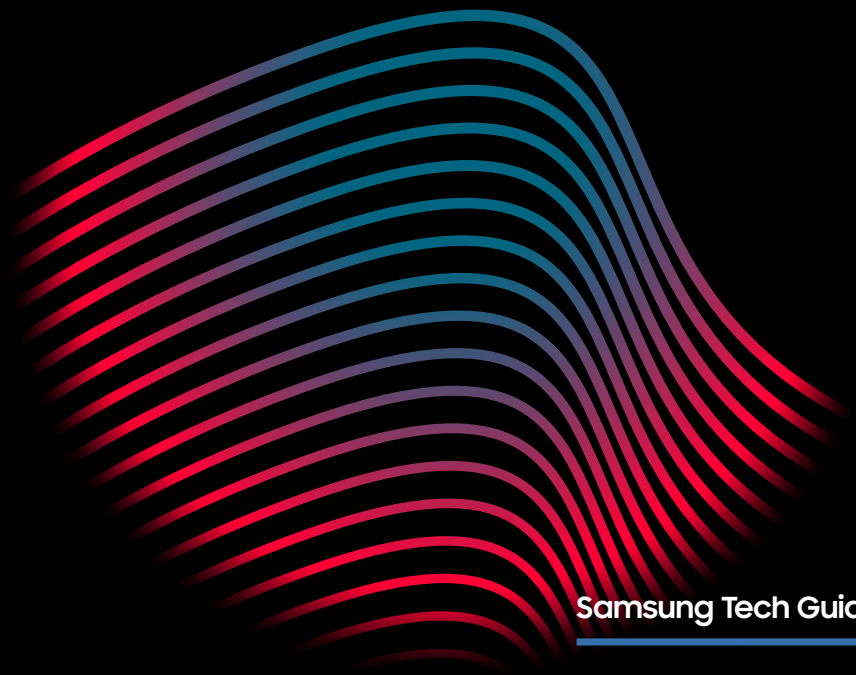
Samsung R&D Institute Poland

Python in Data Science Tech Guide

Poszerzaj i utrwalaj znajomość zagadnień technicznych, korzystając z praktycznych wskazówek naszych ekspertów.



Samsung Tech Guides

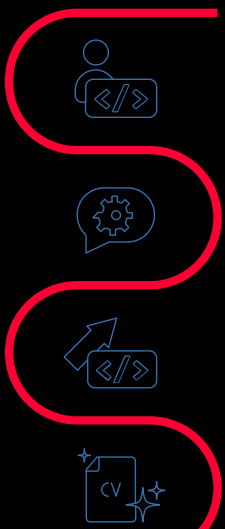
Drogi kandydacie!
Droga kandydatko!

Zapraszamy Cię do zapoznania się z Python in Data Science Technical Guide - przewodnikiem poświęconym jednej z najgłośniejszych dziedzin ostatnich lat!

Znajdziesz w nim tematy, które warto powtórzyć przed spotkaniem technicznym w rekrutacji do Samsung R&D!

Dodatkowo nasi Inżynierowie przygotowali przykładowe zadania testowe, które pomogą Ci sprawdzić swoją wiedzę.

Python in Data Science Technical Guide - Dla kogo?



dla kandydatów aplikujących na stanowiska, w których wymagana jest znajomość Pythona,

dla kandydatów zaproszonych już na spotkanie techniczne do Samsung R&D,

dla osób zaczynających przygodę z Data Science i chcących poszerzyć/utrwalić swoją wiedzę z Pythona,

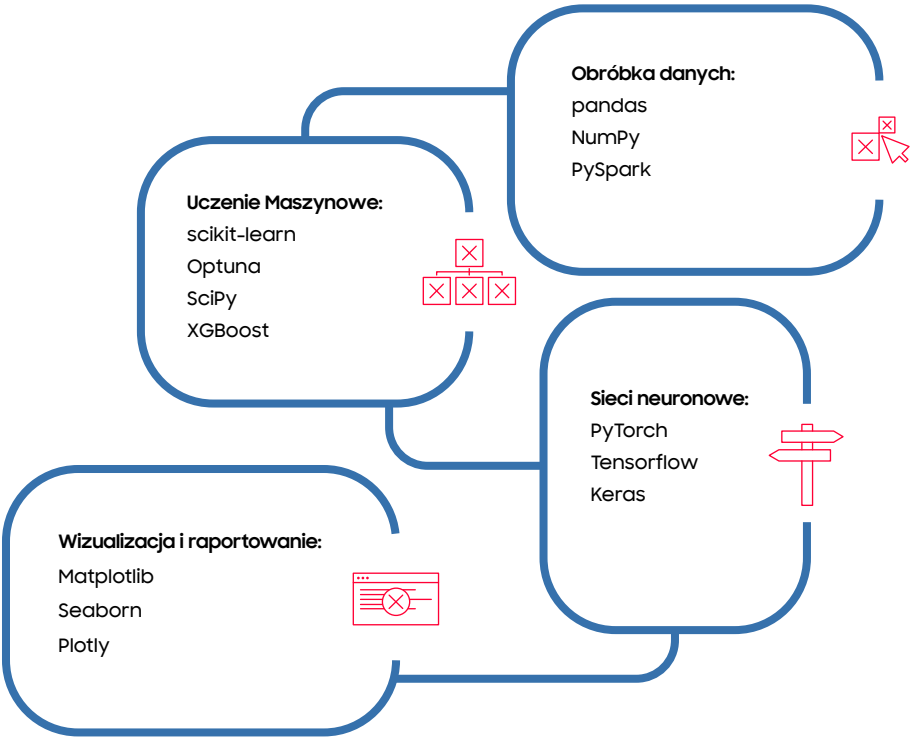
dla osób zainteresowanych w przyszłości rekrutacją do Samsung R&D.

Powodzenia!

Kluczowe pakiety

Codzienne obowiązki Data Scientista wymagają pracy z popularnymi bibliotekami języka Python.

W zależności od zagadnienia są to:



Python in Data Science

Data Science jest prężnie rozwijającą się dziedziną, która łączy umiejętność programowania z wiedzą statystyczną. W związku z tym od kandydatów oczekujemy znajomości następujących zagadnień:

- Czym jest pętla **for** i czym różni się od **while**?
- Instrukcja **if-elif-else** oraz operatory: **and**, **or**, **not**.
- Na czym opiera się logika boolowska?
- Do czego służą instrukcje try, except i finally?
- Jak sprawdzić typ danych?
- Jakie przykładowe konwersje typów można wykonać i w jaki sposób?
- Charakterystyka klas: **list**, **tuple**, **dict**, **set**.
- Definiowanie funkcji standardowych i anonimowych.
- Czym są *args i **kwargs?
- Czym jest polimorfizm i dziedziczenie?
- Czy zawsze należy zaimportować całą bibliotekę?
- Jak zastosować funkcję z innego skryptu? Jakie warunki muszą być spełnione?

Moduły w Pythonie

- Czym różni się numpy.array od listy?
- Jak działają funkcje: numpy.where(), numpy.argmax(),...?
- Do czego służy funkcja .reshape() i operacja .T na obiekcie typu numpy.array?
- Jak wczytać serię plików .csv do jednego pandas.DataFrame?
- Do czego służą funkcje .loc i .iloc? Co je różni?
- Jak zliczyć wystąpienia poszczególnych wartości w kolumnie?
- Jak dodać kolumnę będącą sumą dwóch inych kolumn? Co gdy operacja tworzenia jest bardziej złożona?
- Łączenie (.join(), .merge(), .concat()) i agregacja (.groupby(), .agg()) ramek danych.
- Jaka jest różnica między pandas.Series, a pandas.DataFrame?
- Jaka jest różnica między transformacjami .map() i .filter() w PySpark? Kiedy powinny być stosowane?
- Znajomość API(.fit(), .predict(), .transform(), .fit_transform(), .predict_proba()).
- Znajomość modułu preprocessing (OneHotEncoder, MinMaxScaler, StandardScaler, ColumnTransformer).
- Znajomość modułu model_selection.
- Pakiety wizualizacji danych: matplotlib.pyplot/seaborn/plotly.

Podstawy deweloperskie

- Czym jest mikroserwis i po co się go tworzy? (Dash/Streamlit/Flask).
- Różnica między tworzeniem kodu w skryptach i w notebookach.
- Instalacja pakietów w Pythonie.
- Nawigacja z poziomu terminala i edycja plików.
- Znajomość systemu kontroli wersji Git.

Prawdopodobieństwo i statystyka

- Podstawowe rozkłady ciągłe i dyskretne (normalny, wykładniczy, Bernoulliego, beta, gamma).
- Statystyki opisowe (kwantyle, średnia, odchylenie standardowe, wariancja, entropia).
- Centralne Twierdzenie Graniczne (Z-score).
- Testy statystyczne (równość średnich i median, rozproszenia, normalności, identyczności rozkładu, p-value).

Podstawy Uczenia Maszynowego

- Obserwacje odstające (sposoby detekcji, znaczenie, potencjalne rozwiązania).
- Braki danych (losowe i nielosowe, imputacja).
- Metody podziału zbiorów.
- Bias-variance trade-off, overfitting, underfitting, regularyzacja.
- Próbkowanie, zbiory niezbalansowane.
- Feature selection.
- Feature engineering.
- Uczenie nadzorowane/nienadzorowane.
- Wyjaśnialność modeli.

Uczenie nadzorowane

- Czy współliniowość jest istotna w kontekście zmiennych objaśniających? Jeśli tak, to dlaczego i jak można to zbadać?
- Czy można użyć regresji liniowej do modelowania probabilistycznej zmiennej zależnej?
- Jak wygląda proces tworzenia i predykcji modelu drzewa regresyjnego?
- Jaka jest różnica między regularyzacją L1 i L2?
- Jakie znasz funkcje straty dla zagadnień regresji? Podaj przykłady zastosowania.
- Jak wygląda estymacja regresji liniowej?
- Czy R^2 jest dobrą miarą ewaluacji modelu?
- Co zapewni ADJ R^2?
- Jakie są podstawowe modele klasyfikacji?
- Czym jest boosting na przykładzie XGboost?
- Jakie znasz funkcje straty dla zagadnień klasyfikacji? Podaj przykłady zastosowania.
- Jaka jest różnica między klasyfikacją dwuklasową i wieloklasową?
- Jakie są podstawowe metody ewaluacji modelu klasyfikacji binarnej?
- Jak wygląda to w przypadku wieloklasowym?

Uczenie nienadzoro-wane

- Jakie są rodziny modeli uczenia nienadzorowanego?
- Po co redukuje się wymiarowość?
- Czy PCA ma inne zastosowania niż redukcja wymiarowości?
- Jaka jest różnica między PCA, a t-SNE?
- Jak optymalnie dobrać liczbę skupisk?

Sieci neuronowe

- Implementacja modeli w TensorFlow lub PyTorch.
- Jakie znasz typy architektur sieci neuronowych? Gdzie się je stosuje?
- Jakie znasz architektury sieci neuronowych w zagadnieniach uczenia nadzorowanego i nienadzorowanego?
- Opisz mechanizm wstecznej propagacji błędu.
- Do czego służy optimizer?

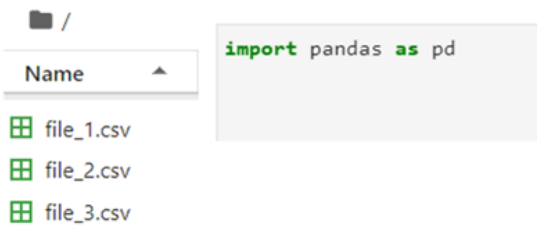
1.
Mamy dany obiekt:

```
import numpy as np
A=np.array([[2,2,1,1],[2,0,1,1],[0,7,5,1],[4,6,1,2]])
```

Wykorzystując własności obiektu numpy.array:

- Znajdź rozmiar i wymiary obiektu A.
- Utwórz macierz B składającą się z elementów wchodzących w skład dwóch ostatnich rzędów i dwóch pierwszych kolumn macierzy A.
- Transponuj macierz B, a następnie przedstaw wynik w postaci listy.

2.
Dana jest seria plików .csv:

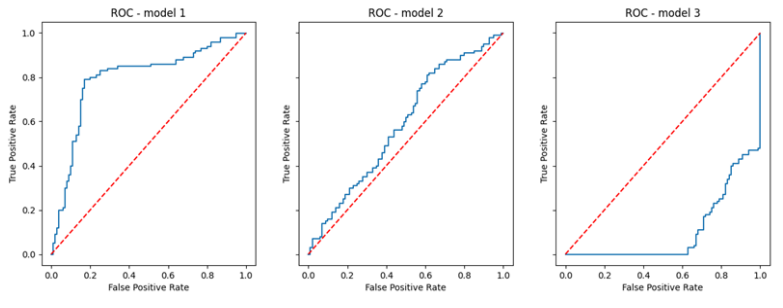


Za pomocą pakietu pandas:

- Załaduj pliki do pojedynczej ramki danych.
- Sprawdź typy danych w poszczególnych kolumnach.
- Znajdź liczbę unikalnych wartości w poszczególnych kolumnach.

3.
Dane są trzy wykresy krzywej ROC:

Zinterpretuj wykresy w kontekście jakości poszczególnych modeli.



4.
Model klasyfikacji binarnej zwrócił następujące prawdopodobieństwa:

```
TRUE VALUES:
[1, 1, 0, 0, 0, 0, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 0, 0, 0, 1]

PREDICTION SCORES:
[0.8, 0.08, 0.52, 0.77, 0.85, 0.01, 0.29, 0.7, 0.55, 0.52,
 0.43, 0.67, 0.75, 0.08, 0.32, 0.3, 0.02, 0.72, 0.82, 0.28]
```

Przy pomocy pakietów NumPy i scikit-learn oraz macierzy pomyłek ustal próg akceptacji (zaokrąglony do dwóch miejsc po przecinku), który zapewnia:

- najwyższy F1-score
- największą precyzję (precision)

5.
Przy pomocy podstawowych bibliotek Pythona napisz funkcję train_test_split(), przy zadanej proporcji zbioru treningowego.

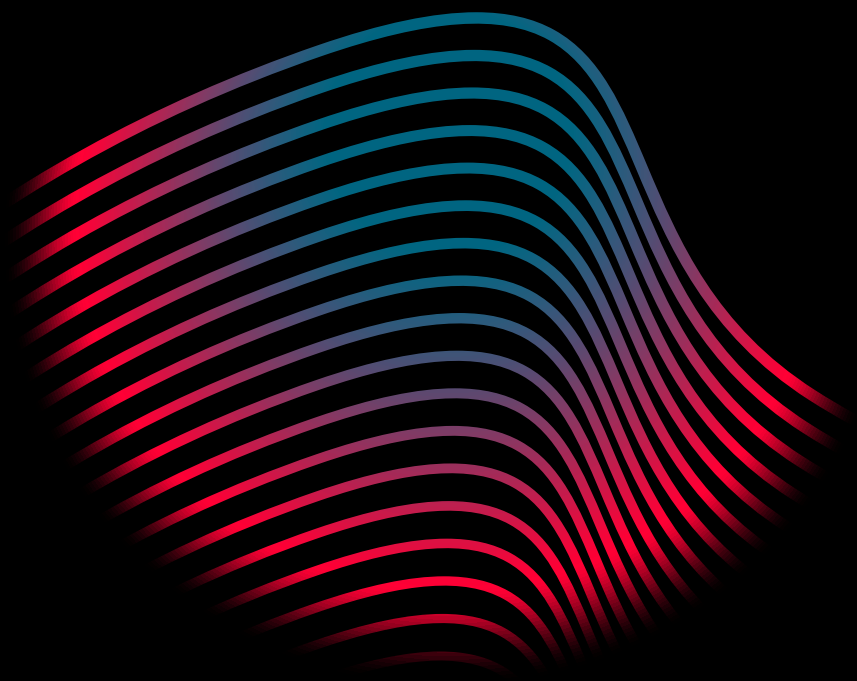
```
def train_test_split(x,y,train_size):
    ####################
    #       KOD        #
    ####################
    return x_train, y_train, x_test, y_test
```

W jaki sposób zmodyfikować powyższą funkcję tak, aby służyła do podziału zbioru według warunku nałożonego na wartości danej zmiennej, np. według dat?

6.
Przy pomocy pakietu TensorFlow lub PyTorch utwórz klasę realizującą model Wielowarstwowego Perceptronu:

```
class MLP(num_of_layers, layer_sizes):
```

Samsung R&D Institute Poland



Samsung Tech Guides