

Reto sobre aplicación de Python para respuesta a preguntas de analítica

Profesores:

Christian Urcuqui (ulcamilo@gmail.com)

Universidad Icesi

Objetivos de esta sesión

1. Usar Python para la exploración de un conjunto de datos.
2. Responder a un conjunto de preguntas a partir de los datos
3. Aplicar Python para el desarrollo de funciones, ciclos, asignación de variables y explorar la información de un dataframe.

Taller práctico

Contexto

Análisis de datos

PetFinder.my Adoption Prediction **How cute is that doggy in the shelter?**

Millones de animales sufren en la calle o son sacrificados en refugios todos los días en todo el mundo. Si se pueden encontrar hogares para ellos, se pueden salvar muchas vidas preciosas —y se pueden crear más familias felices.

Esta competencia busca que se los participantes puedan proponer algoritmos que permitan predecir la capacidad de adopción de las mascotas, ¿con qué rapidez se adopta una mascota?

<https://www.kaggle.com/c/petfinder-adoption-prediction>

 Featured Code Competition

PetFinder.my Adoption Prediction

How cute is that doggy in the shelter?

\$25,000

Prize Money



PetFinder.my · 2,023 teams · 2 years ago

[Overview](#)

[Data](#)

[Code](#)

[Discussion](#)

[Leaderboard](#)

[Rules](#)

[Team](#)

[My Submissions](#)

[Late Submission](#)

Overview

Description

Evaluation

Timeline

Prizes

Kernels-FAQ

Millions of stray animals suffer on the streets or are euthanized in shelters every day around the world. If homes can be found for them, many precious lives can be saved — and more happy families created.

[PetFinder.my](#) has been Malaysia's leading animal welfare platform since 2008, with a database of more than 150,000 animals. PetFinder collaborates closely with animal lovers, media, corporations, and global organizations to improve animal welfare.

Animal adoption rates are strongly correlated to the metadata associated with their online profiles, such as descriptive text and photo characteristics. As one example, PetFinder is currently experimenting with a simple AI tool called the Cuteness Meter. which ranks how cute a pet is based on qualities present in their photos.



PREGUNTAS DE ANÁLISIS

Vamos a utilizar solo el archivo train.csv

Complejidad 1

1. ¿Cuántos registros y variables cuenta el dataset?
2. ¿Cómo están los valores de la variable Age?
 1. ¿Cuál es la distribución por tipo de animal
 2. Cuales son los estadísticos descriptivos para esta variable.
3. ¿Hay más datos de perros o de gatos?
4. ¿Cómo es la velocidad de adopción para perros y gatos?
5. Encuentre los nombres sin valores.
 1. ¿Cuántos para gatos y perros?
 2. ¿Eliminar o reemplazar los registros?, justifique
6. ¿la salud podría significar algo en la velocidad de adopción?

PREGUNTAS DE ANÁLISIS

Vamos a utilizar solo el archivo train.csv

Complejidad 2

7. ¿Cuales son los cinco nombres más populares para los perros?
8. Revise la relación entre la variable de edad y la velocidad de adopción.
9. ¿Existe alguna relación entre la longitud del nombre y la velocidad de adopción?

Complejidad 3

10. Ejecute las siguientes líneas de código e interprete

```
1 df_2 = df.dropna().copy()
2 df_2 = df_2.loc[df_2.Name.isna() != True].copy()
3 df_2["len_name"] = df_2.loc[:, "Name"].apply(len)
4 df_2["new_age"] = pd.cut(df_2.Age, bins=np.arange(0,300,12, dtype=int),
5                          labels=np.arange(0,len(np.arange(0,300,12, dtype=int)))[:-1], include_lowest=True).astype(int)
```

11. Ejecute las siguientes líneas e interprete

```
1 from sklearn import tree
2 from sklearn.model_selection import train_test_split
3
4 features = list(set(df_2.columns) - set(["AdoptionSpeed", "PetID", "RescuerID",
5     "Description", "Name", "Age", 'PhotoAmt', 'VideoAmt']))
6
7 X_train, X_test, y_train, y_test = train_test_split(df_2[features], df_2.AdoptionSpeed, test_size=0.33, random_state=42)
8
```

12. Ejecute e interprete los resultados de clasificación (precision and recall)

```
clf = tree.DecisionTreeClassifier()
clf = clf.fit(X_train, y_train)

from sklearn.metrics import classification_report

y_pred = clf.predict(X_test)

print(classification_report(y_test, y_pred))
```


Complejidad 3

13. Ejecute las siguientes líneas de código ¿Cuál es el propósito de `mutual_info_classif`?

```
1
2 discrete_features = [True, False, False, False, False, False,
3                     True, False, False, False, False,
4                     True, True, False, False, False, False, False]
```

```
1 from sklearn.feature_selection import mutual_info_classif
2
3 def make_mi_scores(X, y, discrete_features):
4     mi_scores = mutual_info_classif(X, y, discrete_features= discrete_features)
5     mi_scores = pd.Series(mi_scores, name="MI Scores", index=X.columns)
6     mi_scores = mi_scores.sort_values(ascending=False)
7     return mi_scores
8
9 mi_scores = make_mi_scores(X_train, y_train, discrete_features)
10 mi_scores[::3]
```

```
1 def plot_mi_scores(scores):
2     scores = scores.sort_values(ascending=True)
3     width = np.arange(len(scores))
4     ticks = list(scores.index)
5     plt.barh(width, scores)
6     plt.yticks(width, ticks)
7     plt.title("Mutual Information Scores")
```

```
import matplotlib.pyplot as plt
import numpy as np
```

```
plt.figure(dpi=100, figsize=(8, 5))
plot_mi_scores(mi_scores)
```

Complejidad 3

14. Con base al anterior resultado seleccione cuatro variables, entrene y evalúe un nuevo modelo con las mismas configuraciones y datos. Interprete el proceso y los nuevos resultados