






Lab 8

Classification

Nov 2, 2023

Department of Computing
The Hong Kong Polytechnic University

New services provided by PolyU GenAI

GPT-3.5  0613	You can steer chat conversation towards a desire length, format, style, level of details and language used. Please refer to Hints & Tips on using ChatGPT for details.
GPT-4  0613	Model has more trained data compare with GPT-3.5 and can handle conversation with increased accuracy and precision.
Microsoft Bing Chat 	Launched Feb 2023, the new search engine built into Microsoft Edge browser include conversational AI chatbot, image to text function running on a version of OpenAI model enhanced by Microsoft. Please refer to Microsoft Bing Chat User Guide for details.
Image To Text 	Service leverage Azure Cloud Cognitive vision AI models and GPT-3.5 to provide textual response based on your upload image and input prompt. Please refer to Hints & Tips on image to generate text .
Text To Image  Stable Diffusion SDXL 1.0	Service leverage Stability.ai Cloud Stable Diffusion SDXL 1.0 model to generate image based on your input prompt. Please refer to Hints & Tips on text to generate image .

Today's Arrangement

- Assignment 4 Check
- Classification basics
- Exercises
- Assignment 5



Assignment 4 Check

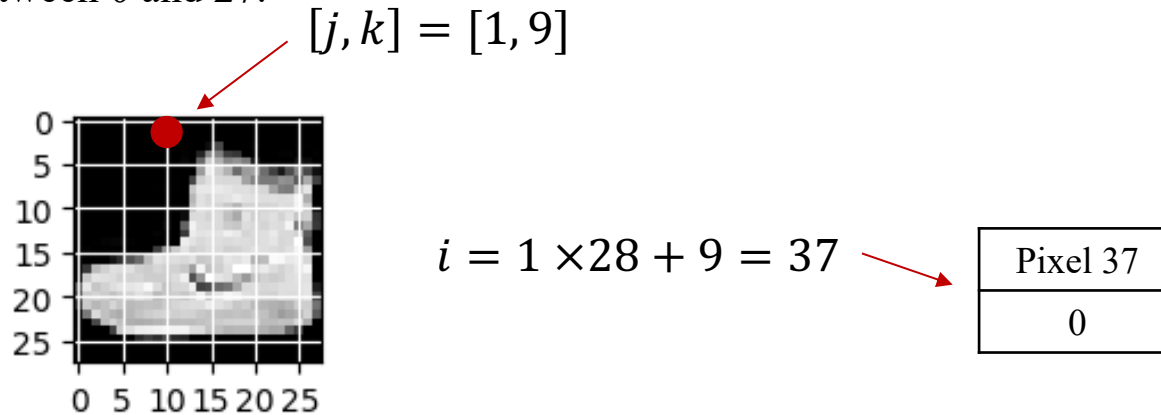


Assignment 4

As an artificial intelligence expert, a fashion company has invited you to help cluster the clothing images. You will be provided with a dataset with each sample corresponding to an image. Each image measures 28 pixels in height and 28 pixels in width, totaling 784 pixels.

These images are stored in a table, representing the pixels across 784 columns. Each column has a single pixel value, which is an integer ranging from 0 (black) to 255 (white).

To locate a pixel on the image, suppose we have the pixel at row j and column k of a 28×28 matrix; we can find its pixel value at the i -th column of the table, where $i = j \times 28 + k$ where j and k are integers between 0 and 27.



Assignment 4

Your task is to use clustering algorithms to cluster these images. For example, you may find some images belonging to the cluster of pants, while others may be T-shirts, etc.

In the training stage, you will be provided with a public dataset comprising 3000 samples. The goal is to **cluster these samples and assign the clustering results** to each sample. **In the test stage**, your predictions will **be evaluated using the adjusted rand score as the final performance**. Please note that using any pre-trained models is not allowed, nor is the use of any form of external data permitted.

Hint: The number of clothing clusters is unknown. You need to determine the cluster number via analysis.

Hint: You can try various clustering algorithms to find the most suitable one.

Hint: You can use **internal evaluation metrics**, like the Davies-Bouldin Index, Dunn Index, and Silhouette Coefficient, to fine-tune hyperparameters.

Assignment 4

- Please download the public dataset from https://drive.google.com/file/d/1XJ0D1L4K-urd83nwCIB4BT75c2rSPTaj/view?usp=share_link
- Please use the following Python template for submission (You can copy the code below from lab7-Exercise.ipynb).
- How should we comprehend the template code?

```
def read_data_from_csv(path):
    """Load datasets from CSV files.
    Args:
        path (str): Path to the CSV file.
    Returns:
        X (np.ndarray): Features of samples.
        y (np.ndarray): Labels of samples.
    """
    assert os.path.exists(path), f'File not found: {path}!'
    assert os.path.splitext(path)[-1] == '.csv', f'Unsupported file type {os.path.splitext(path)[-1]}!'

    data = pd.read_csv(path)
    column_list = data.columns.values.tolist()

    if 'Label' in column_list:
        # For the test phase, the label column is provided for external evaluation.
        column_list.remove('Label')
        X = data[column_list].values
        y = data['Label'].astype('int').values
        return X, y
    else:
        # For the training phase, the label column is not provided. You should use internal evaluation.
        X = data[column_list].values
        return X
```

```
X_public = read_data_from_csv('assignment_4_public.csv')
print('Shape of X_public:', X_public.shape)
```

```
# remove and make your own predictions.
preds = np.full(len(X_public), -1,
                 dtype=int)
```

```
"""
CODE HERE!
"""
```

```
submission = pd.DataFrame({'Label': preds})
submission.to_csv('assignment_4.csv', index=True, index_label='Id')
```

Assignment 4 Check

- Check your answer:
 - **Assignment4_Answer.ipynb**: available on Blackboard or <https://drive.google.com/file/d/1tHf0mdsTq9xrZFMYYMeWBygPuA2-CPz7/view?usp=sharing>
 - **The ground-truth labels of the public dataset**: available on Blackboard

Tips: internal clustering validation

- Internal evaluation metrics: External clustering validation and internal clustering validation are the two main categories of clustering validation. The main difference is whether or not external information (e.g., labels) is used for clustering validation. [1]
- In Assignment 4, you cannot access the ground-truth labels, and thus you should use internal clustering validation methods, like the Silhouette Score, to fine-tune hyperparameters.

- Silhouette Score:
$$s(i) = \frac{b(i) - a(i)}{\max(a(i), b(i))}$$
 - a: The mean distance between a sample and all other points in the same class.
 - b: The mean distance between a sample and all other points in the next nearest cluster.
 - A high value indicates that the object is well-matched to its own cluster and poorly matched to neighboring clusters.
 - Best: 1; Worst: -1

[1] Liu Y, Li Z, Xiong H, et al. Understanding of internal clustering validation measures[C]//2010 IEEE international conference on data mining. IEEE, 2010: 911-916.

Tips: curse of dimensionality

- Many applications require the clustering of large amounts of high-dimensional data. Most clustering algorithms, however, do not work effectively and efficiently in high-dimensional space, which is due to the so-called "curse of dimensionality". [2]
- In Assignment 4, the samples include 28 pixels in height and 28 pixels in width, totaling 784 pixels, thereby constituting 784-dimensional features. Therefore, you might observe ChatGPT making recommendations such as:
 - Since the dataset consists of image data, we can use dimensionality reduction techniques like Principal Component Analysis (PCA) to extract meaningful features.
 - Apply PCA to reduce the dimensionality of the dataset while preserving most of the variance.
 - ...

[2] Hinneburg A, Keim D A. Optimal grid-clustering: Towards breaking the curse of dimensionality in high-dimensional clustering[J]. 1999.

Tips: extensive experiments

- Conduct extensive comparative experiments.
- Which methods should be compared? You can conduct experiments on each stage after constructing the data mining process. For example, you can explore alternative clustering algorithms for comparison:

Please conduct a comprehensive survey on the following questions:

What are some other algorithms that can be used for dimensionality reduction besides PCA?

What are some other clustering algorithms that can be used besides KMeans?



1. Dimensionality Reduction Algorithms:

- Independent Component Analysis (ICA): Separates a multivariate signal into additive subcomponents that are statistically independent.
- t-distributed Stochastic Neighbor Embedding (t-SNE): Reduces high-dimensional data to a lower-dimensional representation while preserving the local structure of the data.
- Linear Discriminant Analysis (LDA): Maximizes the separation between classes while minimizing the variance within each class.
- Non-negative Matrix Factorization (NMF): Decomposes a non-negative matrix into the product of two lower-rank non-negative matrices.
- Autoencoders: Neural network models that learn to reconstruct the input data from a compressed representation, forcing the model to learn meaningful features.

2. Clustering Algorithms:

- DBSCAN (Density-Based Spatial Clustering of Applications with Noise): Groups together data points that are close to each other and separates outliers.
- Agglomerative Clustering: Hierarchical clustering algorithm that starts with each sample as its own cluster and merges clusters based on a linkage criterion.
- Mean Shift: Iteratively shifts the centroids of clusters towards the densest regions of data until convergence.
- Spectral Clustering: Uses the eigenvectors of a similarity matrix to perform dimensionality reduction and then applies a clustering algorithm.
- Gaussian Mixture Models (GMM): Represents each cluster as a Gaussian distribution and assigns data points to the most likely cluster based on the probability distribution.

These are just a few examples of dimensionality reduction and clustering algorithms. There are many more algorithms available, each with its own strengths and weaknesses. The choice of algorithm depends on the specific problem and the characteristics of the data. It is recommended to experiment with different algorithms and evaluate their performance to find the most suitable one for a given task.



Tips: extensive experiments

- Conduct extensive experiments on the hyperparameters and select the satisfactory model, e.g., [3]:
 - Quantitative experiments, e.g., the Silhouette Score.
 - Qualitative experiments, e.g., visualizing each cluster.



Initial solution: PCA + K-means
Clustering



After trying many alternatives: ISOMap + Gaussian
Mixture Model (with tuned hyper-parameters)



Classification basics



Classification basics

- Classification
 - Definition: Given a set of training data points along with associated training labels, determine the class label for an unlabelled test instance.
 - Attempts to learn the relationship between a set of feature variables and a target variable.
 - Many real-world applications:
 - 2-Class: Email Filtering (Spam or not); Disease Diagnosis (E.g., a tumor as malignant or benign); ...
 - Multi-Class: Handwritten Digit Recognition (Classifying digits 0-9); News Article Categorization (Categorizing news articles into sports, politics, entertainment, etc.); ...
 - Two phases of Classification (generally): Training and Testing
 - Output of a classification: Discrete Label or Numerical Score (for each class label)

Classification basics

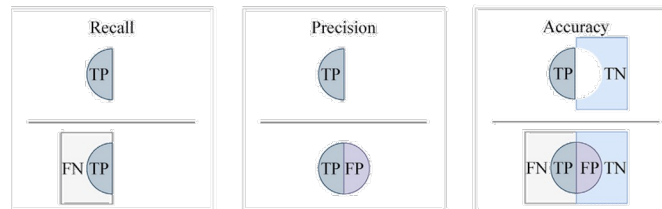
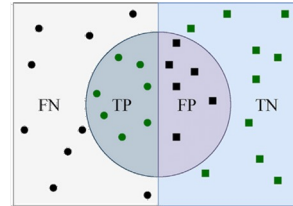
- Classification algorithms
 - Decision Trees (Introduced in lecture 4 and lab 6)
 - Nearest Neighbor Classifier <https://www.youtube.com/watch?v=HVXime0nQeI>
 - SVM Classifiers (Introduced in lecture 5)
 - Neural Networks (Will be introduced lecture 6)
 - Naïve Bayes Classifier <https://www.youtube.com/watch?v=O2L2Uv9pdDA>
 - Logistic Regression <https://www.youtube.com/watch?v=yIYKR4sgzI8>
 - ...

Classification basics

- Evaluation metrics

For Models with discrete output:

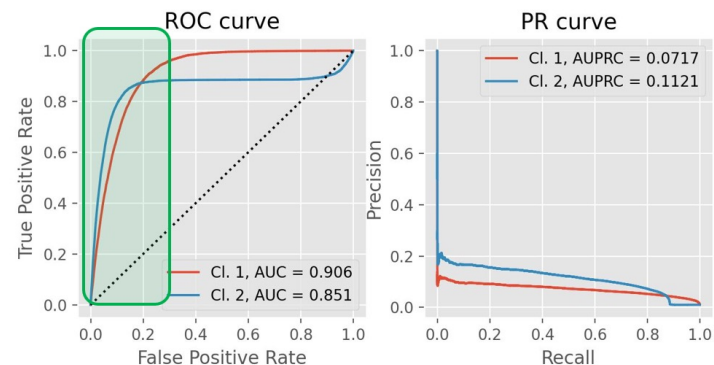
- Accuracy
- Recall
- Precision
- F1-score



$$\text{F1 Score} = \frac{2}{\frac{1}{\text{Precision}} + \frac{1}{\text{Recall}}} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

For Probabilistic Models:

- Receiver Operating Characteristic Area Under Curve (ROC-AUC)
- Precision-Recall Area Under Curve (PR-AUC)
- ...





Exercises



Exercise 1: KNN

- Dataset: Glass Classification

- Features:

1. RI: refractive index;
2. Na: Sodium (unit measurement: weight percent in the corresponding oxide, as are features 3-9)
3. Mg: Magnesium
4. Al: Aluminum
5. Si: Silicon
6. K: Potassium
7. Ca: Calcium
8. Ba: Barium
9. Fe: Iron

- Response: Glass types (discrete 7 values)

1. building_windows_float_processed
2. building_windows_non_float_processed
3. vehicle_windows_float_processed
4. vehicle_windows_non_float_processed (none in this database)
5. Containers
6. Tableware
7. headlamps

Exercise 1: KNN

- Tasks:

1. Feature Selection

- Utilize a correlation matrix to identify and eliminate features that either do not significantly influence the output or are highly correlated with other features.
- This step helps to reduce redundancy in the data and can improve model performance.

2. Feature Scaling

- Implement feature scaling to ensure all data is on a similar scale.
- This is particularly important for distance-based algorithms, as it prevents features with larger magnitudes from dominating the model due to their larger numerical values.

3. Parameter Tuning

- Execute the K-Nearest Neighbors (KNN) algorithm with varying parameters, such as different values of K (the number of neighbors) and various distance metrics (e.g., Euclidean, Manhattan).
- This can help identify the optimal parameters that yield the best model performance.

Exercise 1: KNN

Task 1: Feature Selection

Utilize a correlation matrix to identify and eliminate features that either do not significantly influence the output or are highly correlated with other features.

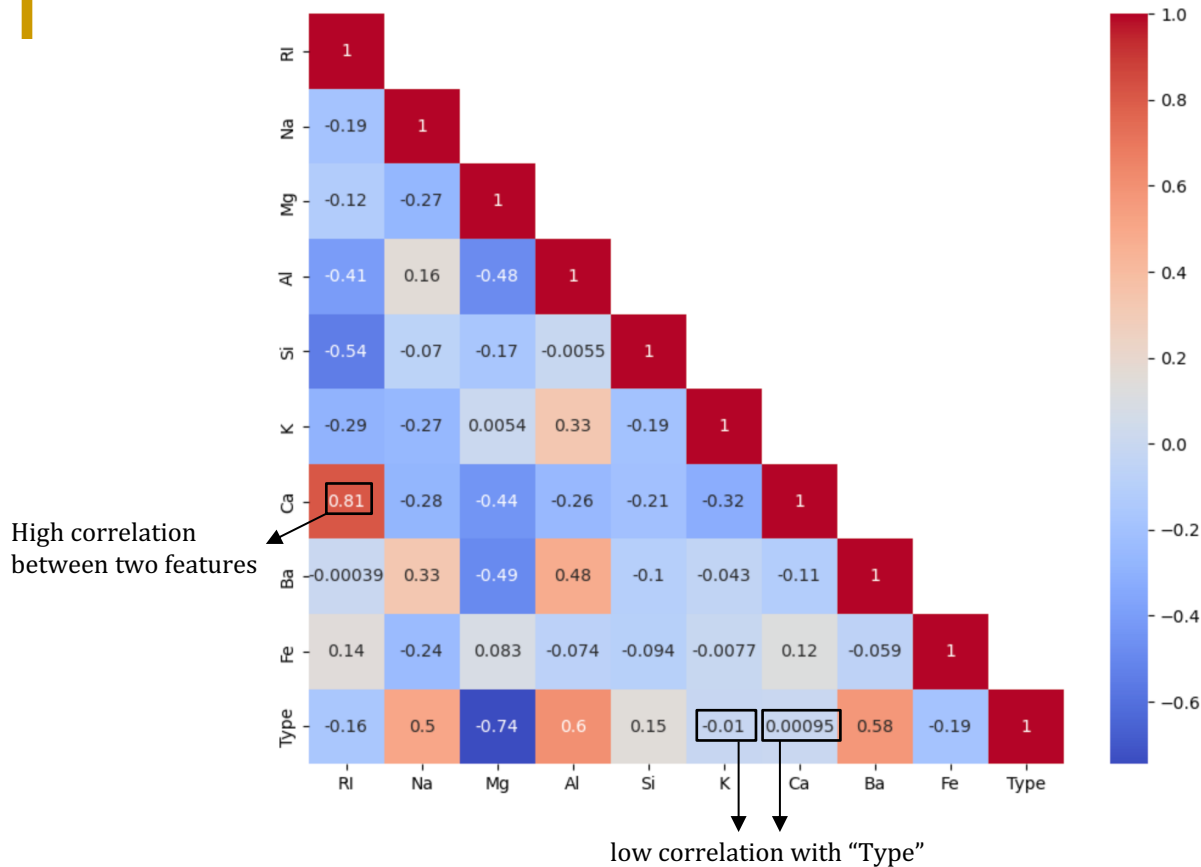
- Prompt:

Write a Python script to do the following:

- Firstly, read the dataset named "glass.csv" with 9 features and 1 output (named "Type").
 - Secondly, apply a correlation matrix to the features and output. Visualize the correlation matrix.
 - Thirdly, based on the correlation matrix, drop some features that do not significantly influence the output.
 - Finally, based on the correlation matrix, if there are some features highly correlated with each other, only remain one of them.
-

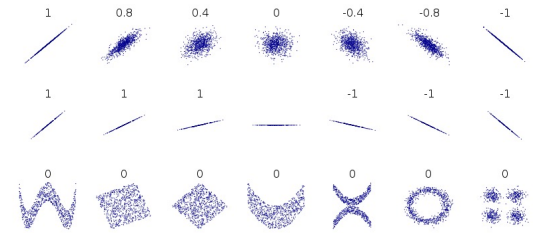
Exercise 1: KNN

- Resulting correlation matrix:



Here we use Pearson correlation coefficient:

$$r = \frac{\sum_{i=1}^n (X_i - \bar{X})(Y_i - \bar{Y})}{\sqrt{\sum_{i=1}^n (X_i - \bar{X})^2} \sqrt{\sum_{i=1}^n (Y_i - \bar{Y})^2}}$$



It only measures linear correlation.

Exercise 1: KNN

Task 2: Feature Scaling

Implement feature scaling to ensure all data is on a similar scale.

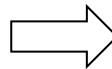
- Prompt:

Based on the above code, write additional lines to do the following. Do not rewrite the above code.

- Implement feature scaling to ensure all features are on a similar scale. Do not change the output column "Type".

- Resulting dataset:

	RI	Na	Mg	Al	Si	Ba	Fe	Type
0	1.52101	13.64	4.49	1.10	71.78	0.00	0.0	1
1	1.51761	13.89	3.60	1.36	72.73	0.00	0.0	1
2	1.51618	13.53	3.55	1.54	72.99	0.00	0.0	1
3	1.51766	13.21	3.69	1.29	72.61	0.00	0.0	1
4	1.51742	13.27	3.62	1.24	73.08	0.00	0.0	1
...
209	1.51623	14.14	0.00	2.88	72.61	1.06	0.0	7
210	1.51685	14.92	0.00	1.99	73.06	1.59	0.0	7
211	1.52065	14.36	0.00	2.02	73.42	1.64	0.0	7
212	1.51651	14.38	0.00	1.94	73.61	1.57	0.0	7
213	1.51711	14.23	0.00	2.08	73.36	1.67	0.0	7



	RI	Na	Mg	Al	Si	Ba	Fe	Type
0	0.432836	0.437594	1.000000	0.252336	0.351786	0.000000	0.0	1
1	0.283582	0.475188	0.801782	0.333333	0.521429	0.000000	0.0	1
2	0.220808	0.421053	0.790646	0.389408	0.567857	0.000000	0.0	1
3	0.285777	0.372932	0.821826	0.311526	0.500000	0.000000	0.0	1
4	0.275241	0.381955	0.806236	0.295950	0.583929	0.000000	0.0	1
...
209	0.223003	0.512782	0.000000	0.806854	0.500000	0.336508	0.0	7
210	0.250219	0.630075	0.000000	0.529595	0.580357	0.504762	0.0	7
211	0.417032	0.545865	0.000000	0.538941	0.644643	0.520635	0.0	7
212	0.235294	0.548872	0.000000	0.514019	0.678571	0.498413	0.0	7
213	0.261633	0.526316	0.000000	0.557632	0.633929	0.530159	0.0	7

Exercise 1: KNN

Task 3: Parameter Tuning

Execute the K-Nearest Neighbors (KNN) algorithm with varying parameters, such as different values of K (the number of neighbors) and various distance metrics (e.g., Euclidean, Manhattan).

- Prompt:

Based on the above code, write additional lines to do the following. Do not rewrite the above code.

- Execute the K-Nearest Neighbors (KNN) algorithm with varying parameters, such as different values of K (the number of neighbors) and various distance metrics (e.g., Euclidean, Manhattan). Utilize grid search in this process.
-

Exercise 1: KNN

- Resulting code and best parameters:

```
: import warnings
warnings.filterwarnings("ignore",category=FutureWarning,module="sklearn")

from sklearn.model_selection import GridSearchCV
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import accuracy_score
from sklearn.model_selection import train_test_split

# Step 7: Split the dataset into input features (X) and output variable (y)
X = dataset.drop(columns=["Type"])
y = dataset["Type"]

# Step 8: Split the dataset into training and testing data
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Step 9: Define the parameter grid
param_grid = {
    'n_neighbors': [3, 5, 7, 10],
    'metric': ['euclidean', 'manhattan']
}

# Step 10: Initialize the K-Nearest Neighbors classifier
knn = KNeighborsClassifier()

# Step 11: Perform grid search to find the best parameters
grid_search = GridSearchCV(knn, param_grid, cv=5)
grid_search.fit(X_train, y_train)

# Step 12: Make predictions on the test data using the best model
best_model = grid_search.best_estimator_
y_pred = best_model.predict(X_test)

# Step 13: Evaluate the accuracy of the best model
accuracy = accuracy_score(y_test, y_pred)
print("Accuracy (Best Model): {:.2f}%".format(accuracy * 100))

# Step 14: Print the best parameters found by grid search
print("Best Parameters:", grid_search.best_params_)

Accuracy (Best Model): 72.09%
Best Parameters: {'metric': 'manhattan', 'n_neighbors': 3}
```


Exercise 2: SVM

- Support Vector Machines (SVM) are well-suited for handling data with high dimensions.
- Dataset: MNIST
 - A large database of handwritten digits
 - Contains 60,000 training images and 10,000 testing images
 - The size of each image is 28x28 pixel
 - A subset of 5,000 digits is used in this exercise.
- Tasks:
 1. Load the dataset and print some examples.
 2. Train SVM models on the dataset and utilize grid search to find the best parameters. Use the F-score as the metric.



Exercise 2: SVM

- Example prompts:

Task 1:

```
'''
```

```
# The code of generating the subset
```

```
'''
```

Based on the above code, add a few lines to show 100 examples from the subset, each class 10 examples.

Task 2:

Based on the above code, add a few lines to do the following:

- Split the subset into training and testing datasets.
- Train SVM models using training datasets. Utilize grid search to find the best parameters. Use the F-score as the metric.
- Test the model on the test dataset. Print accuracy and F-score.

Supplement about SVM

- Support Vector Machines (SVM) have several advantages:
 - **Effectiveness in high-dimensional spaces:** Still effective in cases where the number of dimensions exceeds the number of samples.
 - **Memory efficiency:** Use a subset of training points in the decision function (called support vectors), making them memory efficient.
 - **Versatility:** Different Kernel functions can be specified for the decision function.
 - Support Vector Machines in scikit-learn:
 - Provides a set of supervised learning methods used for classification, regression, and outliers detection.
 - Learn more via <https://scikit-learn.org/stable/modules/svm.html>
-



Assignment 5

Assignment 5

Online news is crucial in providing people with diverse, multifaceted perspectives on political and public issues. As an expert in artificial intelligence, an online news website seeks your assistance in predicting the popularity of online news based on its features (as shown in the table below).

The dataset contains a set of features describing published online news. The goal is to forecast their popularity on social networks. The articles with more than 1400 shares can be as popular. Your task is to predict whether a piece of online news will be popular.

Variable Name	Role	Type	Description
Feature 1	Feature	Continuous	Number of words in the title
Feature 2	Feature	Continuous	Number of words in the content
Feature 3	Feature	Continuous	Rate of unique words in the content
Feature 4	Feature	Continuous	Rate of non-stop words in the content
Feature 5	Feature	Continuous	Rate of unique non-stop words in the content
Feature 6	Feature	Continuous	Number of links
Feature 7	Feature	Continuous	Number of links to other articles
Feature 8	Feature	Continuous	Number of images
Feature 9	Feature	Continuous	Number of videos
Feature 10	Feature	Continuous	Average length of the words in the content
Feature 11	Feature	Continuous	Number of keywords in the metadata
Feature 12	Feature	Categorical	Is the article from the Lifestyle topic?
Feature 13	Feature	Categorical	Is the article from the Entertainment topic?
Feature 14	Feature	Categorical	Is the article from the Business topic?
Feature 15	Feature	Categorical	Is the article from the Social Media topic?
Feature 16	Feature	Categorical	Is the article from the Tech topic?
Feature 17	Feature	Categorical	Is the article from the World topic?
Feature 18	Feature	Continuous	Min. shares of worst keyword
Feature 19	Feature	Continuous	Max. shares of worst keyword
Feature 20	Feature	Continuous	Avg. shares of worst keyword
Feature 21	Feature	Continuous	Min. shares of best keyword
Feature 22	Feature	Continuous	Max. shares of best keyword
Feature 23	Feature	Continuous	Avg. shares of best keyword
Feature 24	Feature	Continuous	Min. shares of avg. keyword
Feature 25	Feature	Continuous	Max. shares of avg. keyword
Feature 26	Feature	Continuous	Avg. shares of avg. keyword

Assignment 5

Online news is crucial in providing people with diverse, multifaceted perspectives on political and public issues. As an expert in artificial intelligence, an online news website seeks your assistance in predicting the popularity of online news based on its features (as shown in the table below).

The dataset contains a set of features describing published online news. The goal is to forecast their popularity on social networks. The articles with more than 1400 shares can be as popular. Your task is to predict whether a piece of online news will be popular.

Variable Name	Role	Type	Description
Feature 27	Feature	Continuous	Min. shares of referenced articles
Feature 28	Feature	Continuous	Max. shares of referenced articles
Feature 29	Feature	Continuous	Avg. shares of referenced articles
Feature 30	Feature	Categorical	Was the article published on a Monday?
Feature 31	Feature	Categorical	Was the article published on a Tuesday?
Feature 32	Feature	Categorical	Was the article published on a Wednesday?
Feature 33	Feature	Categorical	Was the article published on a Thursday?
Feature 34	Feature	Categorical	Was the article published on a Friday?
Feature 35	Feature	Categorical	Was the article published on a Saturday?
Feature 36	Feature	Categorical	Was the article published on a Sunday?
Feature 37	Feature	Categorical	Was the article published on the weekend?
Feature 38	Feature	Continuous	Closeness to Latent Dirichlet Allocation (LDA) topic 0
Feature 39	Feature	Continuous	Closeness to Latent Dirichlet Allocation (LDA) topic 1
Feature 40	Feature	Continuous	Closeness to Latent Dirichlet Allocation (LDA) topic 2
Feature 41	Feature	Continuous	Closeness to Latent Dirichlet Allocation (LDA) topic 3
Feature 42	Feature	Continuous	Closeness to Latent Dirichlet Allocation (LDA) topic 4
Feature 43	Feature	Continuous	Text subjectivity
Feature 44	Feature	Continuous	Text sentiment polarity
Feature 45	Feature	Continuous	Rate of positive words in the content
Feature 46	Feature	Continuous	Rate of negative words in the content
Feature 47	Feature	Continuous	Rate of positive words among non-neutral tokens
Feature 48	Feature	Continuous	Rate of negative words among non-neutral tokens
Feature 49	Feature	Continuous	Avg. polarity of positive words
Feature 50	Feature	Continuous	Min. polarity of positive words
Feature 51	Feature	Continuous	Max. polarity of positive words
Feature 52	Feature	Continuous	Avg. polarity of negative words

Assignment 5

Online news is crucial in providing people with diverse, multifaceted perspectives on political and public issues. As an expert in artificial intelligence, an online news website seeks your assistance in predicting the popularity of online news based on its features (as shown in the table below).

The dataset contains a set of features describing published online news. The goal is to forecast their popularity on social networks. The articles with more than 1400 shares can be as popular. Your task is to predict whether a piece of online news will be popular.

Variable Name	Role	Type	Description
Feature 53	Feature	Continuous	Min. polarity of negative words
Feature 54	Feature	Continuous	Max. polarity of negative words
Feature 55	Feature	Continuous	Subjectivity of the title
Feature 56	Feature	Continuous	Sentiment polarity of the title
Feature 57	Feature	Continuous	Absolute level of subjectivity in the title
Feature 58	Feature	Continuous	Absolute level of sentiment polarity in the title
Label	Label	Categorical	0,1 (0 for not popular and 1 for popular)

Assignment 5

In the public dataset, you can train and validate your model on 30,000 samples. Then, you need to predict the labels for 5,000 samples in the private dataset, and your performance on the private dataset will determine your final score.

Hint 1: Cross-validation is important.

Hint 2: Consider preprocessing and feature engineering if it benefits your model.

Hint 3: Optimize hyperparameters for improved performance.

Hint 4: Utilize any algorithms you have learned, including the decision tree, K-nearest neighbor, support vector machine, etc. You may ensemble their predictions to achieve better performance.

Assignment 5

- Please download the public dataset from https://drive.google.com/file/d/1FoXCtnlw_0DFI3VzUVXLAoSvUHYlzhZ5/view?usp=sharing
- Please download the private dataset from <https://drive.google.com/file/d/1SxEYOYIdSPbAlzCcGp-jGjAkXzgC6FZ6/view?usp=sharing>
- Please use the following Python template for submission. (You can copy the code below from lab8-Exercise.ipynb)
- Your results will be evaluated on 5000 samples in the private dataset, using classification accuracy (The labels will be released in Lab 9).

Assignment 5

```
def read_data_from_csv(path):
    """Load datasets from CSV files.
    Args:
        path (str): Path to the CSV file.
    Returns:
        X (np.ndarray): Features of samples.
        y (np.ndarray): Labels of samples, only provided in the public
        datasets.
    """
    assert os.path.exists(path), f'File not found: {path}!'
    assert os.path.splitext(path)[-1] == '.csv', f'Unsupported file type {os.path.splitext(path)[-1]}!'

    data = pd.read_csv(path)
    column_list = data.columns.values.tolist()

    if 'Label' in column_list:
        # for the public dataset, label column is provided.
        column_list.remove('Label')
        X = data[column_list].values
        y = data['Label'].astype('int').values
        return X, y
    else:
        # for the private dataset, label column is not provided.
        X = data[column_list].values
        return X
```

```
X_public, y_public = read_data_from_csv('assignment_5_public.csv')
print('Shape of X_public:', X_public.shape) # n_sample, m_feature
(30000, 58)
print('Shape of y_public:', y_public.shape) # n_sample (30000,)
```

```
"""
CODE HERE!
"""
```

```
X_private = read_data_from_csv('assignment_5_private.csv')
print('Shape of X_private:', X_private.shape) # k_sample, m_feature
(5000, 58)
```

```
import numpy as np
```

```
# remove and make your own predictions.
preds = np.full(len(X_private), -1,
                dtype=int)
```

```
"""
CODE HERE!
e.g.,
preds = np.full(len(X_private), -1, dtype=int)
"""
```

```
submission = pd.DataFrame({'Label': preds})
submission.to_csv('assignment_5.csv', index=True, index_label='Id')
```