Haneen Zamzami

Steganalysis system

Introdution

- Steganalysis is described as "the art and science of detecting secret messages hidden using steganography“, hidden files can be images or text.

- The goal of this project was to use classification models to predict steganographic images(clean\stego) by extracting a number of image features such as correlation, contrast, homogeneity, and energy.

Design

- In this project, I will predict the catogray of images by extract GLCM features (Contrast , Homogeneity , Energy and Correlation),the analysis will be based on 8683 GLCM features information.

- Classifying statuses accurately via machine learning models would enable the for steganalysers and forensic examiners from stop illegal use of hidden data.

Data

- I download the images from different resource Kaggle , pixabay and gettyimages. then I create my own database by extract the Gray Level Co-Occurrence Matrix (GLCM) properties of correlation, contrast, homogeneity, and energy .

- The dataset contains 8683 records with 8 features are: image name, Image size, Image format (JPG- BMP- TIFF), Contrast of image ,Homogeneity of image, Energy of image, Correlation of image and Catogry of Images (clean\stego), 4623 are clean images and 4060 are stego images.

Data

GLCM features

	Image Name	size	Energy	Contrast	Homogeneity	Correlation	format	catogary
0	0.BMP	23878.0	0.158740	0.73517	0.82530	0.77772	BMP	0
1	1.BMP	23878.0	0.055316	1.07540	0.75824	0.85952	BMP	0
2	10.BMP	23878.0	0.071805	1.19140	0.77469	0.85186	BMP	0
3	100.BMP	23878.0	0.084927	1.18790	0.78018	0.84463	BMP	0
4	101.BMP	23878.0	0.083012	0.81284	0.80488	0.90919	BMP	0
...	...	...	...	...	...	...	...	...
495	00000001_(3).JPG	8609.0	0.105520	0.34301	0.84645	0.95096	JPG	1
496	00000001_(4).JPG	7250.0	0.140930	0.30315	0.85757	0.89873	JPG	1
497	00000001_(5).JPG	8744.0	0.112860	0.55186	0.83887	0.85478	JPG	1
498	00000001_(6).JPG	5281.0	0.145690	0.16412	0.94173	0.96577	JPG	1
499	00000001_(7).JPG	7923.0	0.216240	0.46967	0.83989	0.87520	JPG	1

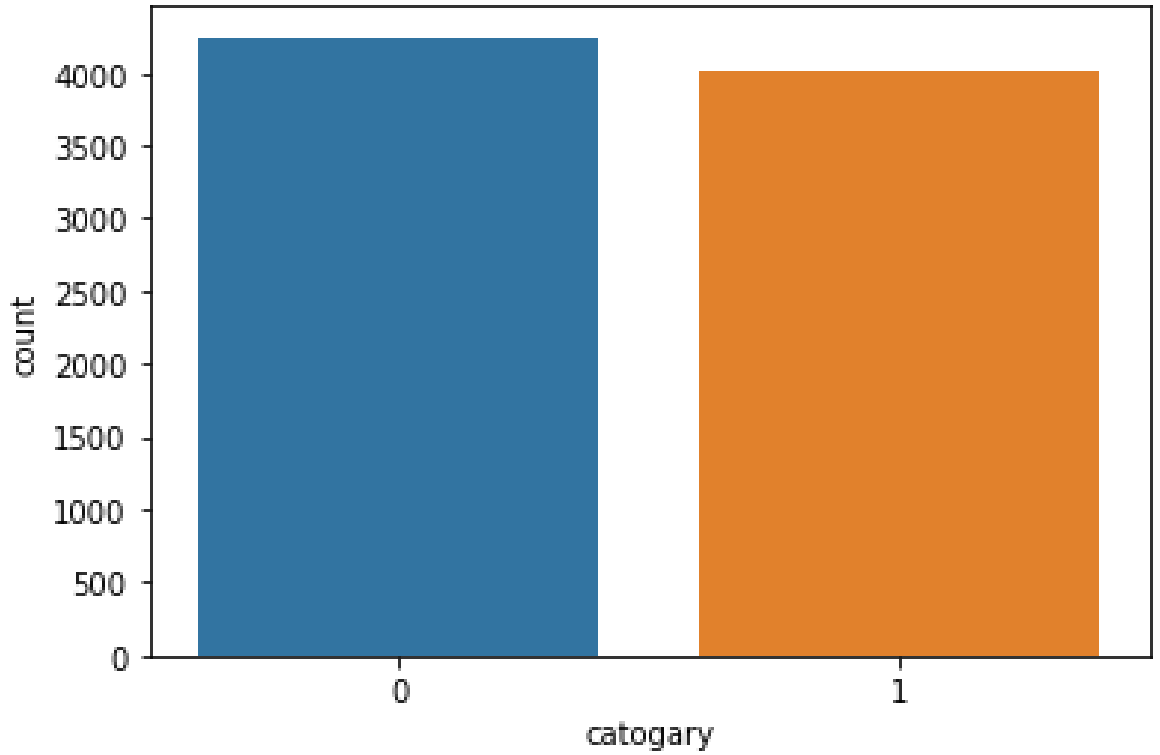
clean

stego

Balanced data

```
clean = len(images[images.catogary == 0])
stego = len(images[images.catogary == 1])
print("Percentage of clean images: {:.2f}%".format((clean / (len(images.catogary))*100)))
print("Percentage of stego images: {:.2f}%".format((stego / (len(images.catogary))*100)))
```

```
Percentage of clean images: 51.36%
Percentage of stego images: 48.64%
```



Algorithms

Support vector machine , k-nearest neighbors, Linear Discriminant Analysis and Decision Tree classifiers were used before settling on Support vector machine as the model with strongest cross-validation performance. Support vector machine feature importance ranking was used directly to guide the choice and order of variables to be included as the model underwent refinement.

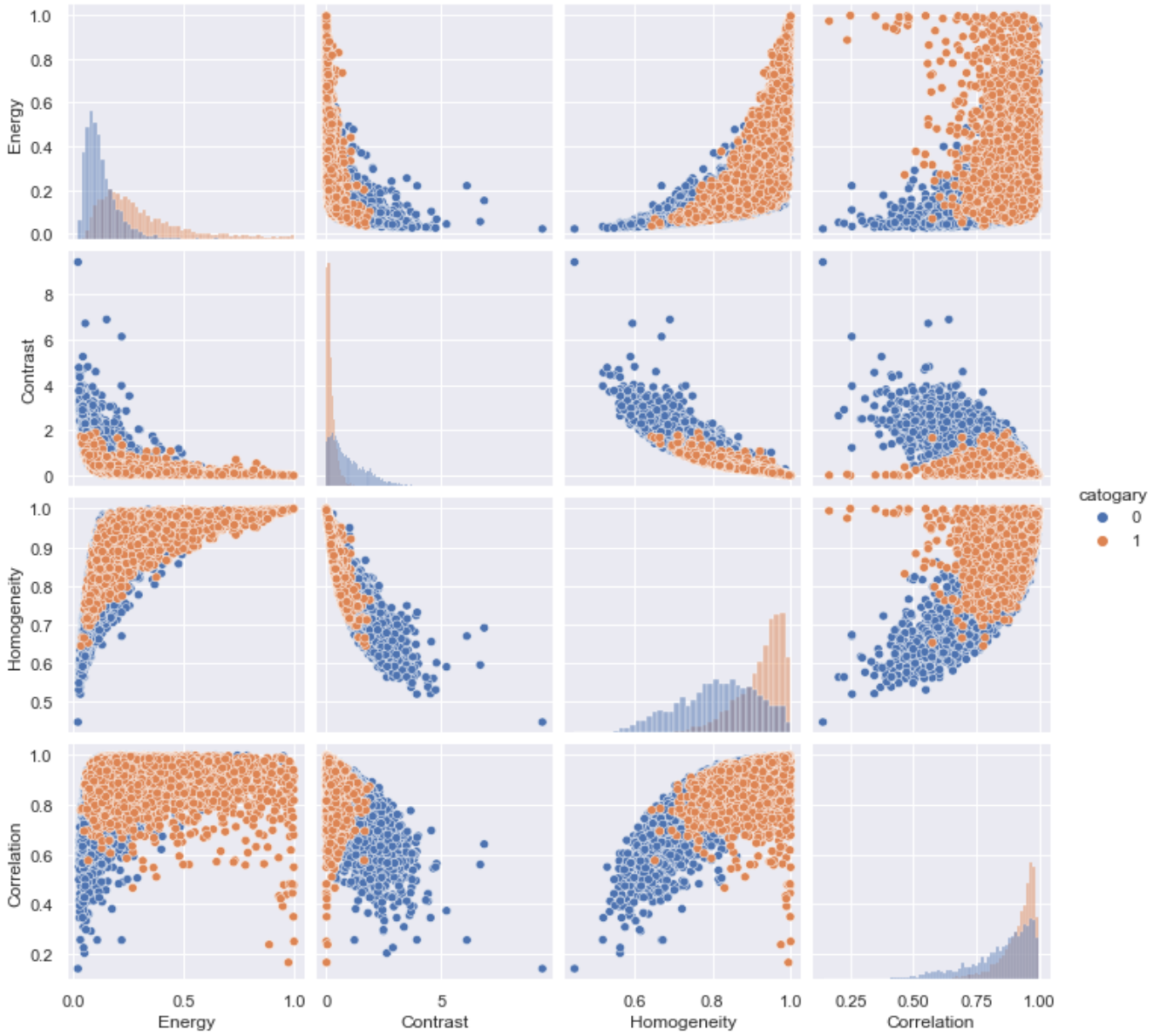
- # 10-fold cross-validation with K=5

Classifier	Tranning set	Testing set
k-nearest neighbors (KNN)	100%	100%
Linear Discriminant Analysi s (LDA)	95.94%	95.62%
Decision Tree	100%	93.98%
Support vector machine (SVM)	95.94%	95.62%

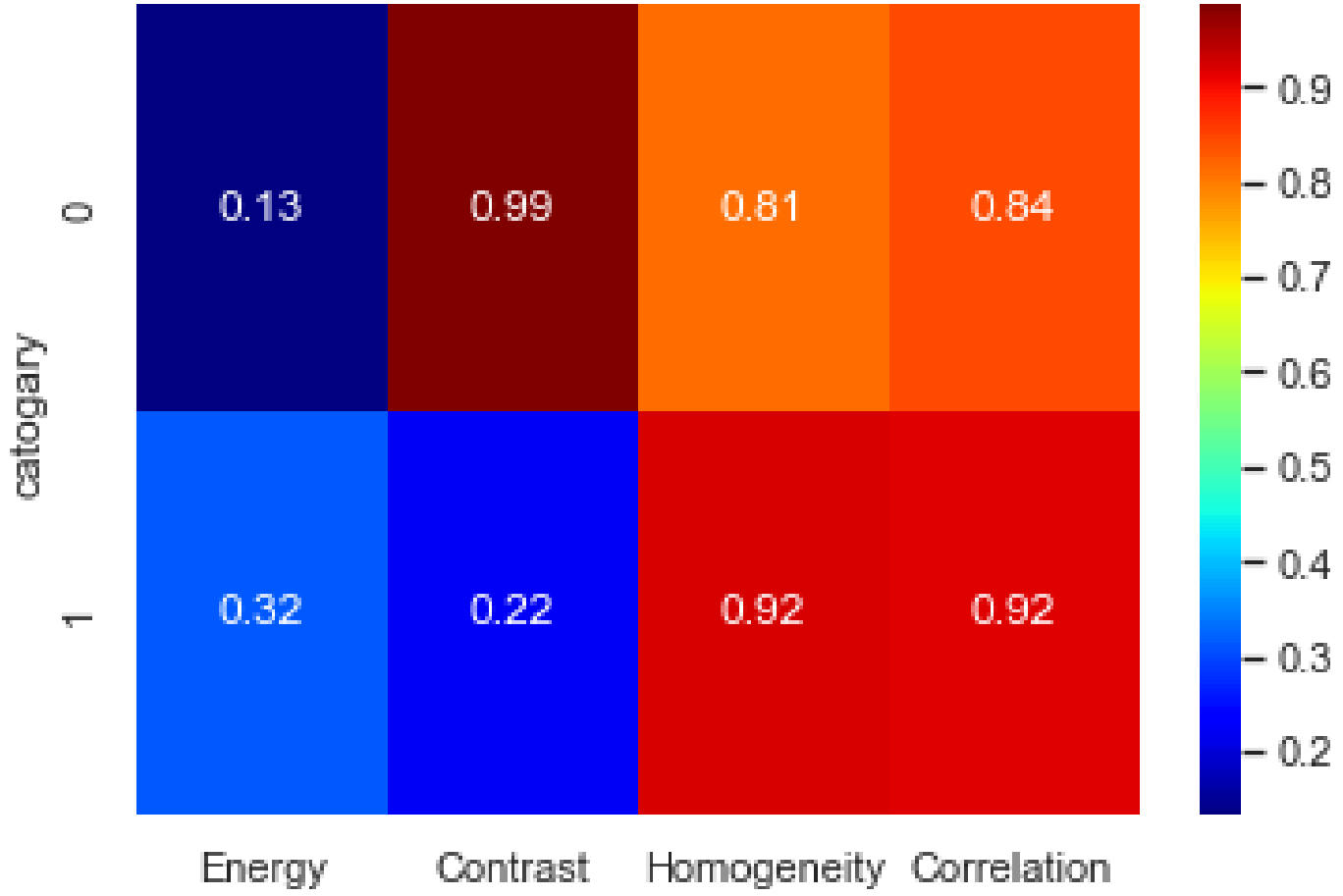
Tools

- Jupyter notebook for coding
- Numpy and Pandas for data manipulation
- Scikit-learn for modeling
- Matplotlib and Seaborn for plotting
- Confusion Matrix for visualizations

Pair polt



Heap mab


catogary
0
1
0.13
0.99
0.81
0.84
0.32
0.22
0.92
0.92
Energy
Contrast
Homogeneity
Correlation
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2

Confusion Matrix