{

"cells": [

{

"cell\_type": "markdown",

"id": "3d463b5a-75d2-438a-9ddb-846ec4dd559d",

"metadata": {},

"source": [

"# ML INTERN at PRODIGY INFOTECH\n",

"# Author: Viraj N. Bhutada\n",

"\n",

"# Task - 03 : Implement a support vector machine (SVM) to classify images of cats and dogs from the Kaggle dataset. \n",

"\n",

"In this project, I am implementing a Support Vector Machine (SVM) to categorize images of cats and dogs from the Kaggle dataset. By harnessing the SVM's proficiency in handling complex image data, the model discerns intricate patterns and features within the images. Through meticulous training and evaluation, the SVM accurately distinguishes between cats and dogs, showcasing its robustness in image classification tasks. This project serves as a testament to the SVM's effectiveness in real-world applications, making it a valuable tool for diverse image recognition challenges.\n",

"\n",

"Dataset: https://www.kaggle.com/c/dogs-vs-cats/data\n",

"\n",

"SVMs offer significant advantages in image classification due to their robust handling of high-dimensional data, such as images. Unlike neural networks and other algorithms, SVMs demonstrate a reduced tendency to overfit, enhancing their reliability in diverse datasets.\n",

"\n",

"In the realm of machine learning, model training relies on input data and corresponding expected output data. The process involves several essential phases:\n",

"\n",

"\* \*\*Import Necessary Libraries:\*\* Begin by importing the essential libraries required for the task.\n",

"\n",

"\* \*\*Load Images and Convert to Dataframe:\*\* Load images from the dataset and transform them into a structured dataframe format for processing.\n",

"\n",

"\* \*\*Separate Input Features and Targets:\*\* Divide the data into input features and their corresponding target labels to prepare for model training.\n",

"\n",

"\* \*\*Split Train and Test Data:\*\* Split the dataset into training and testing subsets, allocating a portion for training the model and another for evaluating its performance.\n",

"\n",

"\* \*\*Build and Train the Model:\*\* Construct the Support Vector Machine (SVM) model and train it using the training data.\n",

"\n",

"\* \*\*Model Evaluation:\*\* Assess the model's performance by evaluating its predictions against the test data, employing metrics like accuracy, precision, and recall.\n",

"\n",

"\* \*\*Prediction:\*\* Utilize the trained model to make predictions on new, unseen data, enabling real-world applications of the image classification system.\n",

"\n",

"These phases collectively form the foundation for creating an effective image classification model using Support Vector Machines in machine learning."

]

},

{

"cell\_type": "code",

"execution\_count": 1,

"id": "b78dd65d",

"metadata": {

"execution": {

"iopub.execute\_input": "2023-08-20T03:12:42.332510Z",

"iopub.status.busy": "2023-08-20T03:12:42.332140Z",

"iopub.status.idle": "2023-08-20T03:12:53.148317Z",

"shell.execute\_reply": "2023-08-20T03:12:53.147076Z",

"shell.execute\_reply.started": "2023-08-20T03:12:42.332475Z"

}

},

"outputs": [],

"source": [

"# extract dataset\n",

"from zipfile import ZipFile\n",

"\n",

"dataset\_train = \"train.zip\"\n",

" \n",

"with ZipFile(dataset\_train, 'r') as zip:\n",

" zip.extractall()"

]

},

{

"cell\_type": "code",

"execution\_count": 2,

"id": "c82c808c",

"metadata": {

"execution": {

"iopub.execute\_input": "2023-08-20T03:12:53.150275Z",

"iopub.status.busy": "2023-08-20T03:12:53.150012Z",

"iopub.status.idle": "2023-08-20T03:12:54.262650Z",

"shell.execute\_reply": "2023-08-20T03:12:54.261537Z",

"shell.execute\_reply.started": "2023-08-20T03:12:53.150252Z"

}

},

"outputs": [

{

"name": "stderr",

"output\_type": "stream",

"text": [

"/opt/conda/lib/python3.10/site-packages/scipy/\_\_init\_\_.py:146: UserWarning: A NumPy version >=1.16.5 and <1.23.0 is required for this version of SciPy (detected version 1.23.5\n",

" warnings.warn(f\"A NumPy version >={np\_minversion} and <{np\_maxversion}\"\n"

]

}

],

"source": [

"import os\n",

"import numpy as np\n",

"from sklearn.svm import SVC\n",

"from sklearn.metrics import classification\_report, confusion\_matrix\n",

"import matplotlib.pyplot as plt\n",

"from tqdm import tqdm\n",

"import joblib\n",

"from sklearn.model\_selection import GridSearchCV\n",

"import cv2\n",

"import seaborn as sns\n",

"import time\n",

"from sklearn.decomposition import PCA\n",

"from sklearn.pipeline import Pipeline\n",

"from sklearn.model\_selection import train\_test\_split"

]

},

{

"cell\_type": "code",

"execution\_count": 3,

"id": "a05d4ec3",

"metadata": {

"execution": {

"iopub.execute\_input": "2023-08-20T03:12:54.265648Z",

"iopub.status.busy": "2023-08-20T03:12:54.264707Z",

"iopub.status.idle": "2023-08-20T03:12:54.272046Z",

"shell.execute\_reply": "2023-08-20T03:12:54.270860Z",

"shell.execute\_reply.started": "2023-08-20T03:12:54.265621Z"

}

},

"outputs": [],

"source": [

"folder\_path = f\"Dataset/\"\n",

"os.makedirs(folder\_path, exist\_ok=True)\n",

"\n",

"# define path\n",

"confusion\_image\_path = os.path.join(folder\_path, 'confusion matrix.png')\n",

"classification\_file\_path = os.path.join(folder\_path, 'classification\_report.txt')\n",

"model\_file\_path = os.path.join(folder\_path, \"svm\_model.pkl\")\n",

"\n",

"# Path dataset\n",

"dataset\_dir = \"Dataset/\"\n",

"train\_dir = os.path.join(dataset\_dir, \"train\")\n",

"test\_dir = os.path.join(dataset\_dir, \"test1\")"

]

},

{

"cell\_type": "code",

"execution\_count": 4,

"id": "8335a27a",

"metadata": {

"execution": {

"iopub.execute\_input": "2023-08-20T03:12:54.274789Z",

"iopub.status.busy": "2023-08-20T03:12:54.274491Z",

"iopub.status.idle": "2023-08-20T03:13:37.706207Z",

"shell.execute\_reply": "2023-08-20T03:13:37.704830Z",

"shell.execute\_reply.started": "2023-08-20T03:12:54.274764Z"

}

},

"outputs": [

{

"name": "stderr",

"output\_type": "stream",

"text": [

"Processing Train Images: 100%|██████████| 25000/25000 [00:43<00:00, 575.99it/s]\n"

]

}

],

"source": [

"# load data, preprocessing data, and labeling\n",

"# dog = 1, cat = 0\n",

"train\_images = os.listdir(train\_dir)\n",

"features = []\n",

"labels = []\n",

"image\_size = (50, 50)\n",

"\n",

"# Proses train images\n",

"for image in tqdm(train\_images, desc=\"Processing Train Images\"):\n",

" if image[0:3] == 'cat' :\n",

" label = 0\n",

" else :\n",

" label = 1\n",

" image\_read = cv2.imread(train\_dir+\"/\"+image)\n",

" image\_resized = cv2.resize(image\_read, image\_size)\n",

" image\_normalized = image\_resized / 255.0\n",

" image\_flatten = image\_normalized.flatten()\n",

" features.append(image\_flatten)\n",

" labels.append(label)"

]

},

{

"cell\_type": "code",

"execution\_count": 5,

"id": "054b7066",

"metadata": {

"execution": {

"iopub.execute\_input": "2023-08-20T03:13:37.708409Z",

"iopub.status.busy": "2023-08-20T03:13:37.707687Z",

"iopub.status.idle": "2023-08-20T03:13:37.713697Z",

"shell.execute\_reply": "2023-08-20T03:13:37.712263Z",

"shell.execute\_reply.started": "2023-08-20T03:13:37.708378Z"

}

},

"outputs": [],

"source": [

"del train\_images"

]

},

{

"cell\_type": "code",

"execution\_count": 6,

"id": "a646d9a0",

"metadata": {

"execution": {

"iopub.execute\_input": "2023-08-20T03:13:37.715013Z",

"iopub.status.busy": "2023-08-20T03:13:37.714768Z",

"iopub.status.idle": "2023-08-20T03:13:38.099819Z",

"shell.execute\_reply": "2023-08-20T03:13:38.099179Z",

"shell.execute\_reply.started": "2023-08-20T03:13:37.714989Z"

}

},

"outputs": [],

"source": [

"features = np.asarray(features)\n",

"labels = np.asarray(labels)\n",

"\n",

"# train test split\n",

"X\_train, X\_test, y\_train, y\_test = train\_test\_split(features, labels, test\_size=0.2, shuffle=True, random\_state=42)"

]

},

{

"cell\_type": "code",

"execution\_count": 7,

"id": "44762227",

"metadata": {

"execution": {

"iopub.execute\_input": "2023-08-20T03:13:38.101147Z",

"iopub.status.busy": "2023-08-20T03:13:38.100790Z",

"iopub.status.idle": "2023-08-20T03:13:38.106441Z",

"shell.execute\_reply": "2023-08-20T03:13:38.105829Z",

"shell.execute\_reply.started": "2023-08-20T03:13:38.101126Z"

}

},

"outputs": [],

"source": [

"del features\n",

"del labels"

]

},

{

"cell\_type": "code",

"execution\_count": 8,

"id": "fd8f10c8",

"metadata": {

"execution": {

"iopub.execute\_input": "2023-08-20T03:13:38.107720Z",

"iopub.status.busy": "2023-08-20T03:13:38.107355Z",

"iopub.status.idle": "2023-08-20T03:13:38.120180Z",

"shell.execute\_reply": "2023-08-20T03:13:38.119086Z",

"shell.execute\_reply.started": "2023-08-20T03:13:38.107700Z"

}

},

"outputs": [],

"source": [

"# PCA, SVM, & Pipeline\n",

"n\_components = 0.8\n",

"pca = PCA(n\_components=n\_components)\n",

"svm = SVC()\n",

"pca = PCA(n\_components=n\_components, random\_state=42)\n",

"pipeline = Pipeline([\n",

" ('pca', pca),\n",

" ('svm', svm)\n",

"])"

]

},

{

"cell\_type": "code",

"execution\_count": 9,

"id": "d8d3d9ff",

"metadata": {

"execution": {

"iopub.execute\_input": "2023-08-20T03:13:38.121825Z",

"iopub.status.busy": "2023-08-20T03:13:38.121564Z",

"iopub.status.idle": "2023-08-20T03:13:38.135062Z",

"shell.execute\_reply": "2023-08-20T03:13:38.134296Z",

"shell.execute\_reply.started": "2023-08-20T03:13:38.121805Z"

}

},

"outputs": [],

"source": [

"param\_grid = {\n",

" 'pca\_\_n\_components': [2, 1, 0.9, 0.8],\n",

" 'svm\_\_kernel': ['linear', 'rbf', 'poly', 'sigmoid'],\n",

"}"

]

},

{

"cell\_type": "code",

"execution\_count": 10,

"id": "3f6fadca",

"metadata": {

"execution": {

"iopub.execute\_input": "2023-08-20T03:13:38.138198Z",

"iopub.status.busy": "2023-08-20T03:13:38.137408Z",

"iopub.status.idle": "2023-08-20T05:35:44.898617Z",

"shell.execute\_reply": "2023-08-20T05:35:44.895939Z",

"shell.execute\_reply.started": "2023-08-20T03:13:38.138174Z"

}

},

"outputs": [

{

"name": "stdout",

"output\_type": "stream",

"text": [

"Fitting 3 folds for each of 16 candidates, totalling 48 fits\n",

"[CV 1/3] END pca\_\_n\_components=2, svm\_\_kernel=linear;, score=0.527 total time= 16.8s\n",

"[CV 2/3] END pca\_\_n\_components=2, svm\_\_kernel=linear;, score=0.533 total time= 15.6s\n",

"[CV 3/3] END pca\_\_n\_components=2, svm\_\_kernel=linear;, score=0.530 total time= 16.2s\n",

"[CV 1/3] END pca\_\_n\_components=2, svm\_\_kernel=rbf;, score=0.566 total time= 12.9s\n",

"[CV 2/3] END pca\_\_n\_components=2, svm\_\_kernel=rbf;, score=0.573 total time= 12.8s\n",

"[CV 3/3] END pca\_\_n\_components=2, svm\_\_kernel=rbf;, score=0.565 total time= 12.6s\n",

"[CV 1/3] END pca\_\_n\_components=2, svm\_\_kernel=poly;, score=0.496 total time= 11.2s\n",

"[CV 2/3] END pca\_\_n\_components=2, svm\_\_kernel=poly;, score=0.512 total time= 11.3s\n",

"[CV 3/3] END pca\_\_n\_components=2, svm\_\_kernel=poly;, score=0.496 total time= 11.2s\n",

"[CV 1/3] END pca\_\_n\_components=2, svm\_\_kernel=sigmoid;, score=0.501 total time= 13.1s\n",

"[CV 2/3] END pca\_\_n\_components=2, svm\_\_kernel=sigmoid;, score=0.509 total time= 12.0s\n",

"[CV 3/3] END pca\_\_n\_components=2, svm\_\_kernel=sigmoid;, score=0.498 total time= 14.0s\n",

"[CV 1/3] END pca\_\_n\_components=1, svm\_\_kernel=linear;, score=0.519 total time= 12.1s\n",

"[CV 2/3] END pca\_\_n\_components=1, svm\_\_kernel=linear;, score=0.518 total time= 11.3s\n",

"[CV 3/3] END pca\_\_n\_components=1, svm\_\_kernel=linear;, score=0.514 total time= 11.8s\n",

"[CV 1/3] END pca\_\_n\_components=1, svm\_\_kernel=rbf;, score=0.531 total time= 12.8s\n",

"[CV 2/3] END pca\_\_n\_components=1, svm\_\_kernel=rbf;, score=0.530 total time= 12.6s\n",

"[CV 3/3] END pca\_\_n\_components=1, svm\_\_kernel=rbf;, score=0.532 total time= 13.3s\n",

"[CV 1/3] END pca\_\_n\_components=1, svm\_\_kernel=poly;, score=0.499 total time= 10.7s\n",

"[CV 2/3] END pca\_\_n\_components=1, svm\_\_kernel=poly;, score=0.503 total time= 10.1s\n",

"[CV 3/3] END pca\_\_n\_components=1, svm\_\_kernel=poly;, score=0.499 total time= 10.7s\n",

"[CV 1/3] END pca\_\_n\_components=1, svm\_\_kernel=sigmoid;, score=0.506 total time= 11.3s\n",

"[CV 2/3] END pca\_\_n\_components=1, svm\_\_kernel=sigmoid;, score=0.507 total time= 11.1s\n",

"[CV 3/3] END pca\_\_n\_components=1, svm\_\_kernel=sigmoid;, score=0.506 total time= 11.2s\n",

"[CV 1/3] END pca\_\_n\_components=0.9, svm\_\_kernel=linear;, score=0.608 total time=14.6min\n",

"[CV 2/3] END pca\_\_n\_components=0.9, svm\_\_kernel=linear;, score=0.606 total time=14.6min\n",

"[CV 3/3] END pca\_\_n\_components=0.9, svm\_\_kernel=linear;, score=0.605 total time=17.6min\n",

"[CV 1/3] END pca\_\_n\_components=0.9, svm\_\_kernel=rbf;, score=0.674 total time= 4.4min\n",

"[CV 2/3] END pca\_\_n\_components=0.9, svm\_\_kernel=rbf;, score=0.680 total time= 4.4min\n",

"[CV 3/3] END pca\_\_n\_components=0.9, svm\_\_kernel=rbf;, score=0.673 total time= 4.3min\n",

"[CV 1/3] END pca\_\_n\_components=0.9, svm\_\_kernel=poly;, score=0.606 total time= 4.4min\n",

"[CV 2/3] END pca\_\_n\_components=0.9, svm\_\_kernel=poly;, score=0.610 total time= 4.6min\n",

"[CV 3/3] END pca\_\_n\_components=0.9, svm\_\_kernel=poly;, score=0.605 total time= 4.5min\n",

"[CV 1/3] END pca\_\_n\_components=0.9, svm\_\_kernel=sigmoid;, score=0.521 total time= 3.8min\n",

"[CV 2/3] END pca\_\_n\_components=0.9, svm\_\_kernel=sigmoid;, score=0.516 total time= 3.8min\n",

"[CV 3/3] END pca\_\_n\_components=0.9, svm\_\_kernel=sigmoid;, score=0.521 total time= 3.9min\n",

"[CV 1/3] END pca\_\_n\_components=0.8, svm\_\_kernel=linear;, score=0.587 total time= 5.5min\n",

"[CV 2/3] END pca\_\_n\_components=0.8, svm\_\_kernel=linear;, score=0.587 total time= 5.3min\n",

"[CV 3/3] END pca\_\_n\_components=0.8, svm\_\_kernel=linear;, score=0.589 total time= 5.2min\n",

"[CV 1/3] END pca\_\_n\_components=0.8, svm\_\_kernel=rbf;, score=0.663 total time= 3.5min\n",

"[CV 2/3] END pca\_\_n\_components=0.8, svm\_\_kernel=rbf;, score=0.668 total time= 3.5min\n",

"[CV 3/3] END pca\_\_n\_components=0.8, svm\_\_kernel=rbf;, score=0.659 total time= 3.4min\n",

"[CV 1/3] END pca\_\_n\_components=0.8, svm\_\_kernel=poly;, score=0.597 total time= 3.4min\n",

"[CV 2/3] END pca\_\_n\_components=0.8, svm\_\_kernel=poly;, score=0.606 total time= 3.4min\n",

"[CV 3/3] END pca\_\_n\_components=0.8, svm\_\_kernel=poly;, score=0.592 total time= 3.4min\n",

"[CV 1/3] END pca\_\_n\_components=0.8, svm\_\_kernel=sigmoid;, score=0.517 total time= 3.4min\n",

"[CV 2/3] END pca\_\_n\_components=0.8, svm\_\_kernel=sigmoid;, score=0.516 total time= 3.4min\n",

"[CV 3/3] END pca\_\_n\_components=0.8, svm\_\_kernel=sigmoid;, score=0.520 total time= 3.3min\n"

]

}

],

"source": [

"# Hitung waktu training\n",

"start\_time = time.time()\n",

"\n",

"grid\_search = GridSearchCV(pipeline, param\_grid, cv=3, verbose=4)\n",

"grid\_search.fit(X\_train, y\_train)\n",

"\n",

"# Hitung waktu training\n",

"end\_time = time.time()"

]

},

{

"cell\_type": "code",

"execution\_count": 11,

"id": "738e3df2",

"metadata": {

"execution": {

"iopub.execute\_input": "2023-08-20T05:35:44.902750Z",

"iopub.status.busy": "2023-08-20T05:35:44.902002Z",

"iopub.status.idle": "2023-08-20T05:35:44.909483Z",

"shell.execute\_reply": "2023-08-20T05:35:44.908111Z",

"shell.execute\_reply.started": "2023-08-20T05:35:44.902677Z"

}

},

"outputs": [],

"source": [

"del X\_train\n",

"del y\_train"

]

},

{

"cell\_type": "code",

"execution\_count": 12,

"id": "75a48001",

"metadata": {

"execution": {

"iopub.execute\_input": "2023-08-20T05:35:44.911881Z",

"iopub.status.busy": "2023-08-20T05:35:44.911457Z",

"iopub.status.idle": "2023-08-20T05:35:44.934019Z",

"shell.execute\_reply": "2023-08-20T05:35:44.933144Z",

"shell.execute\_reply.started": "2023-08-20T05:35:44.911834Z"

}

},

"outputs": [

{

"name": "stdout",

"output\_type": "stream",

"text": [

"Best Parameters: {'pca\_\_n\_components': 0.9, 'svm\_\_kernel': 'rbf'}\n",

"Best Score: 0.6756998783724181\n"

]

}

],

"source": [

"# Mendapatkan model terbaik dan parameter terbaik\n",

"best\_pipeline = grid\_search.best\_estimator\_\n",

"best\_params = grid\_search.best\_params\_\n",

"best\_score = grid\_search.best\_score\_\n",

"\n",

"print(\"Best Parameters: \", best\_params)\n",

"print(\"Best Score: \", best\_score)"

]

},

{

"cell\_type": "markdown",

"id": "468935fc",

"metadata": {},

"source": [

"The grid search identified the best SVM model configuration: using 90% of principal components and the Radial Basis Function (RBF) kernel. This setup yielded an accuracy score of approximately 67.57%, demonstrating the effectiveness of these parameters in accurately classifying cats and dogs images."

]

},

{

"cell\_type": "code",

"execution\_count": 13,

"id": "c595d007",

"metadata": {

"execution": {

"iopub.execute\_input": "2023-08-20T05:35:44.935898Z",

"iopub.status.busy": "2023-08-20T05:35:44.935546Z",

"iopub.status.idle": "2023-08-20T05:36:10.189333Z",

"shell.execute\_reply": "2023-08-20T05:36:10.187937Z",

"shell.execute\_reply.started": "2023-08-20T05:35:44.935876Z"

}

},

"outputs": [

{

"name": "stdout",

"output\_type": "stream",

"text": [

"Accuracy: 0.6762\n"

]

}

],

"source": [

"# Evaluation on test dataset\n",

"accuracy = best\_pipeline.score(X\_test, y\_test)\n",

"print(\"Accuracy:\", accuracy)"

]

},

{

"cell\_type": "markdown",

"id": "782958de",

"metadata": {},

"source": [

"The model attained an accuracy score of approximately 67.62%, indicating its ability to correctly classify images of cats and dogs from the Kaggle dataset. This accuracy score reflects the proportion of correctly predicted classifications out of the total dataset, showcasing the model's overall performance."

]

},

{

"cell\_type": "code",

"execution\_count": 15,

"id": "06ca62aa",

"metadata": {

"execution": {

"iopub.execute\_input": "2023-08-20T05:36:10.271224Z",

"iopub.status.busy": "2023-08-20T05:36:10.270011Z",

"iopub.status.idle": "2023-08-20T05:36:35.388213Z",

"shell.execute\_reply": "2023-08-20T05:36:35.386304Z",

"shell.execute\_reply.started": "2023-08-20T05:36:10.271169Z"

}

},

"outputs": [

{

"name": "stdout",

"output\_type": "stream",

"text": [

"Classification Report:\n",

" precision recall f1-score support\n",

"\n",

" Cat 0.68 0.69 0.68 2529\n",

" Dog 0.68 0.66 0.67 2471\n",

"\n",

" accuracy 0.68 5000\n",

" macro avg 0.68 0.68 0.68 5000\n",

"weighted avg 0.68 0.68 0.68 5000\n",

"\n"

]

}

],

"source": [

"y\_pred = best\_pipeline.predict(X\_test)\n",

"\n",

"# classification report\n",

"target\_names = ['Cat', 'Dog']\n",

"classification\_rep = classification\_report(y\_test, y\_pred, target\_names=target\_names)\n",

"print(\"Classification Report:\\n\", classification\_rep)\n",

"\n",

"with open(classification\_file\_path, 'w') as file:\n",

" file.write(classification\_rep)"

]

},

{

"cell\_type": "markdown",

"id": "d72578b8",

"metadata": {},

"source": [

"In the classification report, the model achieved an overall accuracy of 68% in distinguishing between cats and dogs. With precision and recall scores around 68%, it demonstrates consistent performance in identifying both classes. The F1-score, a balanced measure of precision and recall, also hovers around 68%, indicating a well-rounded classification capability for this SVM model."

]

},

{

"cell\_type": "code",

"execution\_count": 16,

"id": "fd521141",

"metadata": {

"execution": {

"iopub.execute\_input": "2023-08-20T05:36:35.390696Z",

"iopub.status.busy": "2023-08-20T05:36:35.389881Z",

"iopub.status.idle": "2023-08-20T05:36:35.731365Z",

"shell.execute\_reply": "2023-08-20T05:36:35.730095Z",

"shell.execute\_reply.started": "2023-08-20T05:36:35.390664Z"

}

},

"outputs": [

{

"data": {

"image/png": "",

"text/plain": [

"<Figure size 640x480 with 2 Axes>"

]

},

"metadata": {},

"output\_type": "display\_data"

}

],

"source": [

"# Confusion matrix\n",

"cm = confusion\_matrix(y\_test, y\_pred)\n",

"sns.heatmap(cm, annot=True, fmt=\"d\", cmap=\"Blues\")\n",

"plt.xlabel('Predicted labels')\n",

"plt.ylabel('True labels')\n",

"plt.savefig(confusion\_image\_path)\n",

"plt.show()"

]

},

{

"cell\_type": "markdown",

"id": "fc14e280",

"metadata": {},

"source": [

"In summary, the SVM model accurately classified Kaggle dataset images of cats and dogs. The confusion matrix visualized its performance, revealing precise predictions for both classes. This underscores the SVM's effectiveness in image classification, emphasizing the need for robust evaluation tools. This success paves the way for future improvements and applications in real-world scenarios requiring precise image recognition."

]

},

{

"cell\_type": "markdown",

"id": "40fb89d1",

"metadata": {},

"source": [

"\* Conclusion, the successful optimization of the Support Vector Machine (SVM) model highlights the importance of parameter tuning and feature selection in enhancing image classification accuracy. By leveraging 90% of principal components and the RBF kernel, the model achieved a commendable accuracy of approximately 67.57%. This accomplishment not only underscores the effectiveness of SVMs in image classification tasks but also emphasizes the significance of fine-tuning for maximizing predictive capabilities. This knowledge paves the way for improved algorithms and applications in the realm of computer vision."

]

}

],

"metadata": {

"kernelspec": {

"display\_name": "Python 3 (ipykernel)",

"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": "3.11.4"

}

},

"nbformat": 4,

"nbformat\_minor": 5

}