



# זיהוי נחשים ארסיים באמצעות למידת מכונה

הצגת פרויקט גמר בקורס למידת מכונה 55807

## מגישים:

- עידו כהן
- רון אפרים
- זוהר מור

# Outline

1. Motivation
2. EDA
3. Challenges and solutions
4. Pre-processing & training model
5. Finding the best model
6. Prediction and results
7. Conclusions & Answering the research question
8. Future developments



# Motivation

- קטלוג נחשים לנחשים ארסיים / לא ארסיים על סמך תמונה:

- למטייל הסקרן

- הצלת חיים - לעיכוב בטיפול עקב שגיאה בסיווג סוג הנחש יכולות להיות השלכות של חיים או מוות.

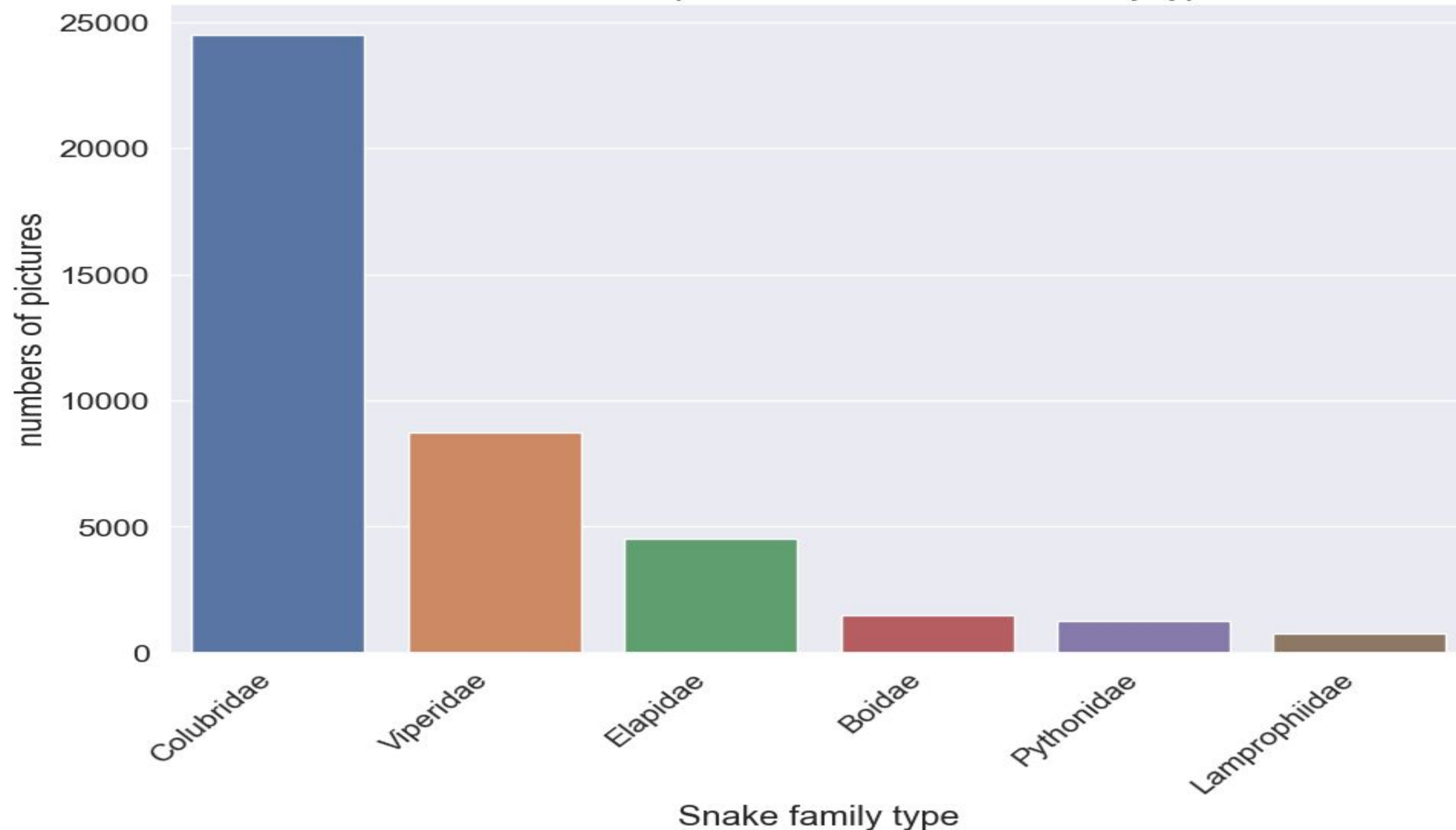
- בני אדם שנתקלים בנחש לעיתים קרובות מנסים להרגו מתוך הנחה מחמירה שמדובר בנחש ארסי.

- נטרול הצורך במומחה לסיווג נחשים ברמת סמך גבוהה שלא תמיד נמצא בפועל בשטח.



# EDA

The numbers of pictures for each snake family type



**קטלוג ע"פ משפחה:**

רוב התמונות של זעמניים  
(colubridae): כ-25000

לאחר מכן צפעוניים (viperidea):  
כ-10000

יתר התמונות – פתנים, חנקיים,  
פיתוניים וכו': פחות מ-5000  
תמונות לכל משפחת נחשים

# EDA

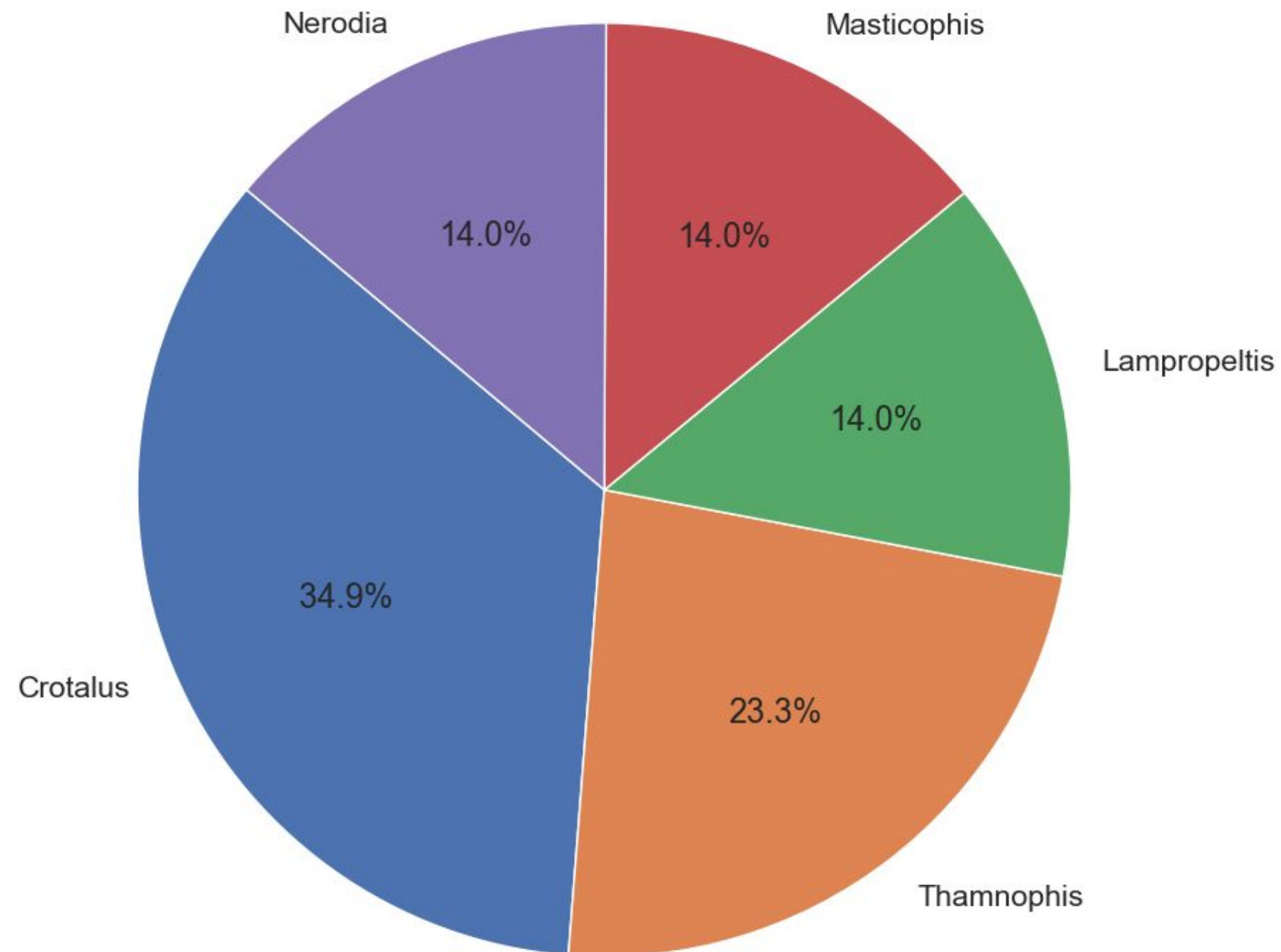
קטלוג על פי סוגי נחשים:

רוב התמונות של עכסניים  
(Crotalus) כ-35%

לאחר מכן נחשי מים  
(Thamnophis): כ-10000

סוגי הנחשים האחרים מתחלקים  
nerodia masticophis  
lamropeltis

Top 5 Most Common Values in the "genus" Column

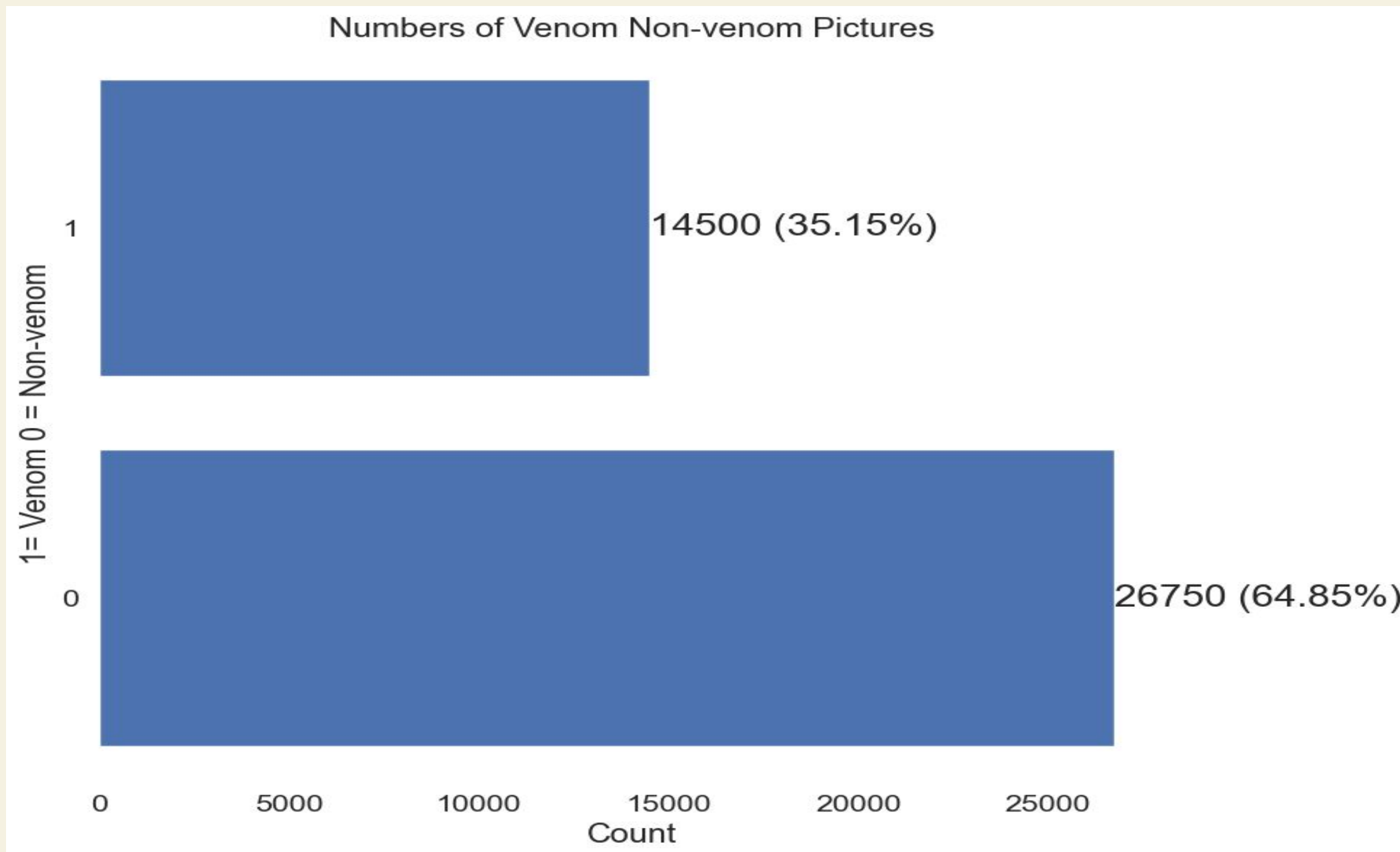


# EDA

קטלוג על פי ארסיות:

14500 – מתוייגות "ארסי"

26750 – מתוייגות "לא ארסי"



# EDA

Word Cloud of Continent Counts



ענן מילים לפי יבשות:  
רוב הנחשים  
מצפון אמריקה



# EDA

## ירידה לפרטים:

ניתן לראות כי רוב הנחשים  
במאגר מסווגים מבחינת יבשת  
כצפון אמריקה.

לאחר מכן - סיווג היבשת  
הוא לא ידוע.

Summary of Snakes by Continent

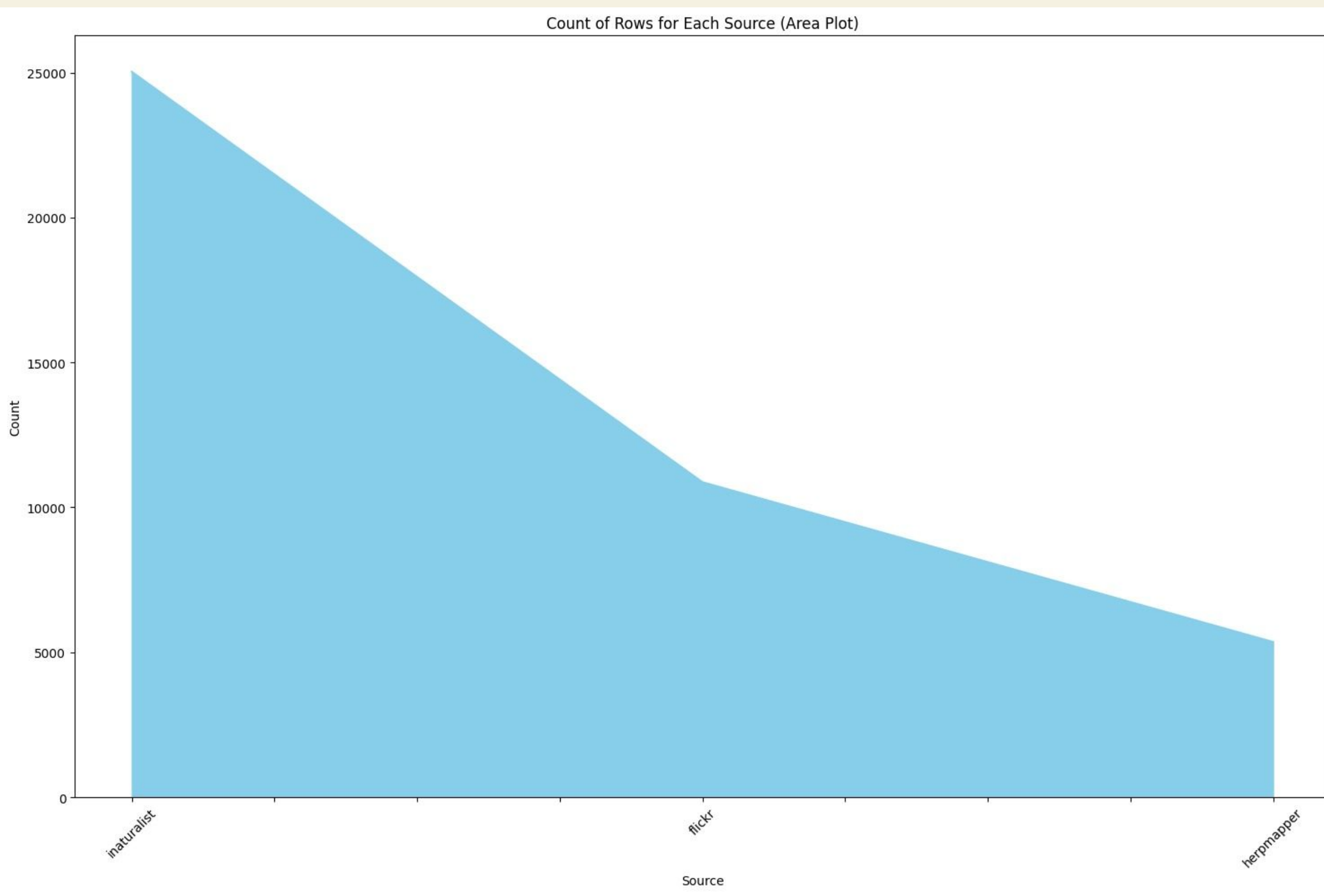
	total_snakes	unique_snakes	unique_venomous_snakes
Africa	1455	19	10
Asia	3835	38	18
Australia	1217	19	11
Europe	2519	50	22
North America	19455	113	29
Oceania	127	12	9
South America	2318	56	13
unknown	10324	165	58
Total	41250	472	170



# EDA

## מקור התמונות

רוב התמונות הן מ- Naturalist - 25000



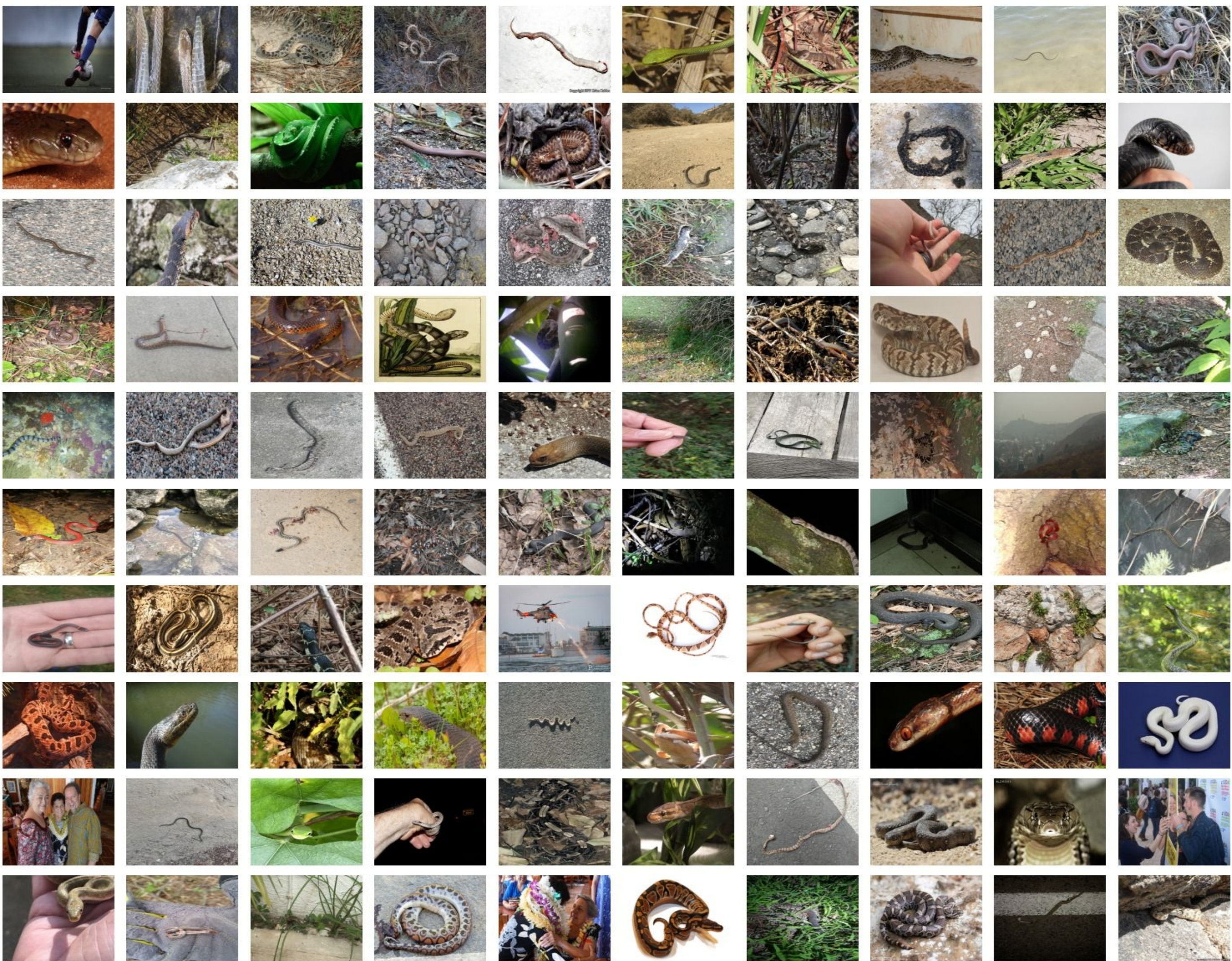
# Challenges and solutions

**תחילה ניסינו לאמן את המודל במחשבים האישיים שלנו, אך מכיוון שכמות התמונות גדולה (כ-42,000 תמונות) והמודל מקבל מערכים מפוקסלים הגענו למערכים בגודל  $240 \times 240 \times 3 \times 42000$  וה-RAM במחשבים האישיים שלנו לא הספיק, וההרצה נכשלה.**

**פתרון:** השתמשנו ב Google Colab וראינו שאכן כמות התמונות שניתן באמצעותן לאמן את המודל גדלה משמעותית אך עדיין זמן הריצה עבור אימון המודל היה גדול מאוד ולכן נקטנו ב 2 טכניקות נוספות:

1. שמרנו את המודל ואובייקטים נוספים בהם המודל משתמש לחיסכון במשאבים בהרצות הבאות.
2. רכשנו כוח חישוב נוסף ע"י רכישת אפשרות שימוש במעבדים גרפיים נוספים - GPU, TPU.







# Challenges and solutions

**תמונות זבל רבות בדאטא. כשהסתכלנו על הגריד ראינו שתמונות רבות אינן של נחשים והן משפיעות על יכולות המודל ללמוד וראינו את זה לאורך אימוני המודל.**

**פתרון:** מבחינה מדוקדקת של הדאטא שמנו לב שהרבה מהתמונות המזבלות הן ממקור Flickr והחלטנו שעל אף אבדן הדאטא לסנן אותה. עם זאת, מדובר בפתרון חלקי ולא הצלחנו מבחינת מגבלת הזמנים להיפטר בצורה חכמה מהתמונות שהן אינן של נחשים או למצוא מכנה משותף לתמונות אלה שיאפשר לפלטר אותן בקלות. מעריכים שביצועי המודל מושפעים מהותית מעובדה זו.

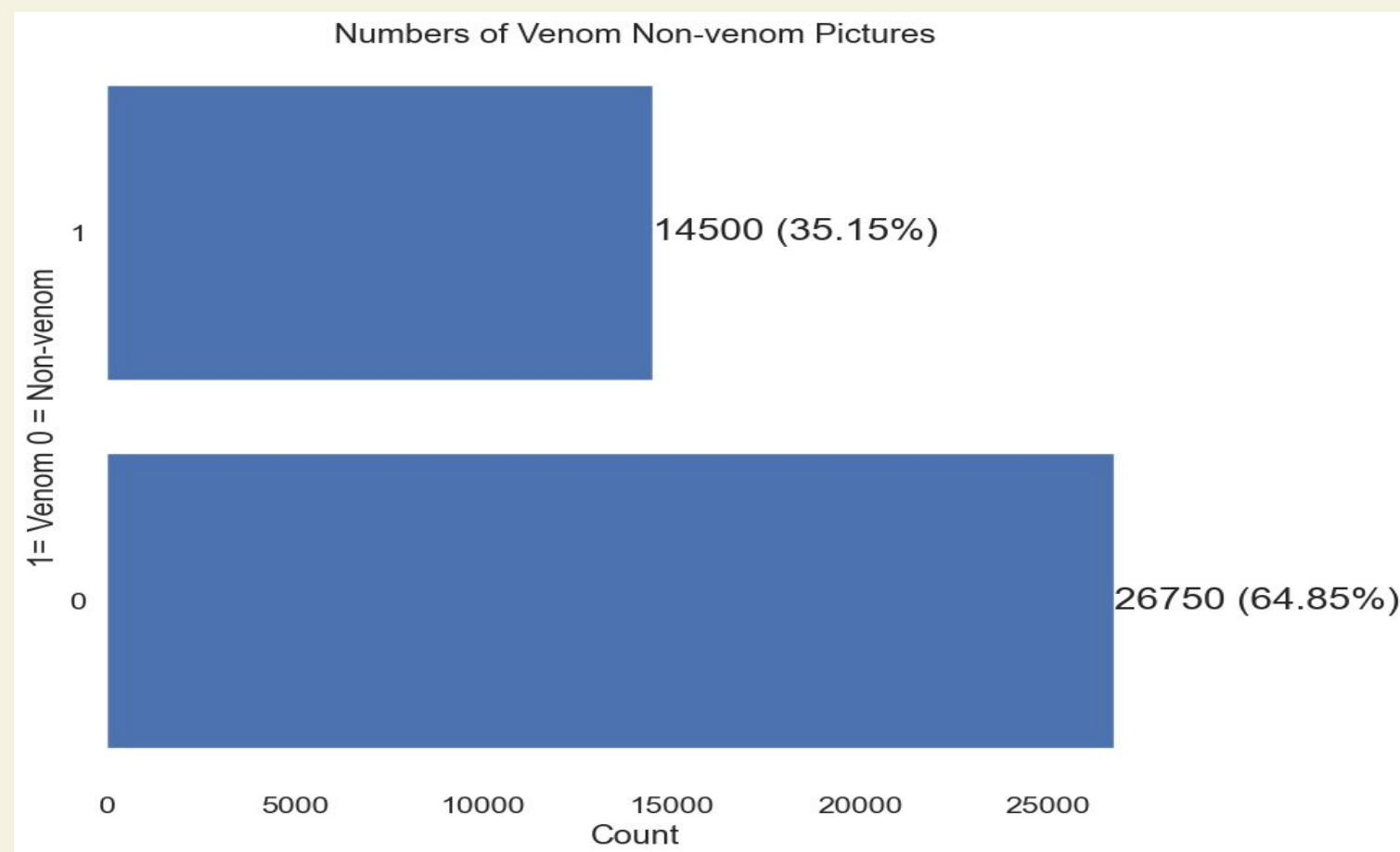




# Challenges and solutions

**חוסר איזון בין תמונות של נחשים ארסיים ולא ארסיים.**

**פתרון:** השמטה רנדומלית של תמונות נחשים שקוטלגו "לא ארסיים", כך שבעת האימון היה איזון. בתום שני השלבים האלה הגענו ל-18000 תמונות בלבד - גורם שהגביל את מידת יכולתנו לבנות מודל מורכב. בחרנו להשתמש ב-10K לאימון.



```
poisonous
0      9609
1      9609
Name: count, dtype: int64
```

# Pre-processing & training model

- Pre-processing - ייצוג כל תמונה באמצעות וקטור - כל תמונה הפכנו לווקטור של פיקסלים המייצג את התמונה.
- בחרנו לאמן את המודל באמצעות 10,000 תמונות כאשר התמונות נבחרו באופן רנדומלי ולא לפי סדר מסוים וזאת כדי לבחור תמונות של נחשים מכל הסוגים.
- השארנו 8000 תמונות ל-TEST.
- הכנת הדאטה למודל - חלוקת 10000 התמונות ל-80:20.

```
Train images shape: (7976, 240, 240, 3)
Val images shape: (1994, 240, 240, 3)
Train labels shape: (7976,)
Val labels shape: (1994,)
```

# Find the best model

**בחינת מספר ארכיטקטורות של נוירונים:**

בשלב זה רצינו לשפר את המודל ולכן בדקנו מה המספר והקומבינציות השונות של השכבות והנוירונים שיתנו לנו את אחוז הניבוי הגבוהה ביותר. להלן התוצאות:

# Reasons for choosing this model

הבחירה ברשתות קונבולוציה

- יכולת לתפוס מאפיינים מורכבים יותר של תמונות (spatial patterns) כגון קצוות, טקסטורות וצורות מורכבות, יכולת שנחוצה עבור מטלות זיהוי תמונה, וכן מאפייניהם העדינים של הנחשים.
- יכולת הרשת לזהות את אותם פיצ'רים ללא קשר למיקומם בתמונה או לאוריינטציה שלהם.
- יעילות וקלות אימון עבור דאטה מוגבל (כמו שיש בידינו) על ידי שיתוף פרמטרים לעומת רשת שהיא fully-connected.



model.summary()		
Model: "sequential_12"		
Layer (type)	Output Shape	Param #
rescaling_6 (Rescaling)	(None, 240, 240, 3)	0
random_flip_6 (RandomFlip)	(None, 240, 240, 3)	0
random_rotation_6 (RandomRotation)	(None, 240, 240, 3)	0
random_contrast_6 (RandomContrast)	(None, 240, 240, 3)	0
random_zoom_6 (RandomZoom)	(None, 240, 240, 3)	0
conv2d_30 (Conv2D)	(None, 238, 238, 32)	896
batch_normalization_48 (BatchNormalization)	(None, 238, 238, 32)	128
re_lu_48 (ReLU)	(None, 238, 238, 32)	0
max_pooling2d_30 (MaxPooling2D)	(None, 119, 119, 32)	0
conv2d_31 (Conv2D)	(None, 117, 117, 64)	18496
batch_normalization_49 (BatchNormalization)	(None, 117, 117, 64)	256
re_lu_49 (ReLU)	(None, 117, 117, 64)	0
max_pooling2d_31 (MaxPooling2D)	(None, 58, 58, 64)	0
conv2d_32 (Conv2D)	(None, 56, 56, 128)	73856
batch_normalization_50 (BatchNormalization)	(None, 56, 56, 128)	512
re_lu_50 (ReLU)	(None, 56, 56, 128)	0
max_pooling2d_32 (MaxPooling2D)	(None, 28, 28, 128)	0
conv2d_33 (Conv2D)	(None, 26, 26, 256)	295168
batch_normalization_51 (BatchNormalization)	(None, 26, 26, 256)	1024
re_lu_51 (ReLU)	(None, 26, 26, 256)	0
max_pooling2d_33 (MaxPooling2D)	(None, 13, 13, 256)	0
conv2d_34 (Conv2D)	(None, 11, 11, 512)	1180160
batch_normalization_52 (BatchNormalization)	(None, 11, 11, 512)	2048
re_lu_52 (ReLU)	(None, 11, 11, 512)	0
max_pooling2d_34 (MaxPooling2D)	(None, 5, 5, 512)	0
flatten_6 (Flatten)	(None, 12800)	0
dense_24 (Dense)	(None, 1024)	13108224

# Pre-processing & training model

## Model Summary

- 5 שכבות קונבולוציה - שבכל אחת סט קרנלים (חלונות שרצים על התמונה).
- אופטימיזציה - Batch Normalization
- פונקצית האקטיבציה לשכבות אלה - ReLU
- שכבה נוספת של Max Pooling 2D - מייצרת החלונות (Strides)
- שכבת אגרגציה / השטחה שעושה את ההמרה משכבת קונבולוציה לשכבות "fully connected"
- 3 שכבות "fully connected" מנורמלות ומאופטמות.
- שכבת יציאה - בינארית (נוירון יציאה אחד)
- פונקציית אקטיביציה SIGMOID ממפה ערכים לערכים בין 0 - 1. ומתאימה מאד לבעיות רגרסיה לוגיסטית ולקלאסיקפציה בינארית.

model.summary()

Model: "sequential\_12"

Layer (type)	Output Shape	Param #
rescaling_6 (Rescaling)	(None, 240, 240, 3)	0
random_flip_6 (RandomFlip)	(None, 240, 240, 3)	0
random_rotation_6 (RandomRotation)	(None, 240, 240, 3)	0
random_contrast_6 (RandomContrast)	(None, 240, 240, 3)	0
random_zoom_6 (RandomZoom)	(None, 240, 240, 3)	0
conv2d_30 (Conv2D)	(None, 238, 238, 32)	896
batch_normalization_48 (BatchNormalization)	(None, 238, 238, 32)	128
re_lu_48 (ReLU)	(None, 238, 238, 32)	0
max_pooling2d_30 (MaxPooling2D)	(None, 119, 119, 32)	0
conv2d_31 (Conv2D)	(None, 117, 117, 64)	18496
batch_normalization_49 (BatchNormalization)	(None, 117, 117, 64)	256
re_lu_49 (ReLU)	(None, 117, 117, 64)	0
max_pooling2d_31 (MaxPooling2D)	(None, 58, 58, 64)	0
conv2d_32 (Conv2D)	(None, 56, 56, 128)	73856
batch_normalization_50 (BatchNormalization)	(None, 56, 56, 128)	512
re_lu_50 (ReLU)	(None, 56, 56, 128)	0
max_pooling2d_32 (MaxPooling2D)	(None, 28, 28, 128)	0
conv2d_33 (Conv2D)	(None, 26, 26, 256)	295168
batch_normalization_51 (BatchNormalization)	(None, 26, 26, 256)	1024
re_lu_51 (ReLU)	(None, 26, 26, 256)	0
max_pooling2d_33 (MaxPooling2D)	(None, 13, 13, 256)	0
conv2d_34 (Conv2D)	(None, 11, 11, 512)	1180160
batch_normalization_52 (BatchNormalization)	(None, 11, 11, 512)	2048
re_lu_52 (ReLU)	(None, 11, 11, 512)	0
max_pooling2d_34 (MaxPooling2D)	(None, 5, 5, 512)	0
flatten_6 (Flatten)	(None, 12800)	0
dense_24 (Dense)	(None, 1024)	13108224

# Pre-processing & training model

## Model Summary

- אוגמנטציות - בגלל כמות הדאטא הקטנה שהייתה לנו ואנדרפיטינג שראינו במודלים הראשוניים שלנו השתמשנו באוגמנטציות שמנפחות את גיוון הדאטא ומשפרות ACCURACY בקונטקסטים שונים.

```
model = Sequential([Rescaling(1./255),
                    RandomFlip("horizontal_and_vertical"),
                    RandomRotation(0.2),
                    RandomContrast(0.2),
                    RandomZoom(height_factor=(0.2, 0.3))])
```

- פונקציית האובדן - binary cross entropy

```
history = model.fit(train_images, train_labels, validation_data=(val_images, val_labels), epochs=10,
                    batch_size=32, verbose=2)
```



# Prediction and Results

בשלב הזה הרצנו 20 תמונות רנדומליות (לאחר וידוא שאינן חלק מהתמונות ששימשו אותנו לאימון המודל) וזאת כדי לחזות את התוצאות שלהן ולראות האם התוצאות שהמודל חוזה אכן זהות לתיוג שלהן ( $y^{\wedge}-y$ ).

התמונות שעברו INFERENCE עברו PREPROCESSING

Filename: 10d059e4b14f42faa230a1666a004d14.jpg  
Image at index 1649 is not listed in the UUIDs text file. Inferencing...  
1/1 [=====] - 0s 20ms/step  
Predicted probability of the snake in picture being venomous: 0.7123411893844604  
Predicted label: Venom  
Actual label: Venom  
True Prediction

Chosen Image



Filename: 1e70c969722746c58810f54bcd10ffbf.jpg  
Image at index 12956 is not listed in the UUIDs text file. Inferencing...  
1/1 [=====] - 0s 19ms/step  
Predicted probability of the snake in picture being venomous: 0.5667087435722351  
Predicted label: Venom  
Actual label: Non-Venom  
False Prediction

Chosen Image



Filename: c6b2c4b801c04bb8ba430dd30b671308.jpg  
Image at index 1993 is not listed in the UUIDs text file. Inferencing...  
1/1 [=====] - 0s 20ms/step  
Predicted probability of the snake in picture being venomous: 0.49716538190841675  
Predicted label: Non-Venom  
Actual label: Non-Venom  
True Prediction

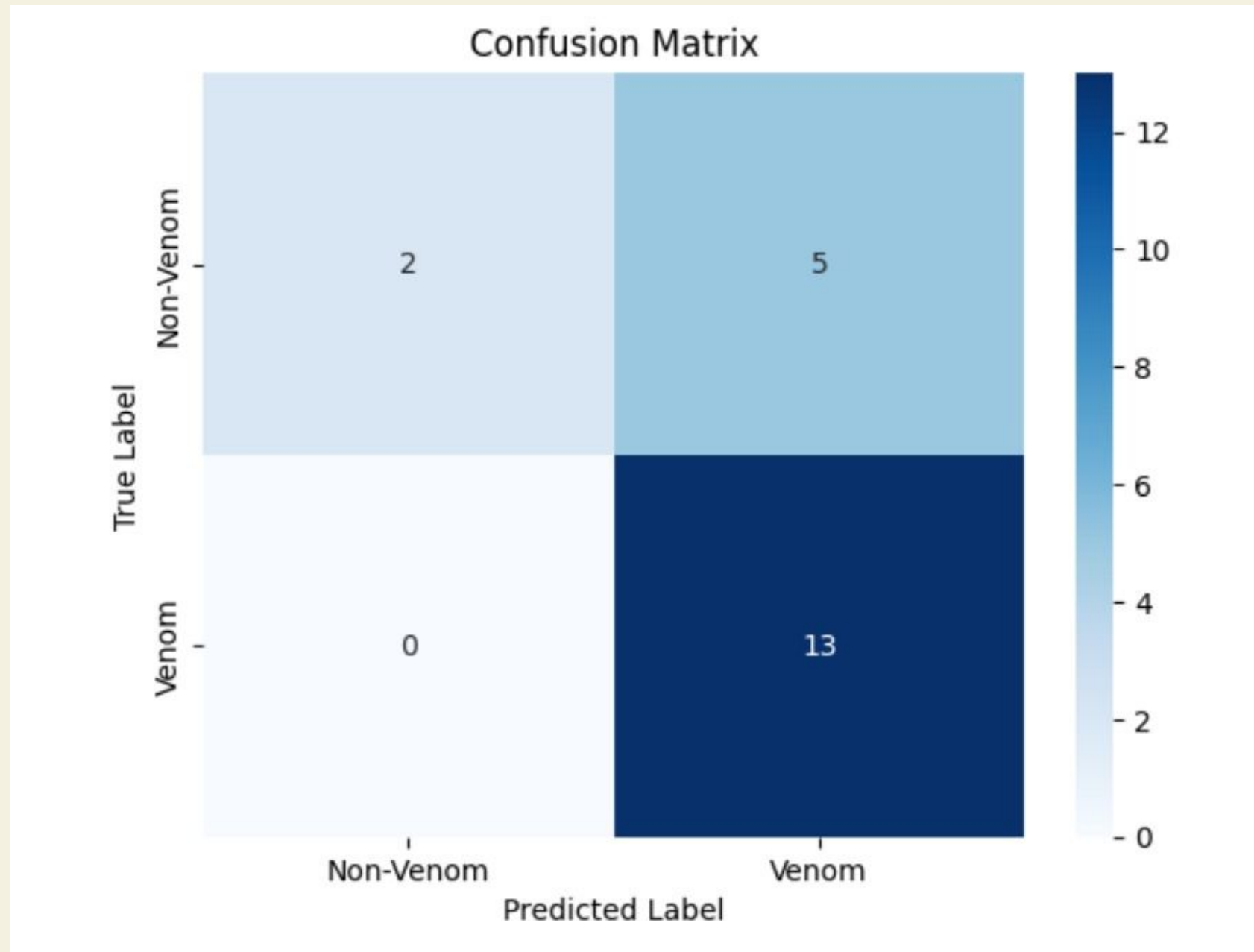
Chosen Image





# Confusion Matrix

63/63 [=====] - 2s 27ms/step - loss: 0.6771 - accuracy: 0.5597  
Test Accuracy: 0.5596790313720703  
Training Time: 261.76941871643066



ההכרעה על פרדיקציה של תמונה התקבלה באמצעות  
השוואה ל- Threshold של 50%.

ניתן לראות מהקונפיושיין מטריקס כי 15 תמונות  
רנדומליות נחזו באופן נכון.



# Model weaknesses

ישנן מספר סיבות שאנחנו חושדים בהן כסיבות שהמודל לא מושלם:

1. **"רעש בתמונות"**: למרות מספר התמונות הרב שהיה ברשותנו, תמונות רבות היו מזובלות והדאטה שנשארנו איתו לא רב. בנוסף, תמונות רבות הכילו אלמנטים רבים מלבד הנחש עצמו, ולא צולמו בצורה מקצועית. מוערך כי המודל מתקשה להבחין בדיוק בנחש ולהפרידו מיתר האלמנטים בתמונה.

באמור, אחת הטכניקות שניסינו ליישם כדי לשפר את הניבוי הוא על ידי Augmentation, אכן טכניקה זו עזרה לשיפור הניבוי אך עדיין לא בצורה מספקת.

2. **מורכבות התמונות**: תמונות הינן צבעוניות ולרוב באיכות טובה ולכן היכולת לאמן מודל עם תמונות מורכבות דורשת כוח עיבוד רב (אותו פתרנו על ידי Google Colab) טכניקות עיבוד תמונה שאולי אותן אנחנו לא מכירים.

# Model weaknesses

ישנן מספר סיבות שאנחנו חושדים בהן כסיבות שהמודל לא מושלם:

3. **מורכבות הדרישה:** מלבד היותן תמונות מורכבות, אנו נדרשים ללמד מודל להבחין בין נחש ארסי ולא ארסי. ההבדלים בין נחשים ארסיים ולא ארסיים פעמים רבות הם מינורים כמו מבנה ראש קצת שונה, צורת עור נחש מעט שונה. ישנן הרבה נחשים דומים מאוד שהם קרובי משפחה וחלקן ארסיים וחלקן לא. אנחנו מאמינים שדרישה מורכבת זו הקשתה על יכולת חיזוי טובה של המודל. לסיכום, אף התייעצנו עם זאולוג שהכריע שלא בהכרח קיים הבדל ויזואלי בין ארסי ללא ארסי.

4. **מגבלת זמן וכוח חישוב לא אפשרה לנו לעשות fine-tuning לפרמטרים ולשחק עם ארכיטקטורות שונות במידה מספקת.** Accuracy - 60% Loss 60%.

Epoch 10/10

250/250 - 25s - loss: 0.6603 - accuracy: 0.5821 - val\_loss: 0.8554 - val\_accuracy: 0.5476 - 25s/epoch - 100ms/step

# Future developments

סינון התמונות המזבלות בצורה אופטימלית.



יצירת אפליקציה למובייל. אפיקי שיתוף פעולה עם גוגל / אפל - התחברות לממשקים שלהם לזיהוי תמונה בצורה נגישה ומהירה יותר למשתמש.



איסוף דאטה ושיפור ריקול על פי מיקום גאוגרפי (לדוגמא תמונה שצולמה בישראל ומזוהה כנחש מברזיל)



הרחבת האפליקציה לזיהוי חיות נוספות אשר מסוכנות בפוטנציה כמו לטאות חרקים או צמחים.



המרת המודל ממודל בינארי (ארסי / לא ארסי) למודל מקטלג שיכול לתת אינפורמציה נוספת על אותו הנחש.



# Conclusions & Answer the research question



**לסיכום: ניתן להגיד שהתקשנו לענות בצורה חד משמעית על שאלת המחקר עבור ניבוי האם נחש הוא ארסי או לא. יחד עם זאת, במהלך העבודה המשותפת העמקנו את הבנתנו ברשתות קונבולוציה ומשימות קלסיפיקציה של עיבוד תמונה**

**במסגרת הפרוייקט התמודדנו עם אתגרים וחווינו לא מעט אכזבות עם זאת, רכשנו כלים שימושיים וסיגלנו שיטות עבודה הקשורות בלמידת מכונה ועל כך אנחנו מודים.**