioship_Wife	0.12

Table 7 - Top ten negatively correlated (Pearson) features with target (income_>50K) used for experiment five.

Feature Name	ρ	Feature Name	ρ
marital-status_Never-married	-0.32	education_HS-grad	-0.13
relatioship_Own-child	-0.23	workclass_Private	-0.13
relatioship_Not-in-family	-0.19	race_Black	-0.09
occupation_Other-service	-0.16	occupation_Handlers-cleaners	-0.09
relatioship_Unmarried	-0.14	education_11th	-0.09

Table 8 - Top ten absolute correlated (Pearson) features with target (income_>50K) used for experiment five.

Feature Name	ρ	Feature Name	ρ
marital-status_Married-civ-spouse	0.44	relatioship_Own-child	-0.23
education-num	0.34	capital-gain	0.22
marital-status_Never-married	-0.32	sex_Male	0.21
age	0.23	occupation_Exec-managerial	0.21
hours-per-week	0.23	relatioship_Not-in-family	-0.19