Detecting Volcanoes on Venus via Classification

Team Mamamia!:Di Ye (diye2)[1], Hao Wang (haow2)[1], and Hannah Pu (chpu2)[1]

[1]Department of Statistics, University of Illinois at Urbana-Champaign

LASSO Regression, logistic regression and neural network have been used in this project to identify whether each image in the Kaggle datasets contains volcanoes or not. The analysis of the images revealed that there are several volcanoes on Venus. Neural network has the best performance to identify whether the images have volcanoes in this project for image classification with an area under the curve 0.9629 and a high F-1 score 0.9747. The comparison between models is also included.

Image Classification | LASSO Regression | Neural Network

I. INTRODUCTION. NASA spacecraft was launched off for a mission to Venus on May 4, 1989, the mission mapped surfaces of Venus and image recorded most 98% of Venus surface. The analysis of the images revealed that there are several volcanoes on Venus. This project aims to construct prediction model to identify whether a image contains volcanoes or not.

Data Source Information. The data was downloaded from Kaggle, which is originally from NASA's Magellan spacecraft database. (https://www.kaggle.com/amantheroot/finding-volcanoes-on-venus/data)

Data Description. 9734 images were captured by the spacecraft and converted to pixels (110 x 110, from 0 to 255), where every image is one row of 12100 columns (all the 110 rows of 110 columns). Images can contain more than one volcanoes or maybe none. The 9000+ images are separated to four datasets (file names : *train_images*, *train_labels*, *test_images*, and *test_labels*):

Image Dataset (*train_images* and *test_images*) *Train_images* : 7000 images as train data with 12100 variables; *Test_images* : 2734 images as test data with 12100 variables; All the variables (V1 to V12100) correspond to the pixel image, 110 pixels * 110 pixels = 12100 pixels.

Label Dataset (*train_labels* and *test_labels*) A summary of the variables in both *train_labels* and *test_label* datasets is listed down below:

1. *Volcano?* : If in the image there exists at least one volcano, the label is 1 (Yes). Otherwise, the label is 0 (No). (If Volcano? equals 0, the following three categories would be "NaN"). 2. *Type* : 1 = definitely a volcano, 2 = probably, 3 = possibly, 4 = only a pit is visible

3. *Radius* : Is the radius of the volcano in the center of the image, in pixels?

4. *Number Volcanoes* : The number of volcanoes in the image.

Literature Review. In Kaggle, there were 11 kernels regarding data analysis of this dataset, all using Python. The kernels included vivid data visualization and exploratory data. Most kernels focused on classification prediction on whether there are volcanoes or not. For this classification prediction, different methods have been used to construct prediction model, from simple models such as logistic regression, to advanced models such as Convolutional Neural Network (CNN) and VGG Neural Network for deep learning. Accuracy of models ranged from 84.1% to 97%.

Scientific Goal. For this project, we focus mainly on doing classification predition on whether each image has a volcano or not. In addition, if the classification prediction goes well, we will also construct models to predict the number of volcanoes in the images. We aim in constructing different classification models and choosing the best model to predict whether there exists a valcano in each image. Identifying valcanoe through IT technology would increase the efficency of space exploration and safty of the crews.

II. EXPLORATORY DATA. The first 6 observations of *train_labels*

```
#    Volcano. Type Radius Number.Volcanoes
# 1         1    3  17.46                1
# 2         0  NaN    NaN              NaN
# 3         0  NaN    NaN              NaN
# 4         0  NaN    NaN              NaN
# 5         0  NaN    NaN              NaN
# 6         0  NaN    NaN              NaN
```

The first 6 observations of *test_labels*

```
#     Volcano. Type Radius Number.Volcanoes
# 1        0  NaN    NaN               NaN
# 2        0  NaN    NaN               NaN
# 3        1    1  17.00                 1
# 4        0  NaN    NaN               NaN
# 5        1    3  15.13                 1
# 6        0  NaN    NaN               NaN
```


Fig. 2. Obs 39 Type: 2 Radius: 19.31 Number Volcanoes: 1

After exploring the datasets, we found only labels have NaNs. We have set the those values to 0, an insignicant value in our dataset.

Data Visualization. 6 observations are picked to demonstrate how the images got labeled. Figure 1 - 4 show how well the volcanoes can be seen from the images (*Type* : 1 = definitely a volcano, 2 = probably, 3 = possibly, 4 = only a pit is visible). Figure 5 contains 2 volcanoes, and Figure 6 contains no volcano at all. We can tell from the images that a bright white dot indicates a potential volcano, while the white dot might not be clear enough to see in the image, which means that it is hard to identify whether there is a volcano or not.
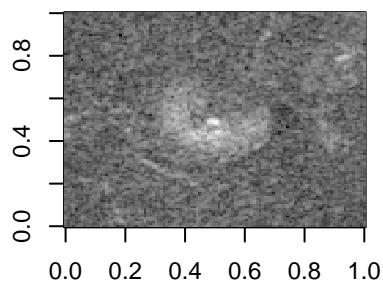

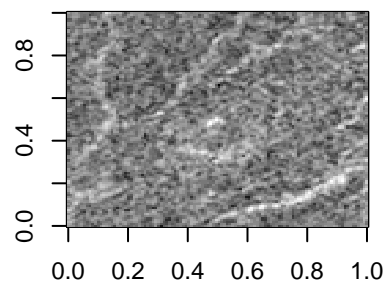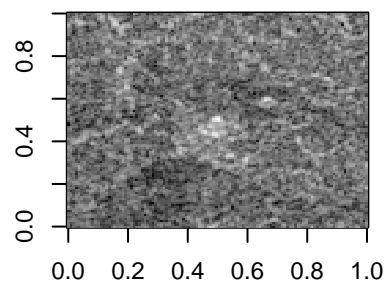**Fig. 3.** Obs 1 Type: 3 Radius: 17.46 Number of Volcanoes: 1

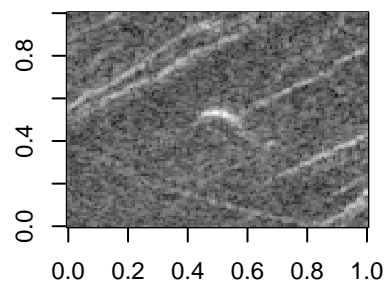**Fig. 1.** Obs 10 Type: 1 Radius: 22.02 Number Volcanoes: 1

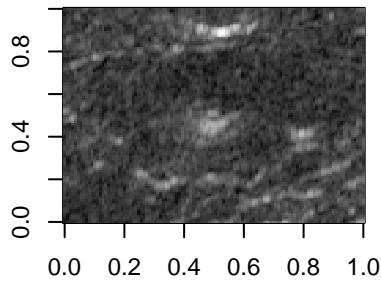**Fig. 4.** Obs 30 Type: 4 Radius: 6.40 Number Volcanoes: 1

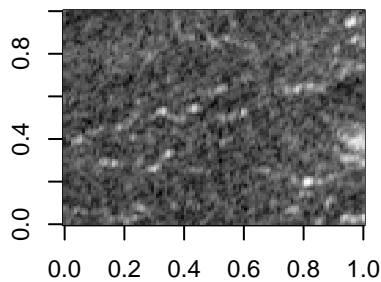**Fig. 5.** Obs 289 Type: 1 Radius: 11.05 Number Volcanoes: 2

Fig. 6. Obs 2 No Volcano

Fig. 7. How many images have volcanoes?

Fig. 8. How clealy can we identify the volcanoes?

Data Summary. To facilitate the analysis process and explore the data, we have summarized the data. The `ggplot` has been used to visualize the training set labels. Figure 7 shows that only 1000 images in the training dataset have volcanoes. Figure 8 shows that within 1000 images that have volcanoes, how clearly we can identify the volcanoes. Figure 9 shows how many volcanoes in each image. The number of volcanoes in the images ranges from 0 to 5 in the training set.

III. METHODOLOGY.

Lasso Regression. To apply Lasso Regression, we first selected reasonable interval for tuning parameter λ by ploting cross-validation error, which gives us the interval (-6.5, -5). Afterthat, we did the model selection by
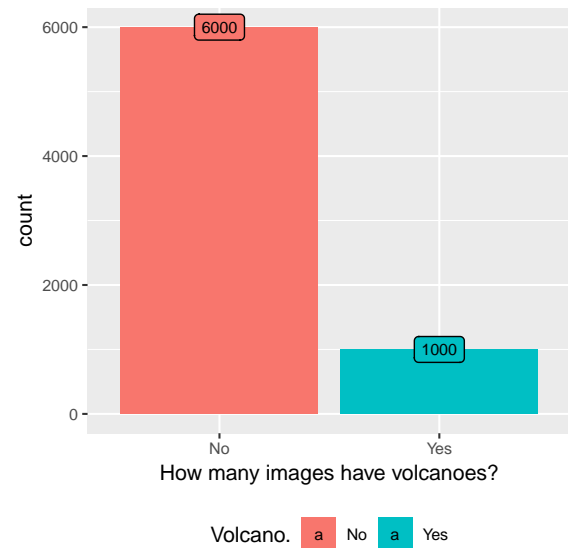
Fig. 9. How clealy can we identify the volcanoes?
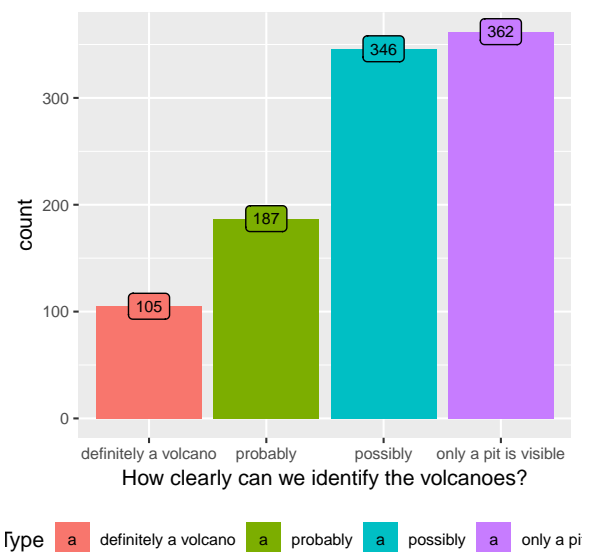
The model, similar to the first one, used 20% of the training data as validation data.

Fig. 10. Neural Network to classify whether an image contains volcanoes

	No	Yes
No	2270	88
Yes	30	346

The loss function and accuracy:

```
#  $loss
#  [1] 0.3018637
#
#  $acc
#  [1] 0.9568398
```


Fig. 11. Neural Network to classify how many volcanoes in each image

	0	1	2	3
0	2272	67	15	2
1	23	266	44	7
2	5	26	4	3

The loss function and accuracy:

```
#  $loss
#  [1] 0.554538
#
#  $acc
```

using cv.glmnet and use the "lambda.min" as final tuning parameter, which gives us 470 selected pixels. The prediction using those 470 pixels yielded a good start, a high prediction accuracy, for our following analysis.

Logistic Regression. Before applying logistic regression, we applied tow variable selection methods to reduce dimension. 400 pixels were selected by using Marginal Screening and 470 pixels were selected by using above Lasso Regression. Then, we applied logistic regression on those 400 and 470 pixels repectively. We then recorded their F1 score and AUC.

Neural Network. Neural Network was used in the project for seeking better classfication results. The raw data *train_images* and *test_images* was converted into numeric variables and reshaped into 3D array with dimension 7000 × 110 × 110 and 2734 × 110 × 110 respectively. The pixels were shrinked from 0-255 to 0-1 by dividing each pixel by 255. The target label (Volcano?) was converted into categorical variable for classification. The neural network used several convolutional dense layers for classification. The neural network model yielded a satisfying classification result.

Two neural network models were run in this project. Figure 10 shows the result of the first neural network model which is designed to classify whether an image contains a volcano. The model used 20% of the training data as validation data. Figure 11 shows the result of the second neural network model which is designed to classify whether an image contains at least one volcano.
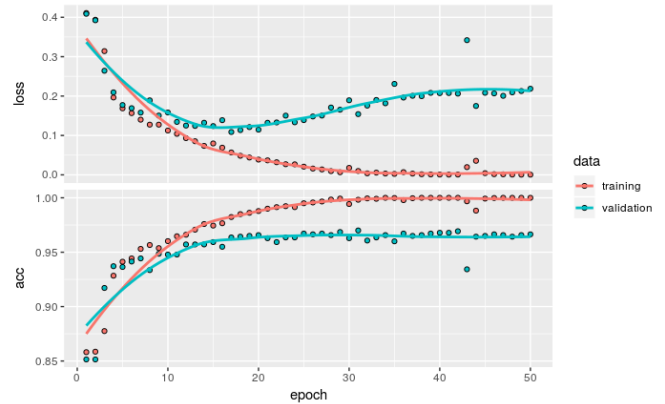
```
#  [1] 0.9297732
```

Variable selection methods.

Shrinking method (Lasso) By applying Lasso Regression, we were able to shrink highly correlated pixels to 0 and only had important one left. With the tuning parameter λ's interval we selected by examing the cv plot, we selected 470 important pixels and apply those to lasso and logistic regression respectively.

Marginal Screening To apply Marginal Screening, we use for loop to model each pixels with outcome by using glm function, logistic regression. After sorting the p values from smallest to largest, we selected the top 400 pixels. The reason why we select top 400 pixels is that, affter trying different number of top pixels, we found 400 pixels gave us highest prediction error.

Variable selection methods comparison We originally only applied Marginal Screening to logistic regression. But after realizing Lasso helped us selected variables as well, we tried both methods on logistic regression.

By looking at the pixels' index, we found that the pixels selected by Marginal Screening focused between 4000-7000, which corresponds to the center of pictures; while pixels selected by lasso regression were more spread.

By applying those two different selection methods, we found the accuracy of Lasso selection is higher. Therefore, we may conclude that lasso' more spread pixels made more sense because Marginal Screening's centered selection might ignore some important pixels at the edges.

IV. RESULT & CONCLUSION.

Table : Test dataset prediction result

Model	AUC	F1 Score
Neural Network	0.9629	0.9747
xgboost	0.9680	0.9628
Lasso	0.9347	0.9574
Logistic (Marginal Screening)	0.8429	0.9386
Logistic (Lasso Selection)	0.8672	0.9527

V. APPENDIX.