{ "cells": [ { "cell\_type": "markdown", "metadata": {}, "source": [ "\*\*Task Submitted by Mandar Karanjkar\*\*" ] }, { "cell\_type": "markdown", "metadata": { "colab\_type": "text", "id": "UID9RK1qDlVB" }, "source": [ "### Prediction using unsupervised machine learning (Task 2)\n", "\n", "K-Means Clustering is used to create different clusters from given data." ] }, { "cell\_type": "code", "execution\_count": 1, "metadata": { "colab": { "base\_uri": "https://localhost:8080/", "height": 195 }, "colab\_type": "code", "executionInfo": { "elapsed": 835, "status": "ok", "timestamp": 1544240454142, "user": { "displayName": "A M Aditya", "photoUrl": "https://lh3.googleusercontent.com/-WI8p7JNWLic/AAAAAAAAAAI/AAAAAAAAAfs/vS8ElgH0p0c/s64/photo.jpg", "userId": "15341571102300750919" }, "user\_tz": -480 }, "id": "kO\_1kOEGDTws", "outputId": "24bb4b2a-2342-4c91-c702-b127a5391152" }, "outputs": [], "source": [ "# Importing the libraries\n", "import numpy as np\n", "import matplotlib.pyplot as plt\n", "import pandas as pd\n", "from sklearn import datasets\n" ] }, { "cell\_type": "code", "execution\_count": 2, "metadata": {}, "outputs": [ { "data": { "text/html": [ "<div>\n", "<style scoped>\n", " .dataframe tbody tr th:only-of-type {\n", " vertical-align: middle;\n", " }\n", "\n", " .dataframe tbody tr th {\n", " vertical-align: top;\n", " }\n", "\n", " .dataframe thead th {\n", " text-align: right;\n", " }\n", "</style>\n", "<table border=\"1\" class=\"dataframe\">\n", " <thead>\n", " <tr style=\"text-align: right;\">\n", " <th></th>\n", " <th>sepal length (cm)</th>\n", " <th>sepal width (cm)</th>\n", " <th>petal length (cm)</th>\n", " <th>petal width (cm)</th>\n", " </tr>\n", " </thead>\n", " <tbody>\n", " <tr>\n", " <th>0</th>\n", " <td>5.1</td>\n", " <td>3.5</td>\n", " <td>1.4</td>\n", " <td>0.2</td>\n", " </tr>\n", " <tr>\n", " <th>1</th>\n", " <td>4.9</td>\n", " <td>3.0</td>\n", " <td>1.4</td>\n", " <td>0.2</td>\n", " </tr>\n", " <tr>\n", " <th>2</th>\n", " <td>4.7</td>\n", " <td>3.2</td>\n", " <td>1.3</td>\n", " <td>0.2</td>\n", " </tr>\n", " <tr>\n", " <th>3</th>\n", " <td>4.6</td>\n", " <td>3.1</td>\n", " <td>1.5</td>\n", " <td>0.2</td>\n", " </tr>\n", " <tr>\n", " <th>4</th>\n", " <td>5.0</td>\n", " <td>3.6</td>\n", " <td>1.4</td>\n", " <td>0.2</td>\n", " </tr>\n", " </tbody>\n", "</table>\n", "</div>" ], "text/plain": [ " sepal length (cm) sepal width (cm) petal length (cm) petal width (cm)\n", "0 5.1 3.5 1.4 0.2\n", "1 4.9 3.0 1.4 0.2\n", "2 4.7 3.2 1.3 0.2\n", "3 4.6 3.1 1.5 0.2\n", "4 5.0 3.6 1.4 0.2" ] }, "execution\_count": 2, "metadata": {}, "output\_type": "execute\_result" } ], "source": [ "# Load the iris dataset\n", "iris = datasets.load\_iris()\n", "iris\_df = pd.DataFrame(iris.data, columns = iris.feature\_names)\n", "iris\_df.head() # See the first 5 rows" ] }, { "cell\_type": "markdown", "metadata": { "colab\_type": "text", "id": "q\_pPmK9GIKMz" }, "source": [ "#### How do you find the optimum number of clusters for K Means? How does one determine the value of K?" ] }, { "cell\_type": "code", "execution\_count": 6, "metadata": { "colab": { "base\_uri": "https://localhost:8080/", "height": 376 }, "colab\_type": "code", "executionInfo": { "elapsed": 1649, "status": "ok", "timestamp": 1544110062756, "user": { "displayName": "A M Aditya", "photoUrl": "https://lh3.googleusercontent.com/-WI8p7JNWLic/AAAAAAAAAAI/AAAAAAAAAfs/vS8ElgH0p0c/s64/photo.jpg", "userId": "15341571102300750919" }, "user\_tz": -480 }, "id": "WevSKogFEalU", "outputId": "d0161248-67f2-48a4-cf4b-224f8faae045" }, "outputs": [ { "data": { "image/png": "\n", "text/plain": [ "<Figure size 432x288 with 1 Axes>" ] }, "metadata": { "needs\_background": "light" }, "output\_type": "display\_data" } ], "source": [ "# Finding the optimum number of clusters for k-means classification\n", "\n", "x = iris\_df.iloc[:, [0, 1, 2, 3]].values\n", "\n", "from sklearn.cluster import KMeans\n", "wcss = []\n", "\n", "for i in range(1, 11):\n", " kmeans = KMeans(n\_clusters = i, init = 'k-means++', \n", " max\_iter = 300, n\_init = 10, random\_state = 0)\n", " kmeans.fit(x)\n", " wcss.append(kmeans.inertia\_)\n", " \n", "# Plotting the results onto a line graph, \n", "# `allowing us to observe 'The elbow'\n", "plt.plot(range(1, 11), wcss)\n", "plt.title('The elbow method')\n", "plt.xlabel('Number of clusters')\n", "plt.ylabel('WCSS') # Within cluster sum of squares\n", "plt.show()" ] }, { "cell\_type": "markdown", "metadata": { "colab\_type": "text", "id": "IUXmLTh4Ih6r" }, "source": [ "The optimum clusters is where the elbow occurs. This is when the within cluster sum of squares (WCSS) doesn't decrease significantly with every iteration.\n", "\n", "From this we choose the number of clusters as \*\* '3\*\*'." ] }, { "cell\_type": "code", "execution\_count": 7, "metadata": { "colab": {}, "colab\_type": "code", "id": "aJbyXuNGIXI9" }, "outputs": [], "source": [ "# Applying kmeans to the dataset / Creating the kmeans classifier\n", "kmeans = KMeans(n\_clusters = 3, init = 'k-means++',\n", " max\_iter = 300, n\_init = 10, random\_state = 0)\n", "y\_kmeans = kmeans.fit\_predict(x)" ] }, { "cell\_type": "code", "execution\_count": 8, "metadata": { "colab": { "base\_uri": "https://localhost:8080/", "height": 364 }, "colab\_type": "code", "executionInfo": { "elapsed": 670, "status": "ok", "timestamp": 1544110172140, "user": { "displayName": "A M Aditya", "photoUrl": "https://lh3.googleusercontent.com/-WI8p7JNWLic/AAAAAAAAAAI/AAAAAAAAAfs/vS8ElgH0p0c/s64/photo.jpg", "userId": "15341571102300750919" }, "user\_tz": -480 }, "id": "Q42-XPJjIyXv", "outputId": "12284613-40c5-41c6-93ba-6b66fae0aa5f" }, "outputs": [ { "data": { "text/plain": [ "<matplotlib.legend.Legend at 0x1c036d77708>" ] }, "execution\_count": 8, "metadata": {}, "output\_type": "execute\_result" }, { "data": { "image/png": "\n", "text/plain": [ "<Figure size 432x288 with 1 Axes>" ] }, "metadata": { "needs\_background": "light" }, "output\_type": "display\_data" } ], "source": [ "# Visualising the clusters - On the first two columns\n", "plt.scatter(x[y\_kmeans == 0, 0], x[y\_kmeans == 0, 1], \n", " s = 100, c = 'red', label = 'Iris-setosa')\n", "plt.scatter(x[y\_kmeans == 1, 0], x[y\_kmeans == 1, 1], \n", " s = 100, c = 'blue', label = 'Iris-versicolour')\n", "plt.scatter(x[y\_kmeans == 2, 0], x[y\_kmeans == 2, 1],\n", " s = 100, c = 'green', label = 'Iris-virginica')\n", "\n", "# Plotting the centroids of the clusters\n", "plt.scatter(kmeans.cluster\_centers\_[:, 0], kmeans.cluster\_centers\_[:,1], \n", " s = 100, c = 'yellow', label = 'Centroids')\n", "\n", "plt.legend()" ] }, { "cell\_type": "code", "execution\_count": null, "metadata": {}, "outputs": [], "source": [] } ], "metadata": { "colab": { "collapsed\_sections": [], "name": "KMeans\_Clustering.ipynb", "provenance": [], "version": "0.3.2" }, "kernelspec": { "display\_name": "Python 3", "language": "python", "name": "python3" }, "language\_info": { "codemirror\_mode": { "name": "ipython", "version": 3 }, "file\_extension": ".py", "mimetype": "text/x-python", "name": "python", "nbconvert\_exporter": "python", "pygments\_lexer": "ipython3", "version": "3.7.6" } }, "nbformat": 4, "nbformat\_minor": 1 }