Epoch IIT Hyderabad
Core Selections
Round-1 Report

Pranjal Prajapati - CS23BTECH11048

Contents

1 Task 1

1.1 Task Objective

You have been provided with a dataset containing information about various pincodes across India, including their corresponding longitudes and latitudes (clustering_data.csv). Your task is to focus specifically on the pincodes of your home state. Here's what you need to do:

1. **Data Filtering:**

- Extract the entries corresponding to your home state from the dataset. Ensure you accurately filter out only those pincodes that belong to your home state.

2. **Data Visualization:**
   - You can utilize the longitude and latitude data to plot the geographical locations of these pincodes on a map (get creative!). This will help in visualizing the distribution of the pincodes across your state.
3. **Clustering Analysis:**
   - Implement the k-means clustering algorithm from scratch (do not use any pre-built k-means function).
   - Apply this algorithm to the longitude and latitude data of your filtered pincodes to identify distinct clusters within your state.
4. **Inference and Insights:**
   - Draw meaningful inferences from your clustering results. Analyze the characteristics of the clusters you identified and provide insights about the geographical distribution and potential implications.
5. **Visualization and Preprocessing:**
   - Use appropriate visualization techniques (like scatter plots, maps, etc.) to illustrate the clusters and any other relevant observations.

1.2 Solution Overview

1. **Data Preprocessing**
   - Read the clustering_data.csv file using the pandas library to obtain the corresponding pandas dataframe.
   - Remove the data entries from the dataframe which have atleast one null value.
   - Ensure each entry in the dataframe is unique by removing all duplicate entries.
   - Ensure that all the values in the Longitude and Latitude columns are numeric.
2. **Data Filtering:**
   - Keep only those entries in the dataframe which have StateName equal to the home_state variable.
   - Remove the outliers from the data for improving the effectiveness of the clustering. Here, this was accomplished via trial and error during the data visualisation step.
3. **Data Visualisation**
   - Use your home state shapefile along with the libraries geopandas and contextlily to display the plot of the (latitude,longitude) coordinates on the state map itself.

- For clarity purposes, ensure this is done after the outliers have been removed.

4. **Implementing the clustering process:**

   - Define functions for finding the euclidean distances of a particular data point from a given dataframe, normalizing a given dataframe based on extremes of the other given dataframe, and reverting the normalization of given dataframe based on extremes of the other given dataframe.
   - Implement the k_means function, which takes three parameters: dataset, k (default val = 8) and max_iterations (default value = 300) and returns the array of centroid locations and the WCSS (Within Cluster Sum of Squares) for the k-clusters finalized.

5. **Choosing an optimal value for k:**

   - Display a plot of the WCSS values vs k for all k's from k = 1 to k = 21 using the matplotlib library.
   - Find the k-value corresponding to the elbow point using the plot.
   - Call the k-means function for the optimal k value and obtain the centroid locations array.
   - Find the cluster_labels for each point in the dataframe.
   - Use the seaborn library to display a scatterplot with the point colour depending on its cluster_label.

1.3 Inferences

1. Removing the outliers from the dataset before data visualisation and clustering is of paramount importance for effective clustering.
2. Normalised distances should be used to cluster the data, in order to avoid bias towards higher-valued parameters

1.4 References

1. Understanding K-Means Clustering: https://www.youtube.com/embed/4b5d3muPQmA?si=2MWRRsMuAEiicsEy
2. Detailed explanation with algorithmic steps: https://www.simplilearn.com/tutorials/machine-learning-tutorial/k-means-clustering-algorithm
3. Shapefile obtained from: https://www.google.com/url?sa=t&source=web&rct=j&opi=89978449&url=https://github.com/HindustanTimesLabs/shapefiles/blob/master/state_ut/gujarat/district/gujarat_district_2011.csv
4. Introduction to Geopandas: https://geopandas.org/en/stable/gallery/plotting_basemap_background.html

5. Introduction to Contextlily: https://contextily.readthedocs.io/en/latest/intro_guide.html

1.5 Result / Conclusion

- Given that the dataframe corresponds to office locations in the state, the compact clusters show a sign of urbanisation in those areas: with relatively less distance from one office to the next.
- This serves as a crucial insight for companies which are looking to start / expand their business in the state.
- It also provides a basic overview to the government of the areas which require more attention from their side for development.

2 Task 2

2.1 Task Objective

The primary objective of this project is to use artificial intelligence to convert handwritten text images into digital text and subsequently perform sentiment analysis on the extracted text. The project will leverage the provided labelled dataset of handwritten alphabets (alphabets_dataset.zip) and the sentiment analysis dataset (sentiment_analysis_dataset.csv). Use of pre-trained models is not allowed. Finally, the performance is to be tested and reported on the images provided in the target_images folder (whose labels are in target_labels.csv).

2.2 Solution Overview

1. **Data Preprocessing**
   - Read the alphabets_28x28.csv file using the pandas library to obtain the corresponding pandas dataframe. Important: Get the csv file from here: https://drive.google.com/drive/folders/1oG96SkGUUQnNmI_PIiiJl7vipAM9s8P8?usp=sharing
   - Remove the data entries from the dataframe which have atleast one null value.
   - Ensure each entry in the dataframe is unique by removing all duplicate entries.
   - Ensure that all the values in the pixel columns are numeric.
   - Convert each label from its character to a numeric index for training the neural network.
   - Split the training and testing data using the popular sklearn library.
   - Reshape the training and testing data into the proper format for feeding into the CNN (Convolutional Neural Network).
   - Ensure the label values are categorical and represent 26 different classes for the CNN to classify into.
2. **Setting up the CNN**

- Use Tensorflow / Keras to set up the model architecture for image classification.
- Feed the training data into the model to train the CNN with parameters: optimizer=”adam”, loss=”categorical_crossentropy” and metrics=[’accuracy’].

3. **Taking input of the image**
   - Use the openCV library to read the image into the variable.
   - Convert the image into grayscale form to obtain the pixel data similar to the training data on which the CNN is trained.

4. **Extracting the phrase from the main image**
   - Define the functions for predicting the character from given image using the model, and padding the main image to fit the window size (here, 28).
   - Iterate through the main image in step sizes of 28 (window size) to obtain the various images of window size (dubbed character images).
   - If the character image obtained is blank, predict it as whitespace. Else, use the predict_character function defined above for the prediction.
   - Append all the predicted characters onto a blank string to obtain the extracted phrase of the main image.

5. **Predicting the sentiment from the phrase**
   - Train a simple Naive Bayes classifier using the data from the given sentiment_analysis_dataset.csv file using a vectorizer from the sklearn library.
   - Obtain the predicted sentiment of the extracted phrase by feeding it into the classifier and printing the output.

2.3 Inferences

1. Using a CNN improves the model’s accuracy of classifying images by a great extent compared to traditional ANNs.
2. Using contouring techniques to extract the characters is a good alternative to the sliding window method, but makes it very difficult to extract spaces.

2.4 References

1. How Neural Networks work: https://www.youtube.com/@3blue1brown
2. Introduction to CNNs: https://www.youtube.com/watch?v=QzY57FaENXg
3. Training a CNN for character recognition: https://www.kaggle.com/code/pahirathannithilan/character-recognition-from-handwriting
4. Contouring method: https://pyimagesearch.com/2020/08/24/ocr-handwriting-recognition-wit
5. Sliding window method: https://dl.acm.org/doi/10.1007/s11042-021-10988-9

2.5 Result / Conclusion

- Handwritten text recognition is indeed a challenging problem to tackle, especially due to the infinite variations possible in the font styles, image characteristics, etc.
- The sliding window method is a very resourceful way of tackling the problem, with the main image first being padded to be split into equal sections; with each section holding information about a particular character.
- The section size is determined by the shape of the training data on which the model is trained.