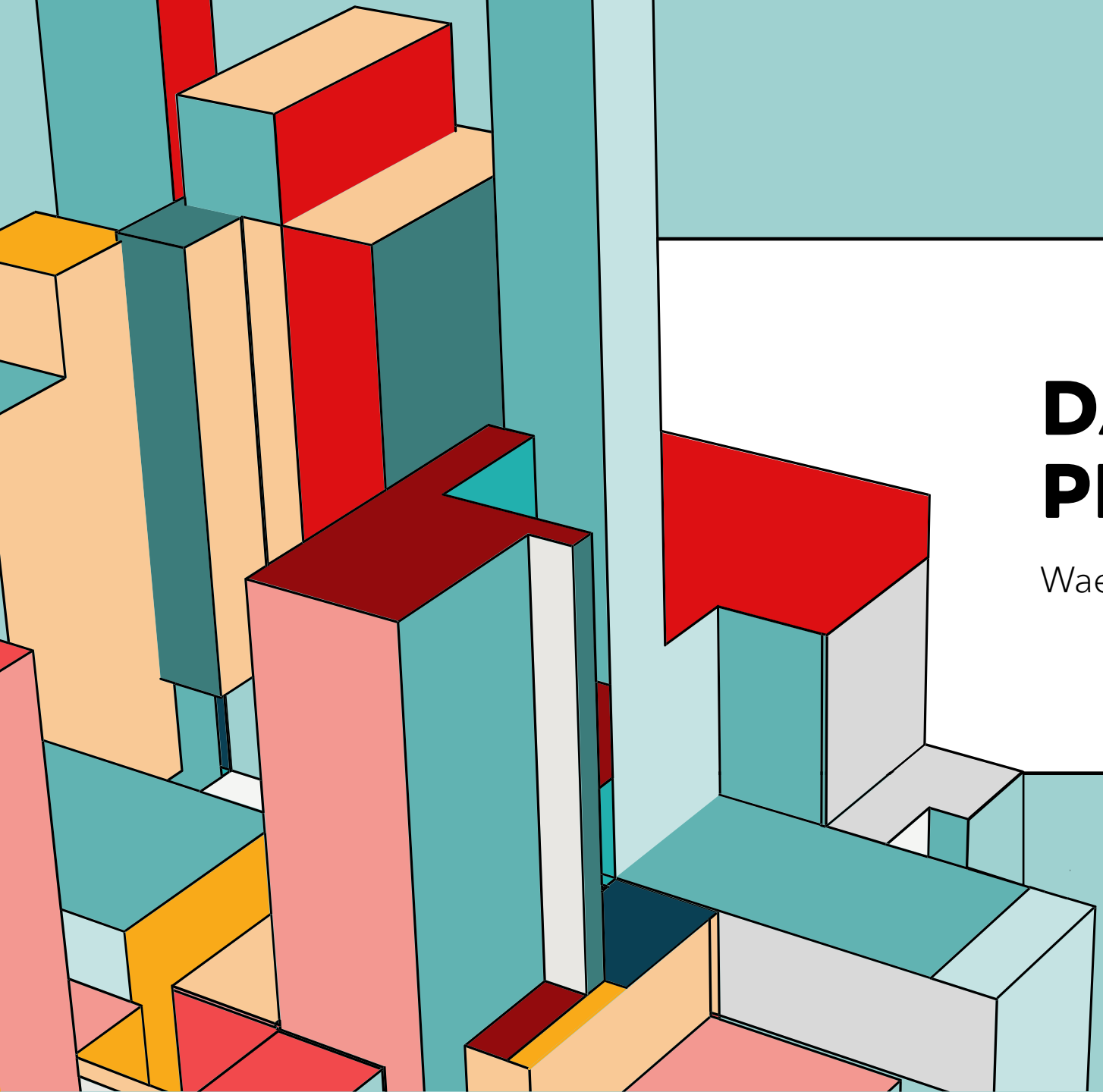


DATA PROFILES PREDICTOR

Wael SAIDENI

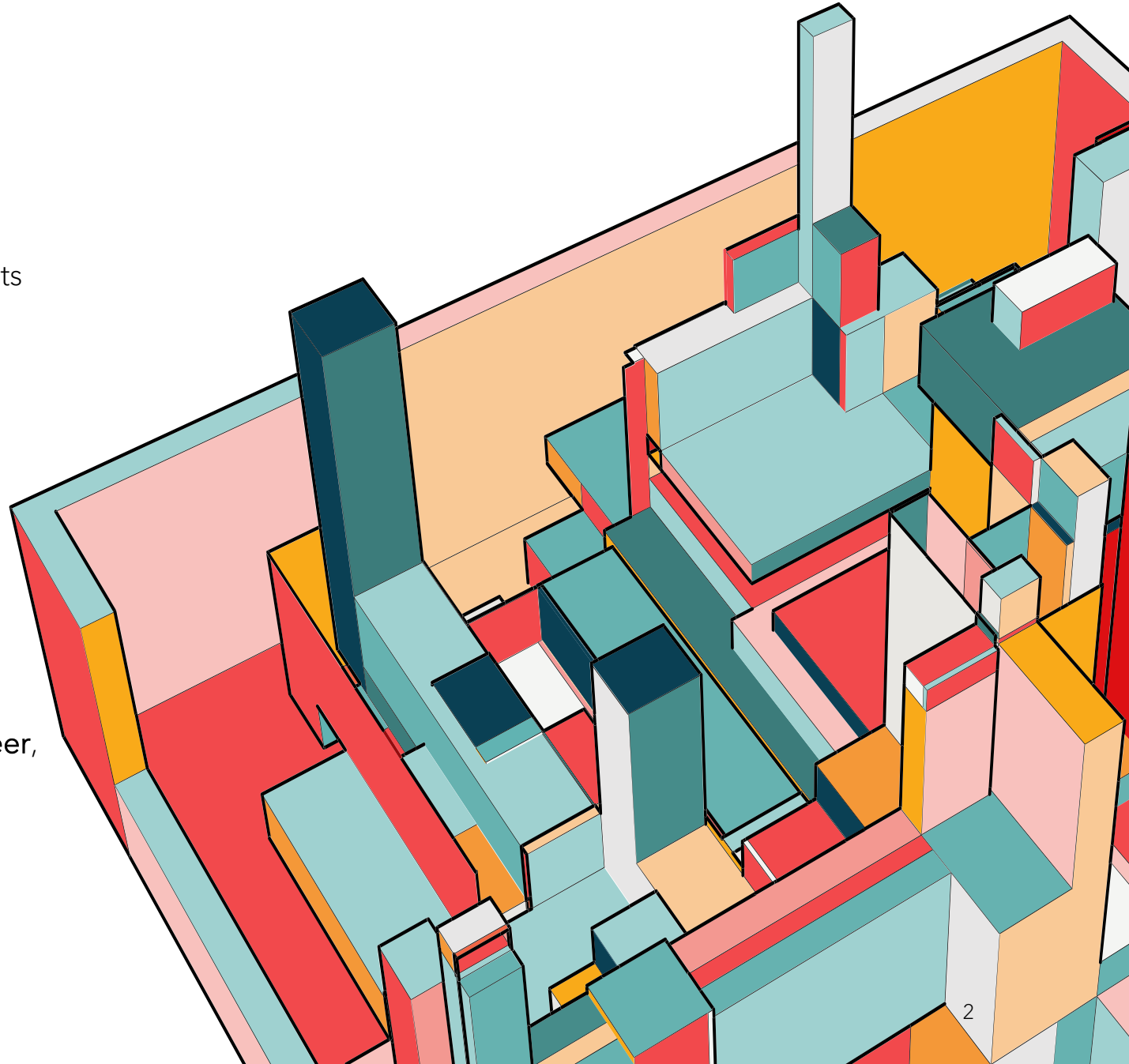
CONTEXT

This project aims to help companies and HR departments to classify data profiles and resumes when applying to data related positions.

The idea is to make sense of confusing data science job postings.

Our use case is to predict the data related profiles from some information about the data specialist such as the **experience**, the **technologies**, the **diploma**, etc.

The data specialist can be a **data scientist**, a **data engineer**, a **data architect** or a **lead data scientist**.



WHY DO YOU NEED THIS APP

Of course you do ☺

For recuiters

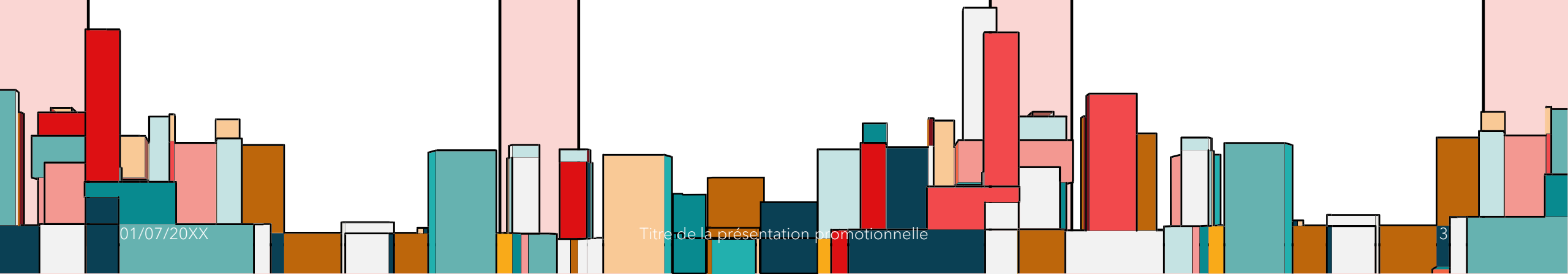
To find the best position for the best profile

For businesses

All businesses know what they want. However, what an organization actually needs is less defined.

For data specialists

To increase the chances of getting hired

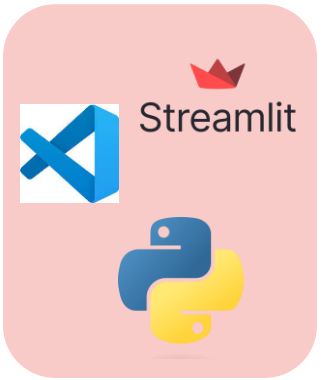


OVERVIEW OF THE TECHNICAL SOLUTION

Step1: Data Visualization with Vscode Jupyter

Step2: Pre-processing

Step3: Model development



Step4: Deployment

Step5: Monitoring

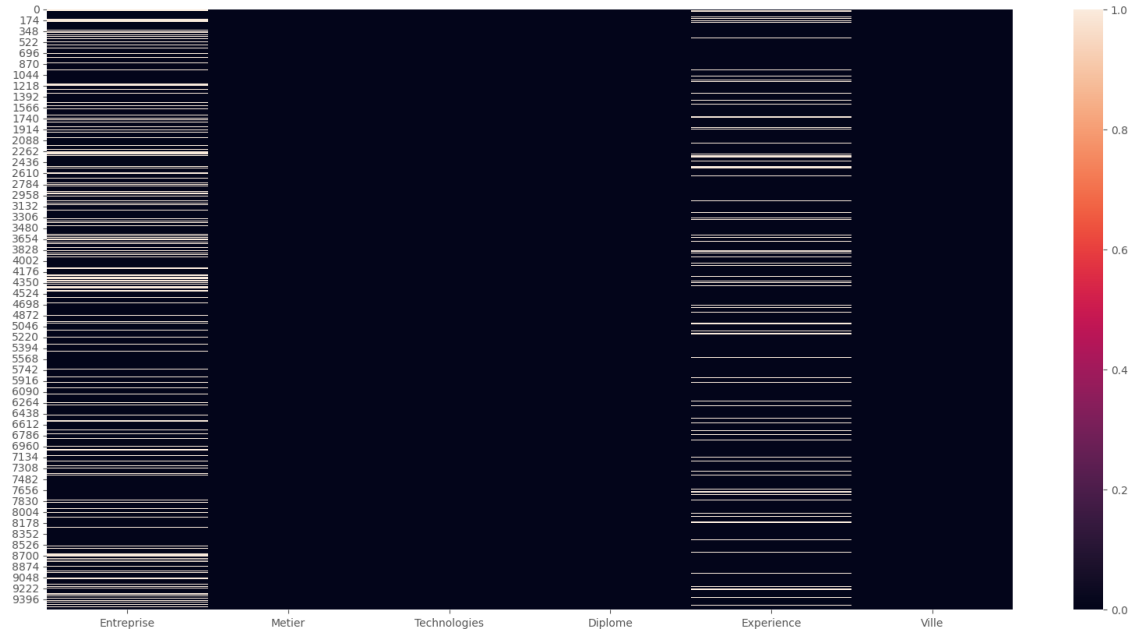
DATA INSIGHTS

Input tabluar data looks like:

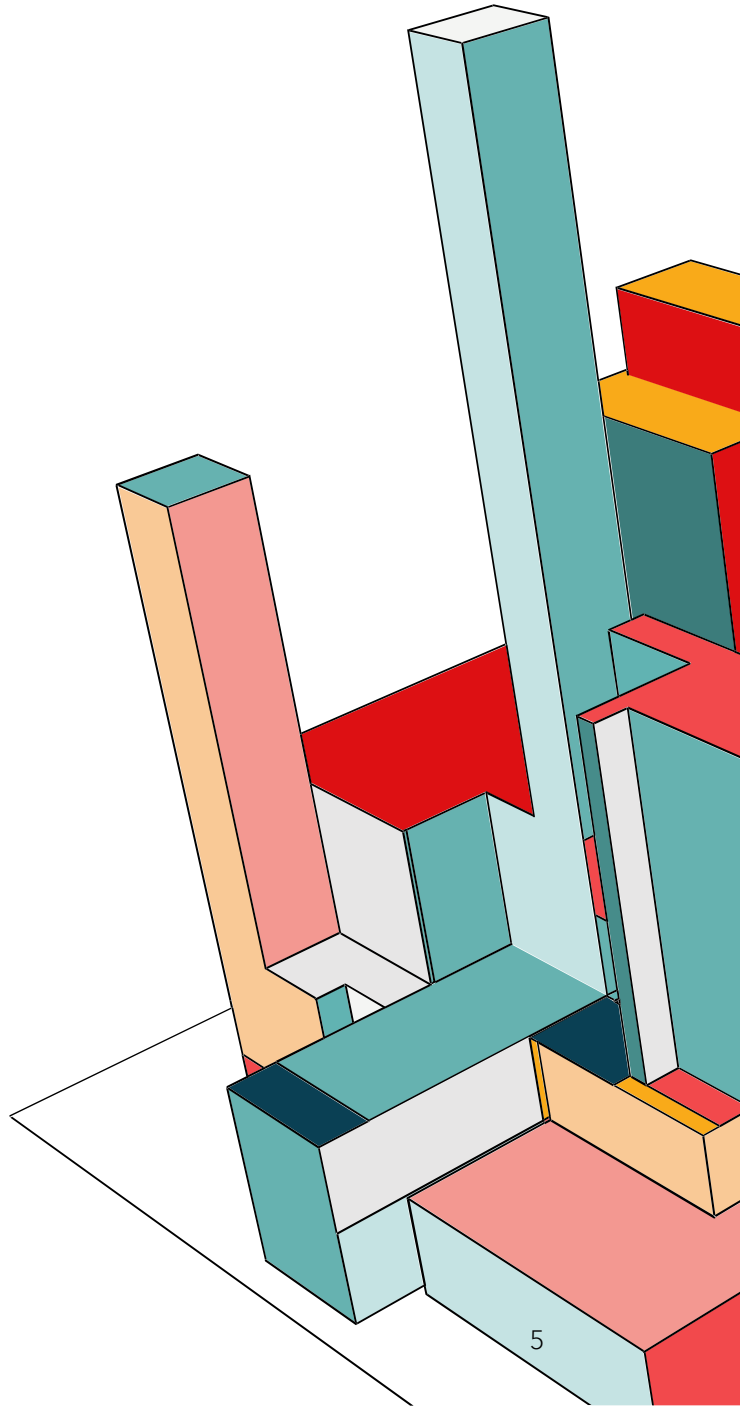
	Entreprise	Metier	Technologies	Diplome	Experience	Ville
0	Sanofi	Data scientist	Matlab/Python/Pyspark/Scikit-learn/Tensorflow	Master	1	Paris
1	Massachusetts General Hospital(MGH)	Data architecte	Python/Java/Scala/MongoDB	Master	3	Marseille
2	NaN	Lead data scientist	SPSS/SQL/Teradata/R/Python/Tensorflow/scikit-l...	Master	3	Nantes
3	Ann & Robert H. Lurie Children's Hospital of C...	Data scientist	C/C++/Java/Python	Master	1,5	Marseille
4	NaN	Data scientist	Matlab/Python/C++/numpy/Tensorflow/scikit-learn	Phd	NaN	Bordeaux

9562 lines ; 6 columns

Missing Values: 1926 on "Entreprise"; 1031 on "Experience"



The heatmap to visualize missing values

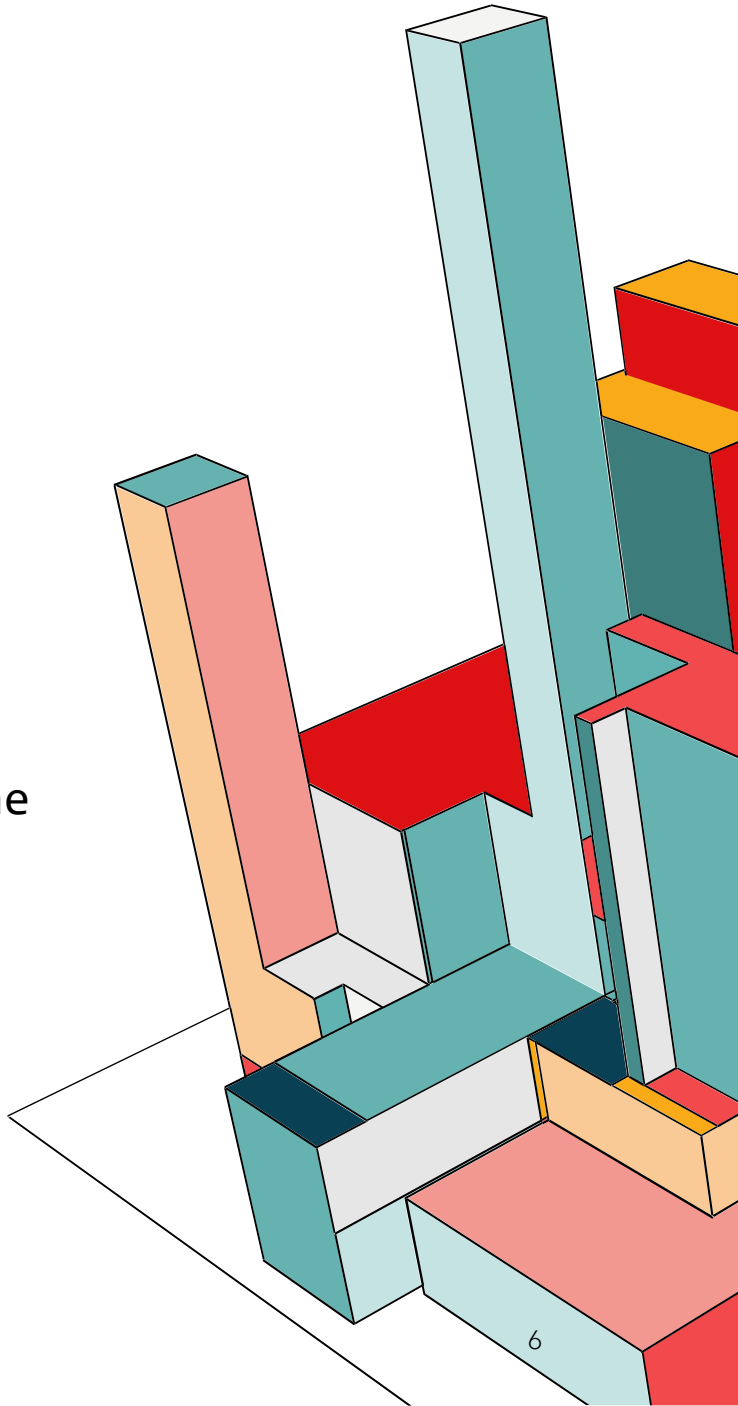


DATA INSIGHTS

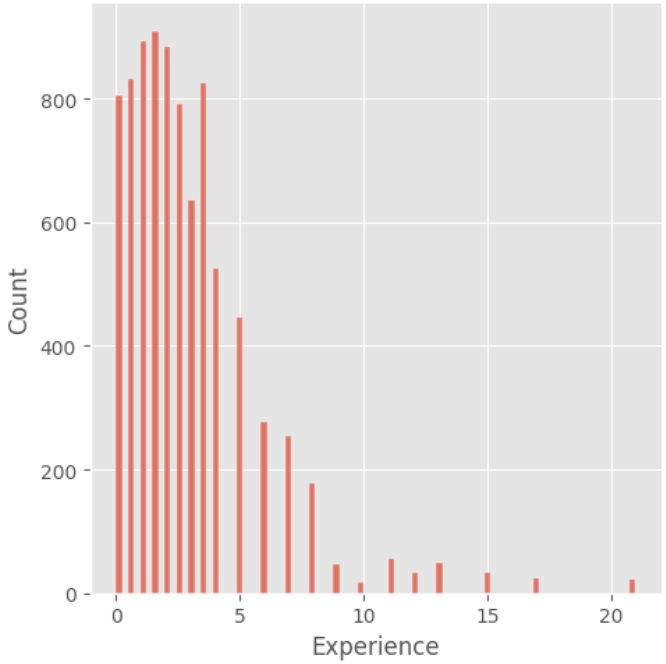
```
df['Metier'].value_counts()
```

```
Data scientist          3865
Data engineer           2347
Data architecte         2123
Lead data scientist     1227
Name: Metier, dtype: int64
```

We have imbalanced data if we will use Metier as a target in the last question so we will use score F1, recall and precision as metrics

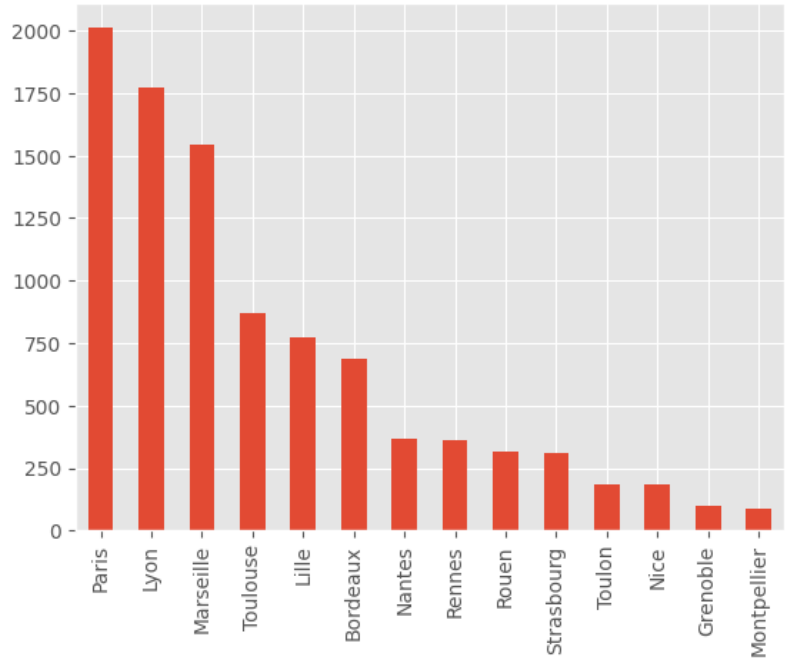


DATA INSIGHTS



We notice that we have a right-skewed distribution: extreme values are far from the peak on the high end more frequently than on the low

DATA INSIGHTS

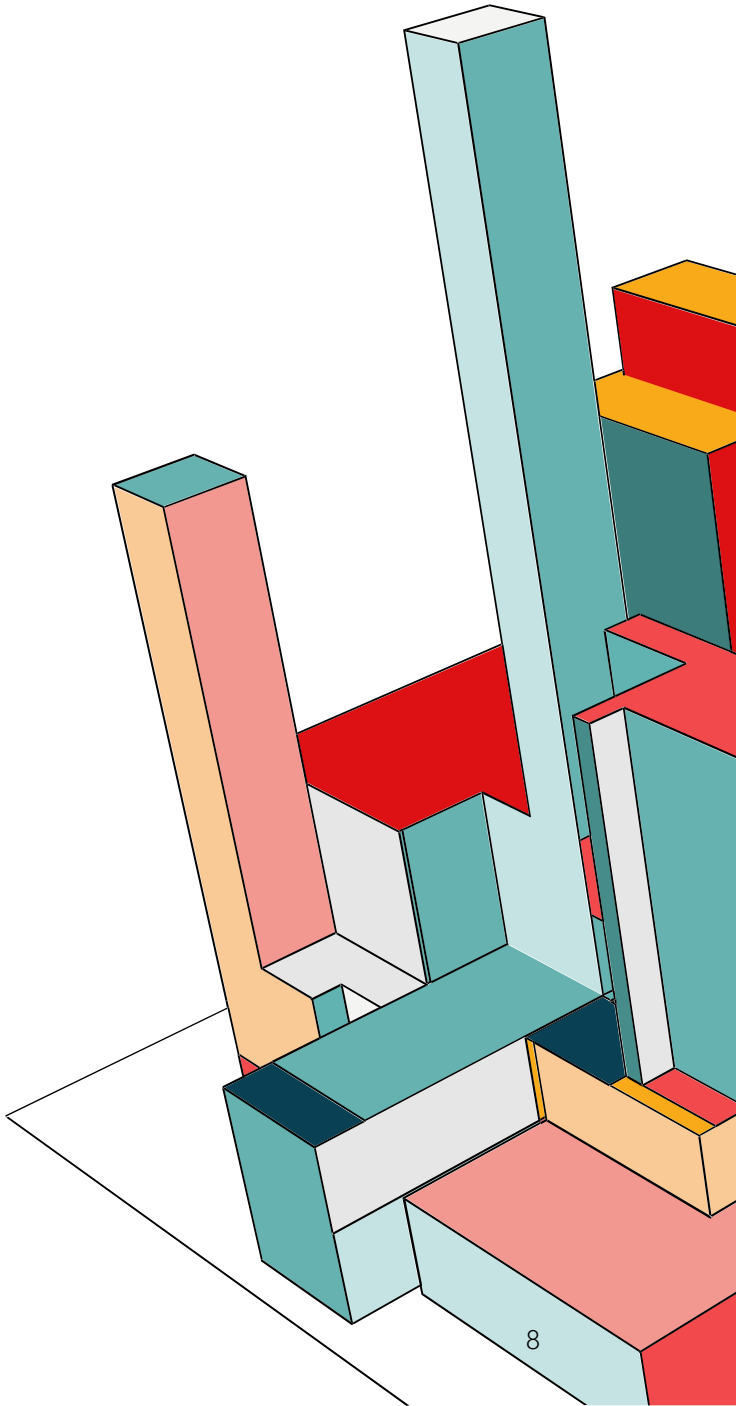



```
df['Ville'].value_counts()
```

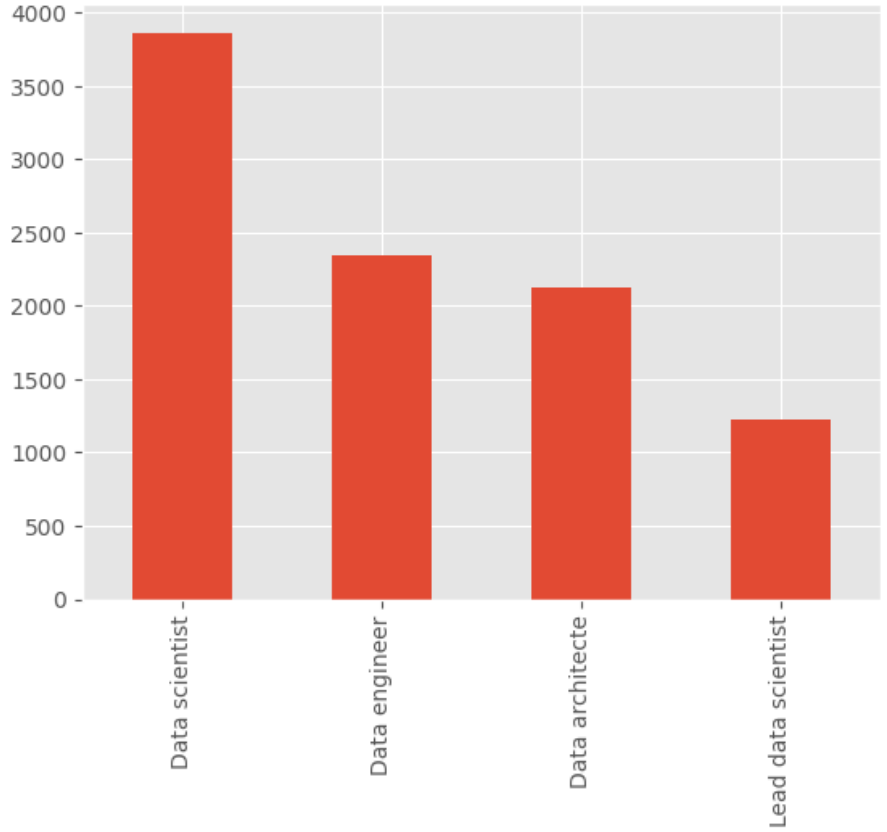
```
Paris          2011
Lyon           1775
Marseille      1544
Toulouse        869
Lille           771
Bordeaux        690
Nantes          365
Rennes          359
Rouen           315
Strasbourg      309
Toulon          186
Nice            183
Grenoble         98
Montpellier      87
Name: Ville, dtype: int64
```

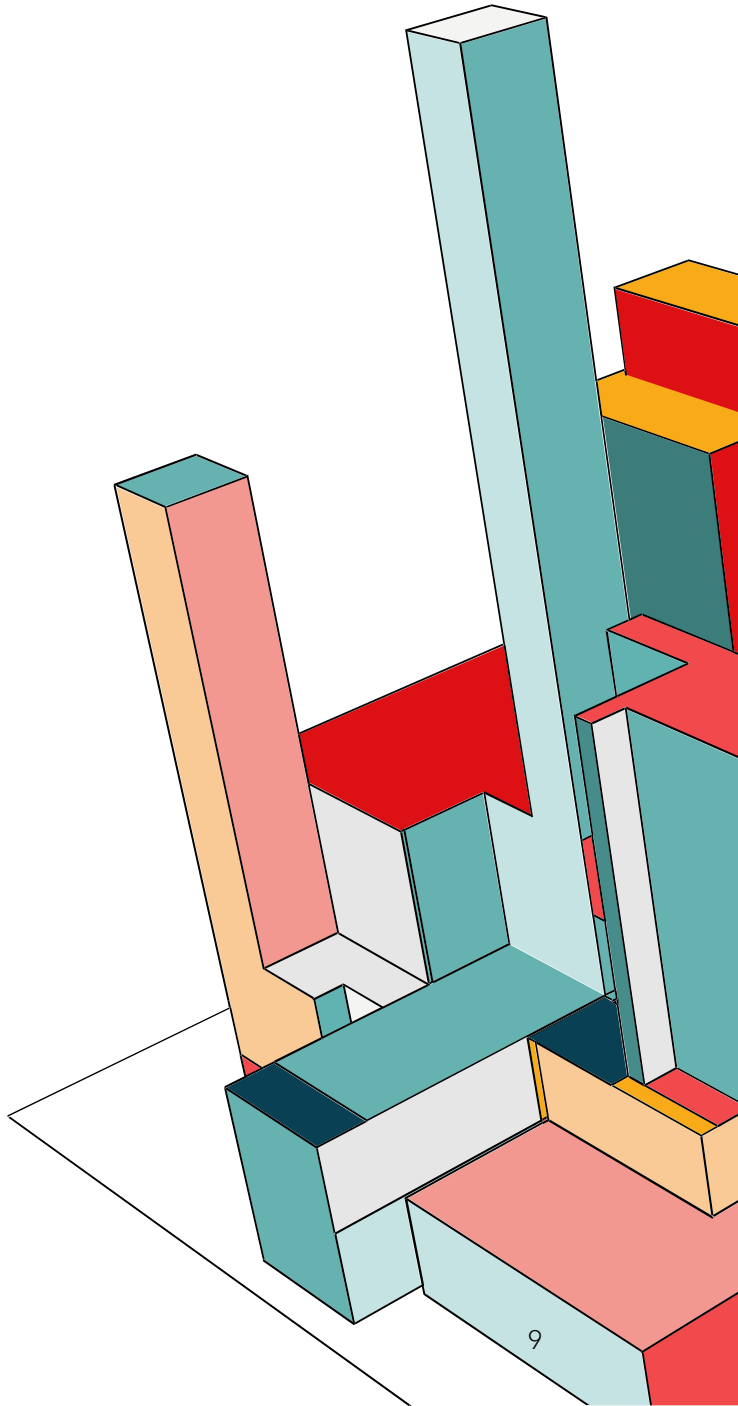
```
df['Entreprise'].value_counts()
```

```
Ball Aerospace                    598
Amazon.com                        105
KPMG                               95
Brigham & Women's Hospital(BWH)    94
McKinsey & Company                 88
                                  ...
Getty Images                        1
Transfix.io                         1
Lockstep, Inc.                      1
Projectline                         1
Ra Pharmaceutical                   1
Name: Entreprise, Length: 1320, dtype: int64
```

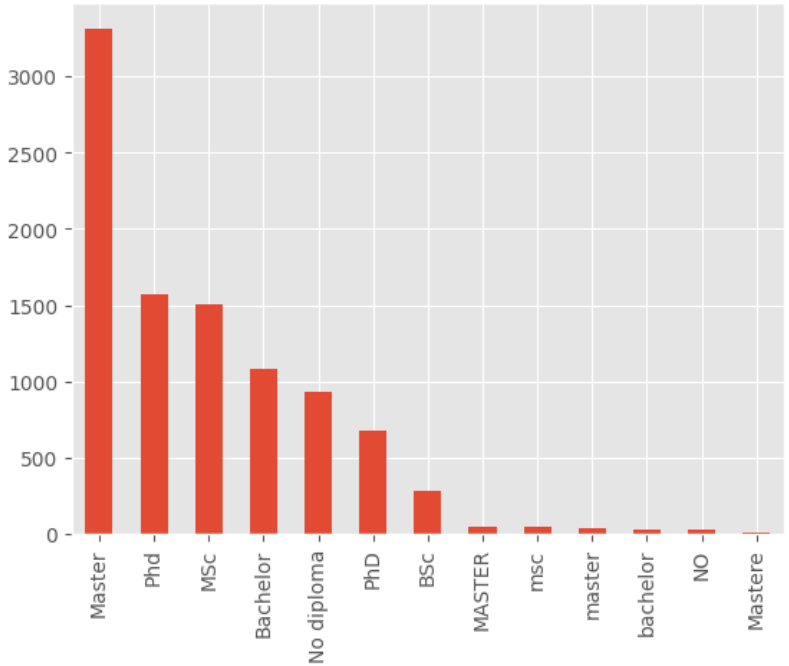


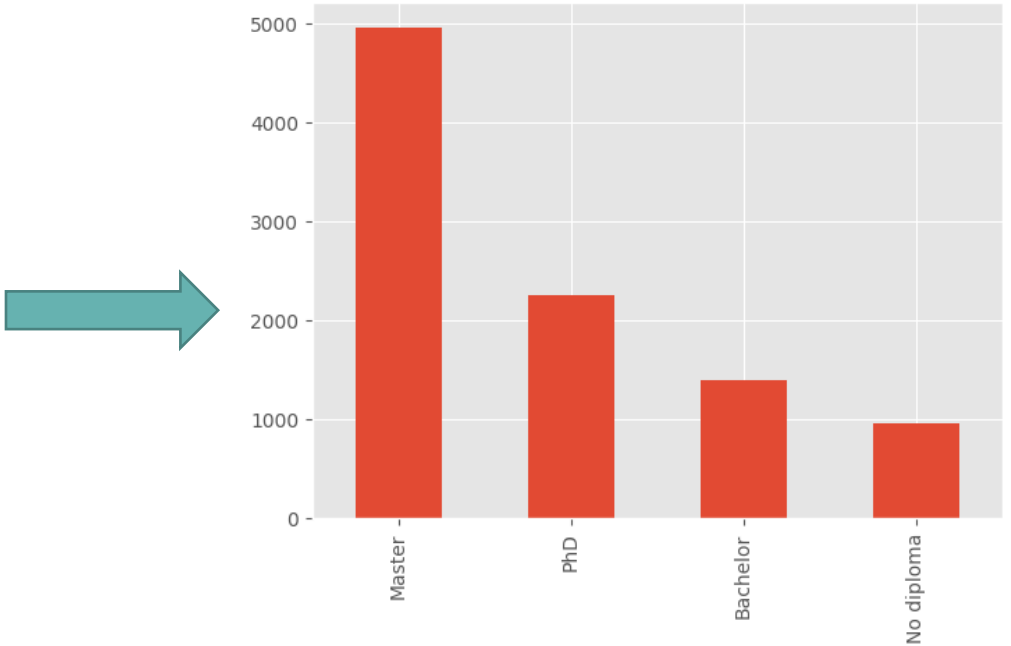
DATA INSIGHTS



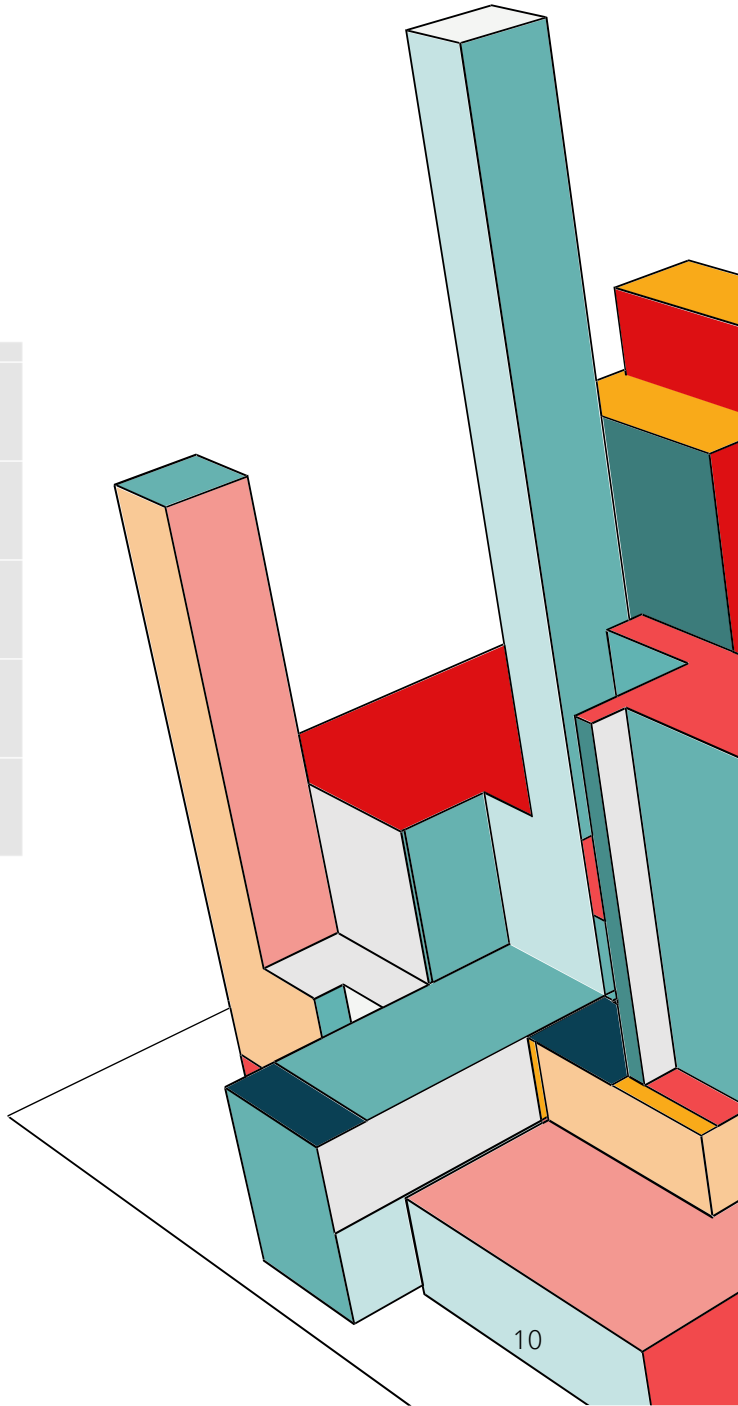


DATA INSIGHTS

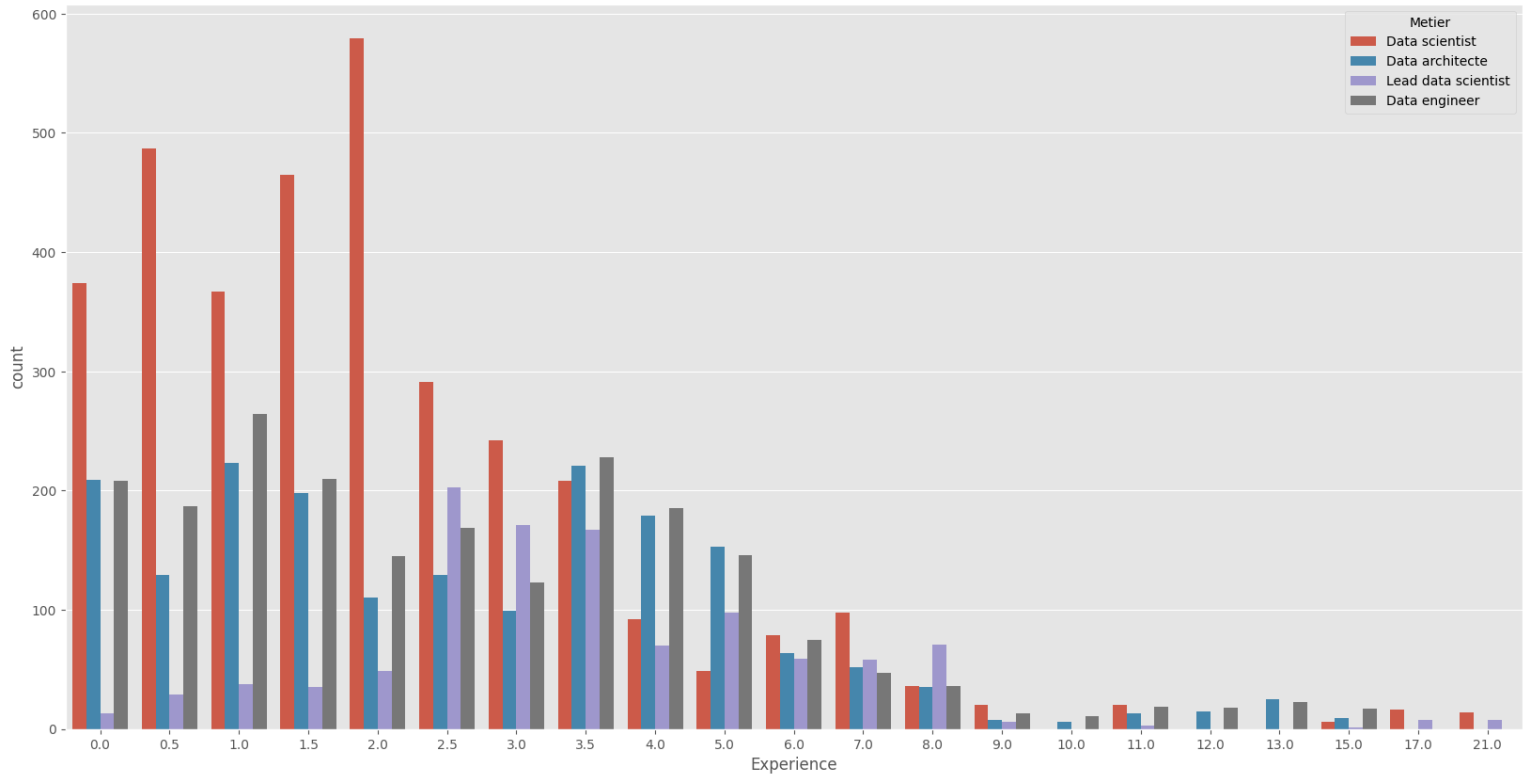




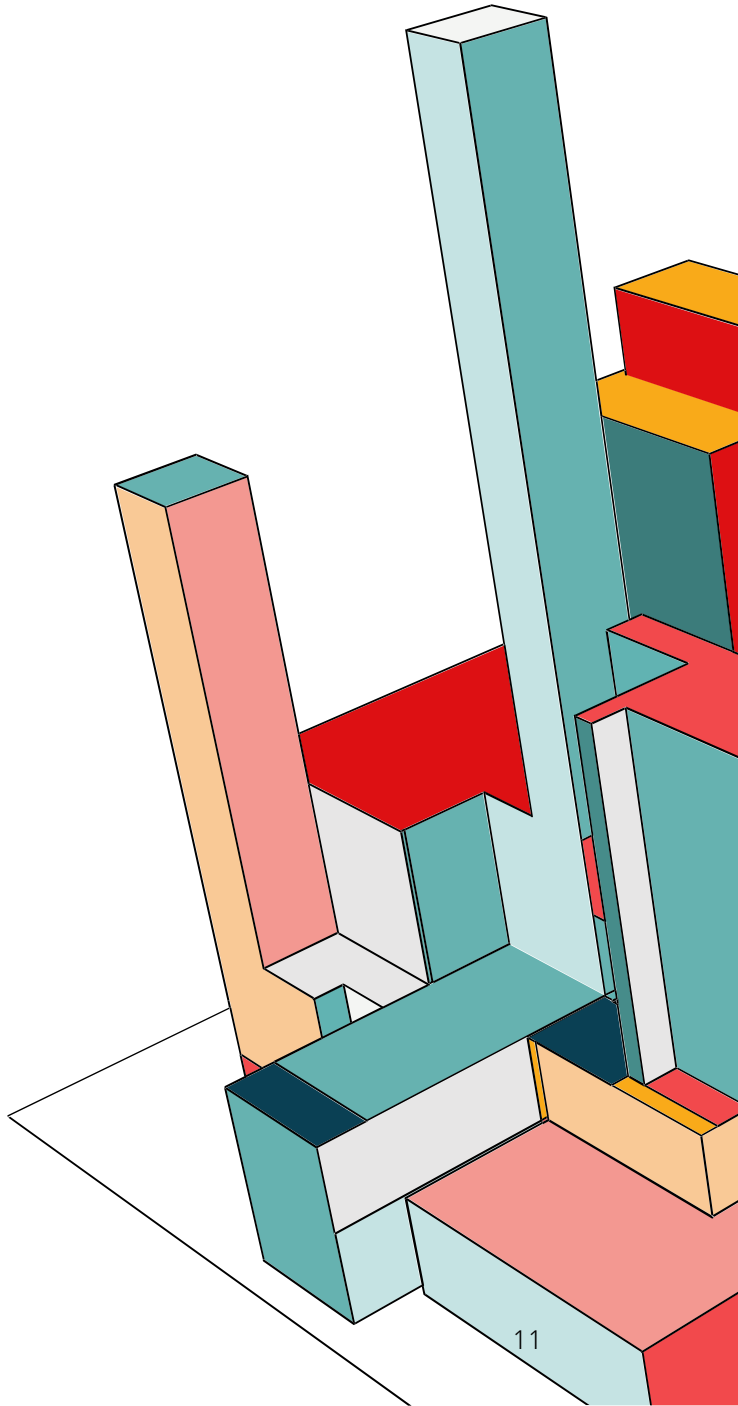
In this case, we normalize the spelling of the different diploma: PhD, Master, Bachelor and No diploma



DATA INSIGHTS



This graph enables to visualize the relationship between the feature "Experience" and the target "Metier". Clearly, Experience is an important feature but we can't define its obvious impact on our target.

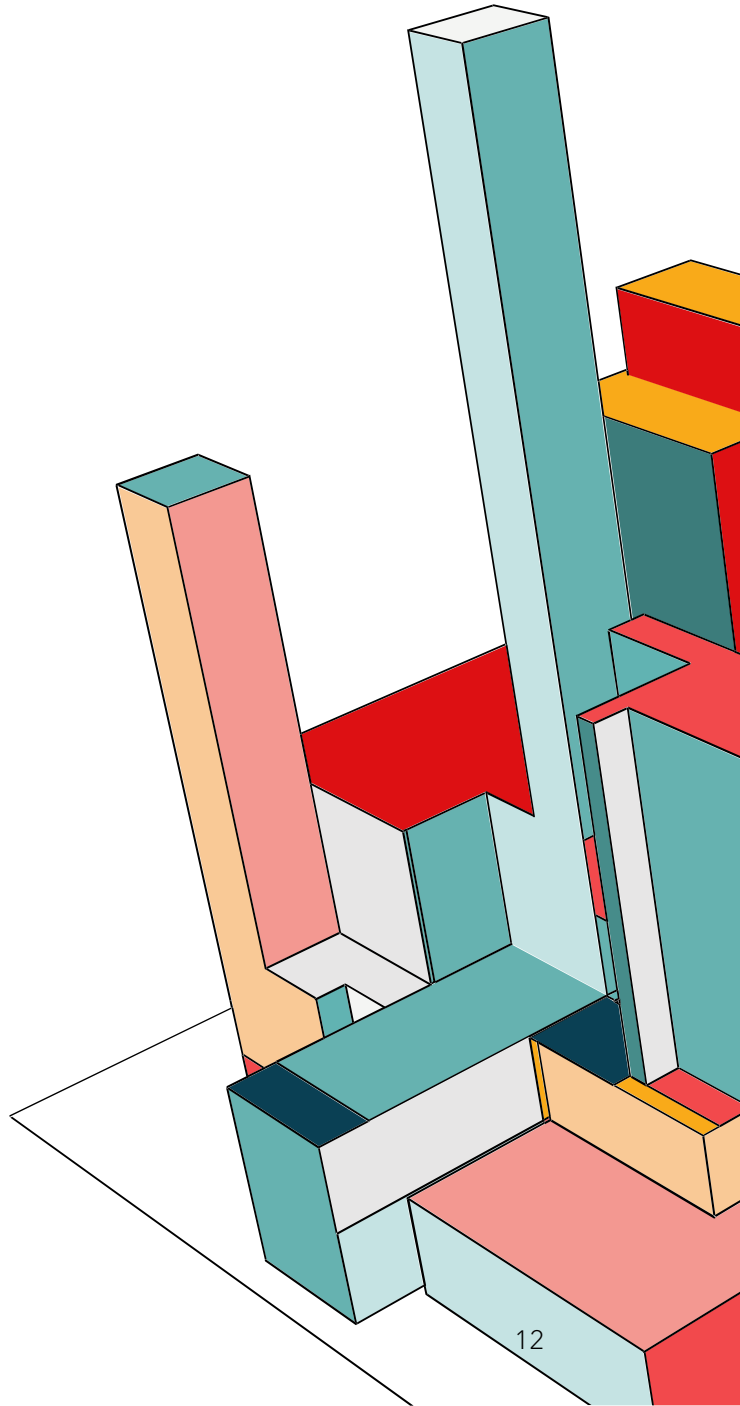


DATA INSIGHTS

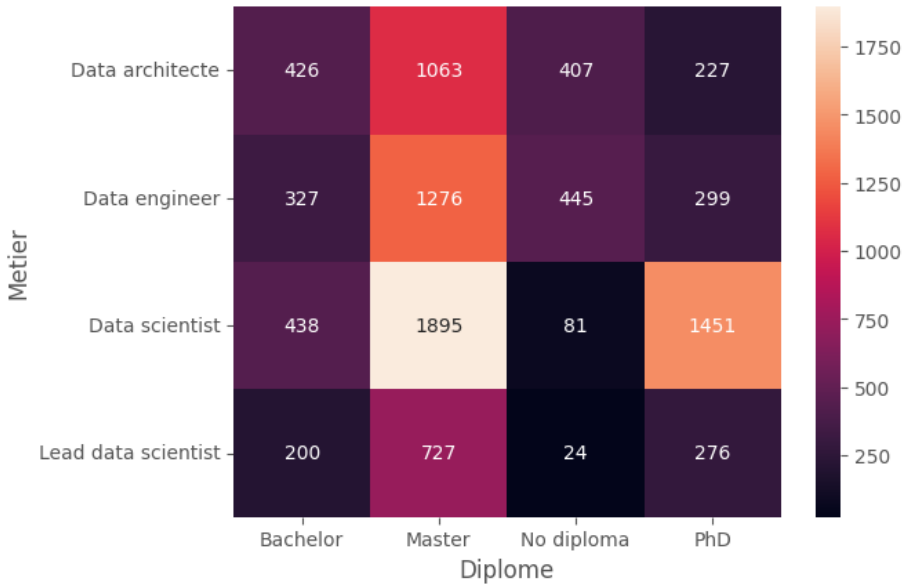
```
pd.crosstab(df['Metier'], df['Diplome'])
```

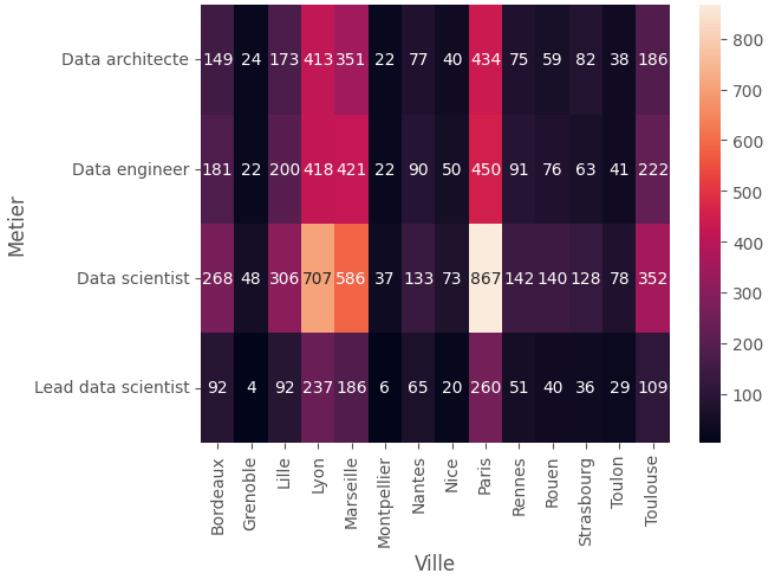
Metier \ Diplome	Bachelor	Master	No diploma	PhD
Data architecte	426	1063	407	227
Data engineer	327	1276	445	299
Data scientist	438	1895	81	1451
Lead data scientist	200	727	24	276

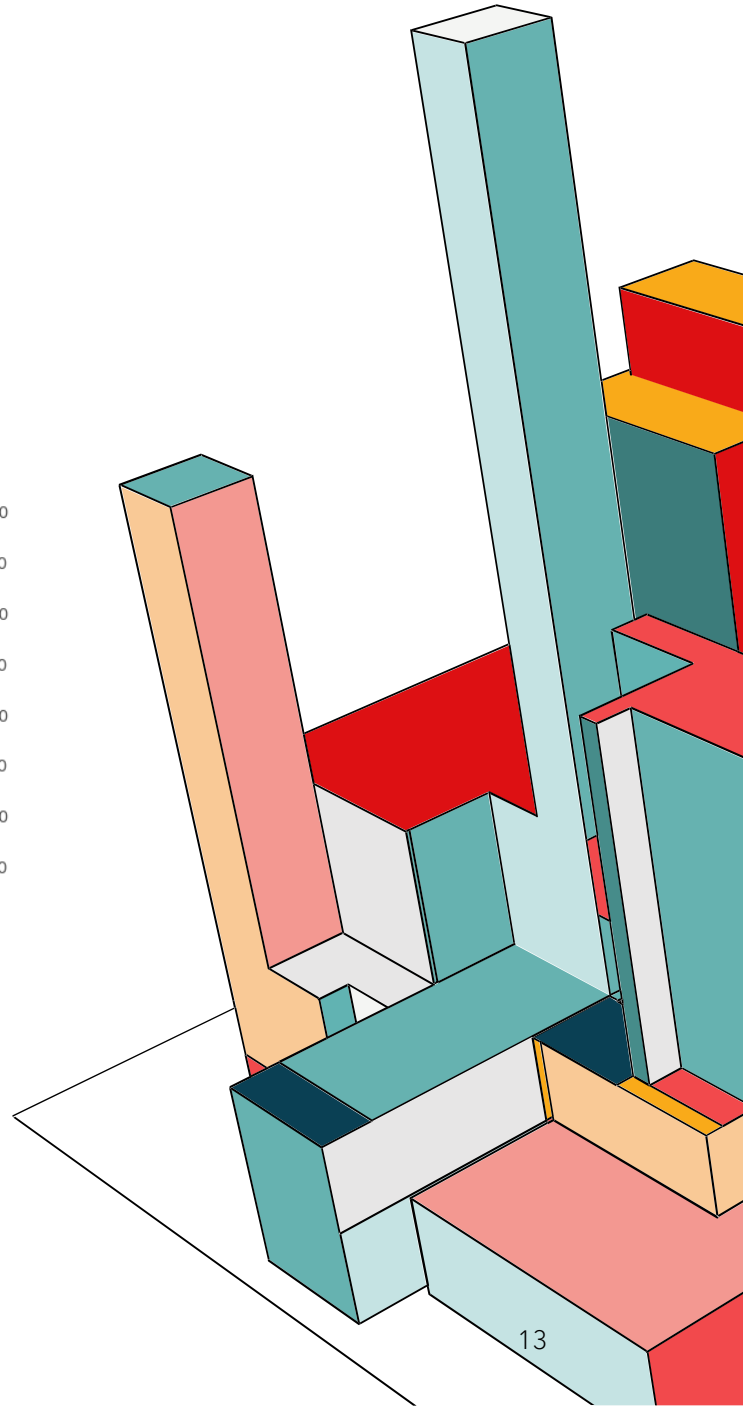
This table enables to visualize the cross tabulation between our target "Metier" and the categorical feature "Diplome". It outputs the frequency table of these features



DATA INSIGHTS





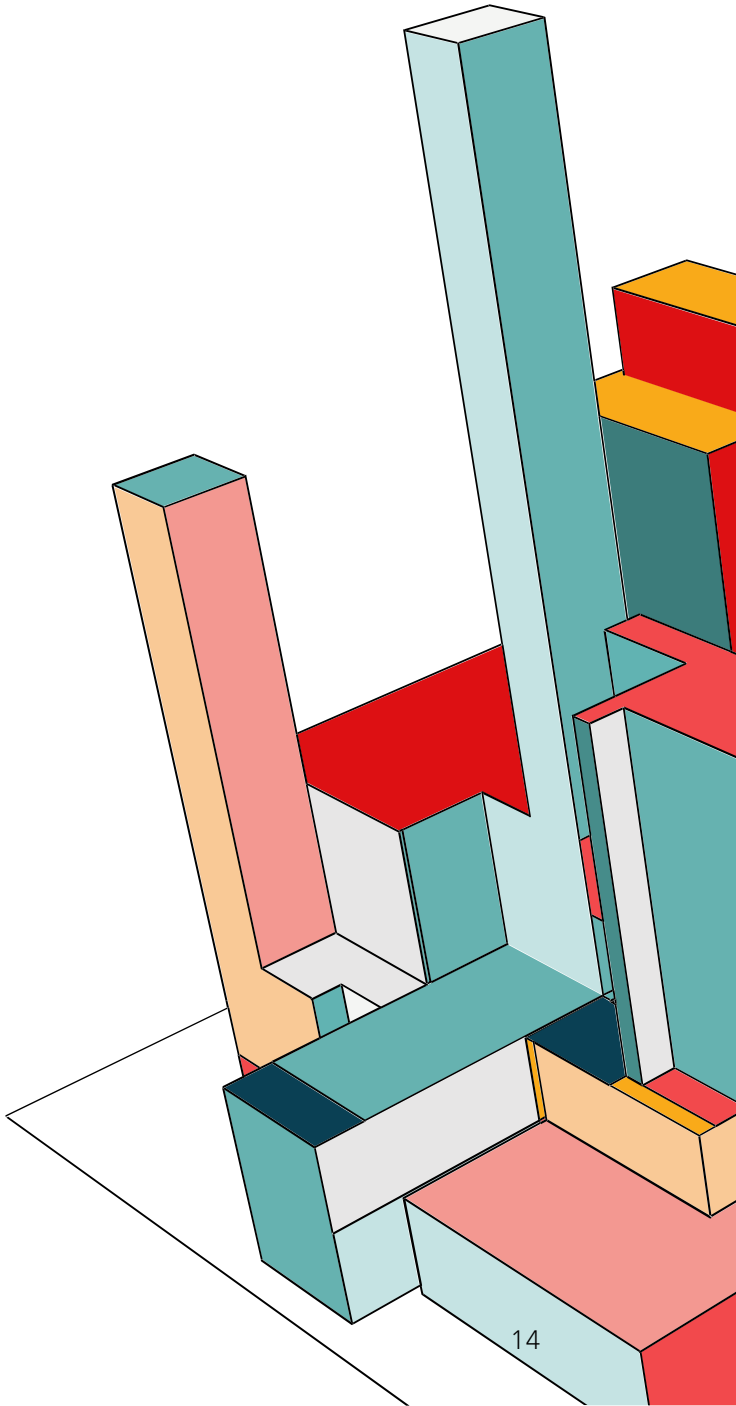


DATA INSIGHTS

```
for job in df['Metier'].unique():
    print(job, df[df['Metier']==job]['Experience'].isnull().sum())
```

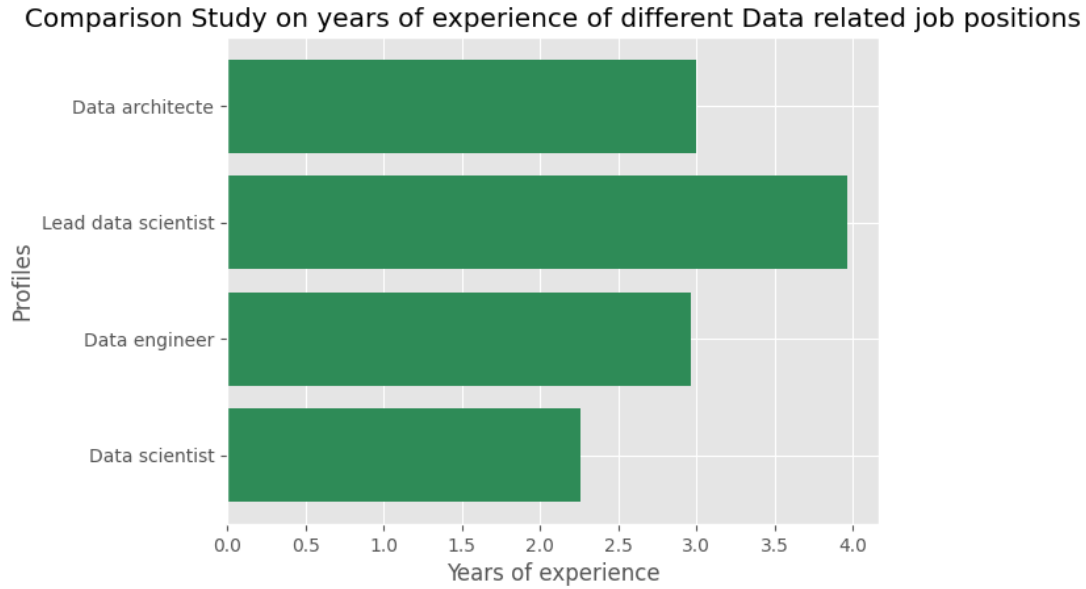
```
Data scientist 422
Data architecte 246
Lead data scientist 140
Data engineer 223
```

422 missing Experience values will be imputer for data scientists
246 missing Experience values will be imputer for data architects
140 missing Experience values will be imputer for Lead data scientists
223 missing Experience values will be imputer for data engineers



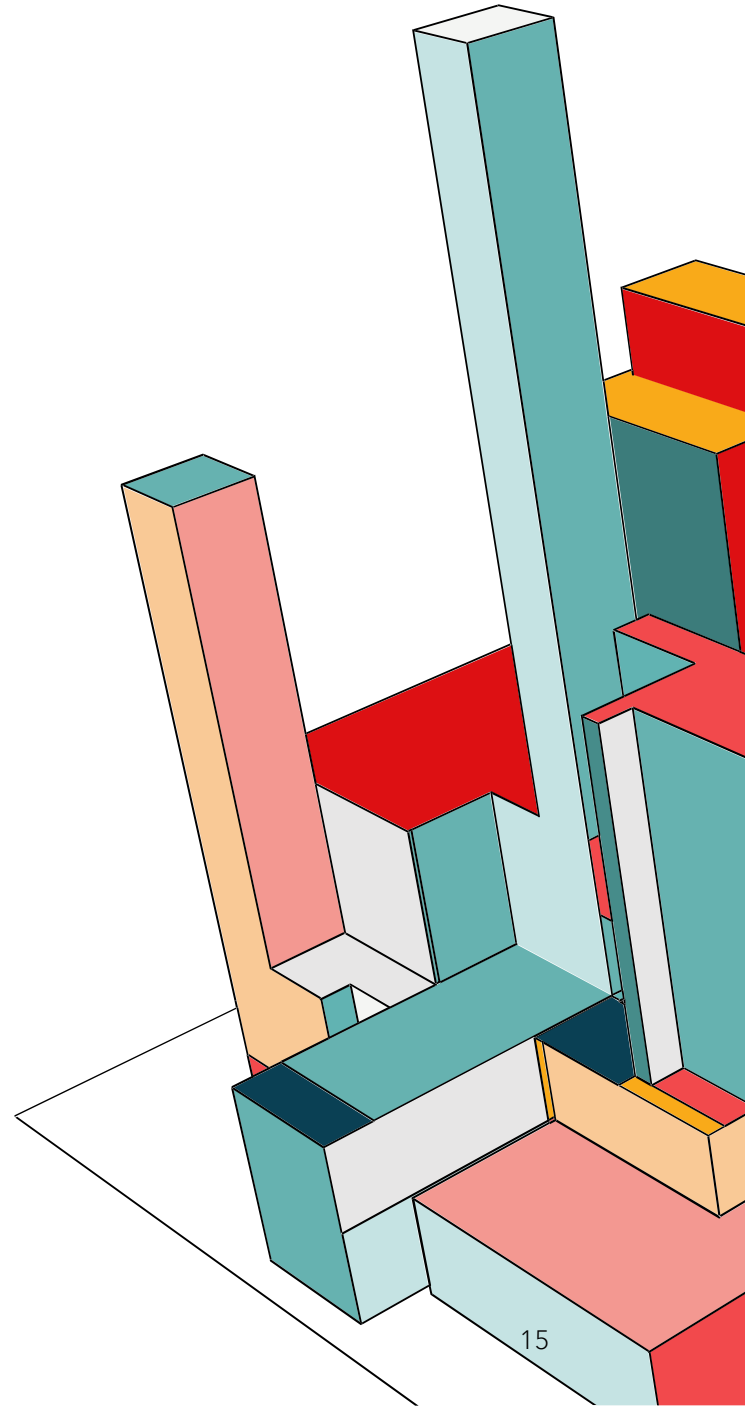
IMPUTE MISSING VALUES

Missing Experience values for each 'Metier' are replaced by the mean of other experience values of the same Metier




```
Profiles = ['Data scientist','Data engineer', 'Lead data scientist', 'Data architecte' ]
for profile in Profiles:
    print(profile, df[df['Metier']==profile]['Experience'].mean())
```

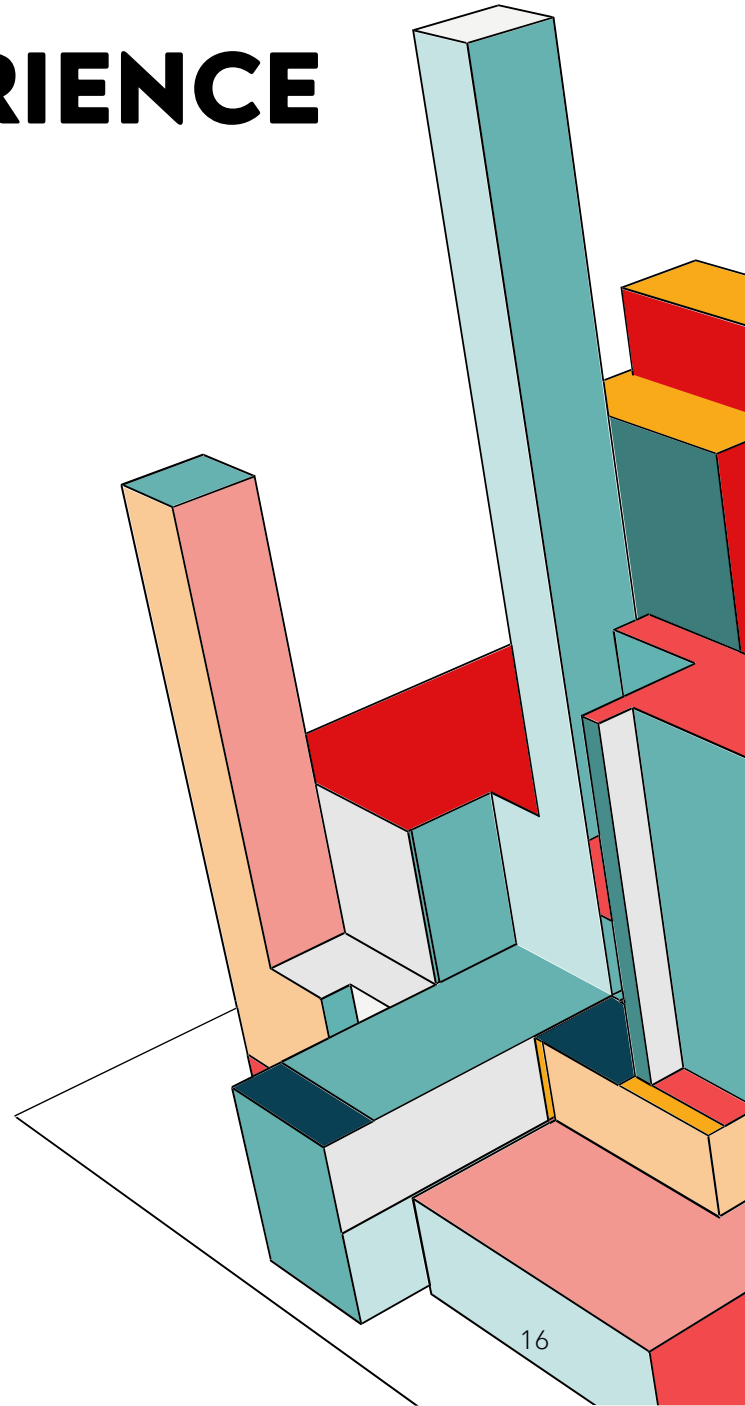
```
Data scientist 2.255155387743247
Data engineer 2.963747645951036
Lead data scientist 3.96688132474701
Data architecte 2.9949387320191794
```



CREATE ‘EXP_LEVEL’ FOR EXPERIENCE

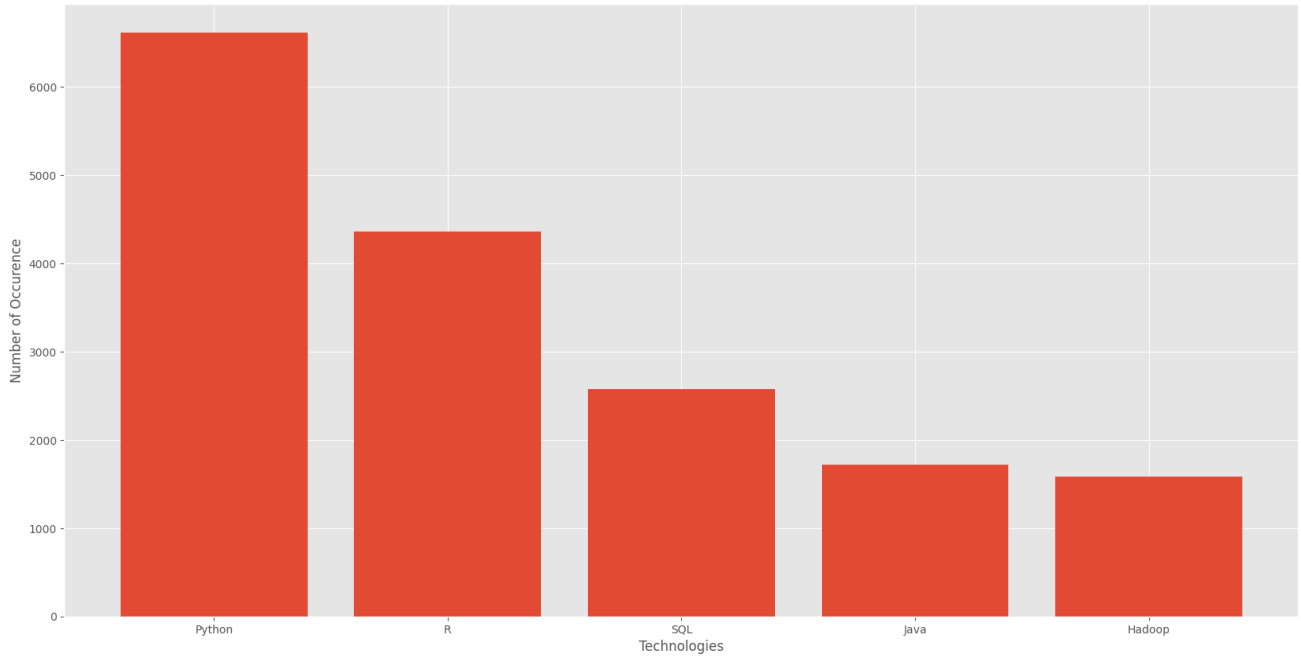
The idea is to improve the SNR (Signal to Noise Ratio): Fitting a model to bins reduces the impact that small fluctuates in the data has on the model, often small fluctuates are just noise. Each bin "smooths" out the fluctuates/noises in sections of the data.

In my implementation:
debutant: has less than 2 years of experience
Confirme: between 2 and 5 years of experience
Avance: between 5 and 8 years of experience
Expert: more than 8 years of experience



TECHNOLOGIES

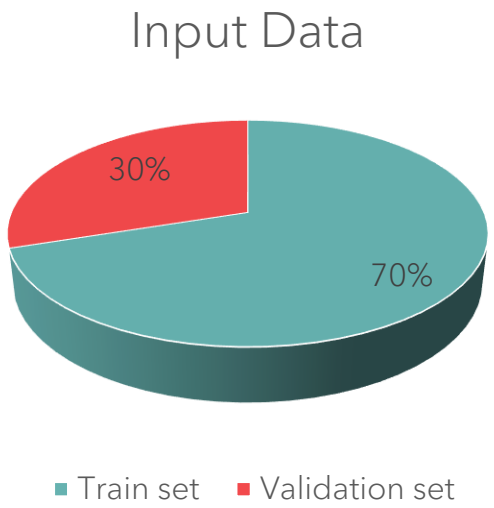
Most detected technologies among 61 technologies

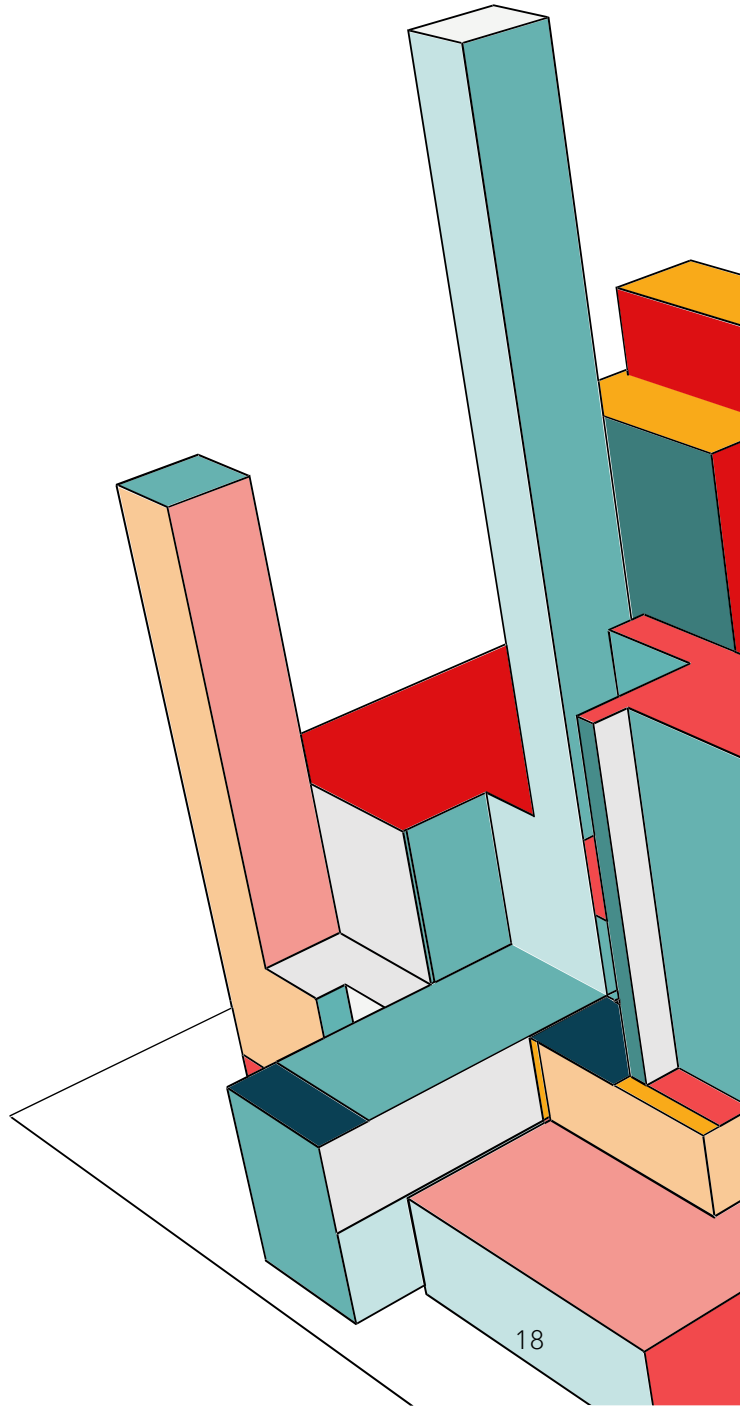


Python, R, SQL, Java and Hadoop are the most used technologies by data specialists
It is imporatant to highlight the fact that there are 61 different technologies used by the different profiles or exactly 60 since we have '' has a technology

HOLD-OUT

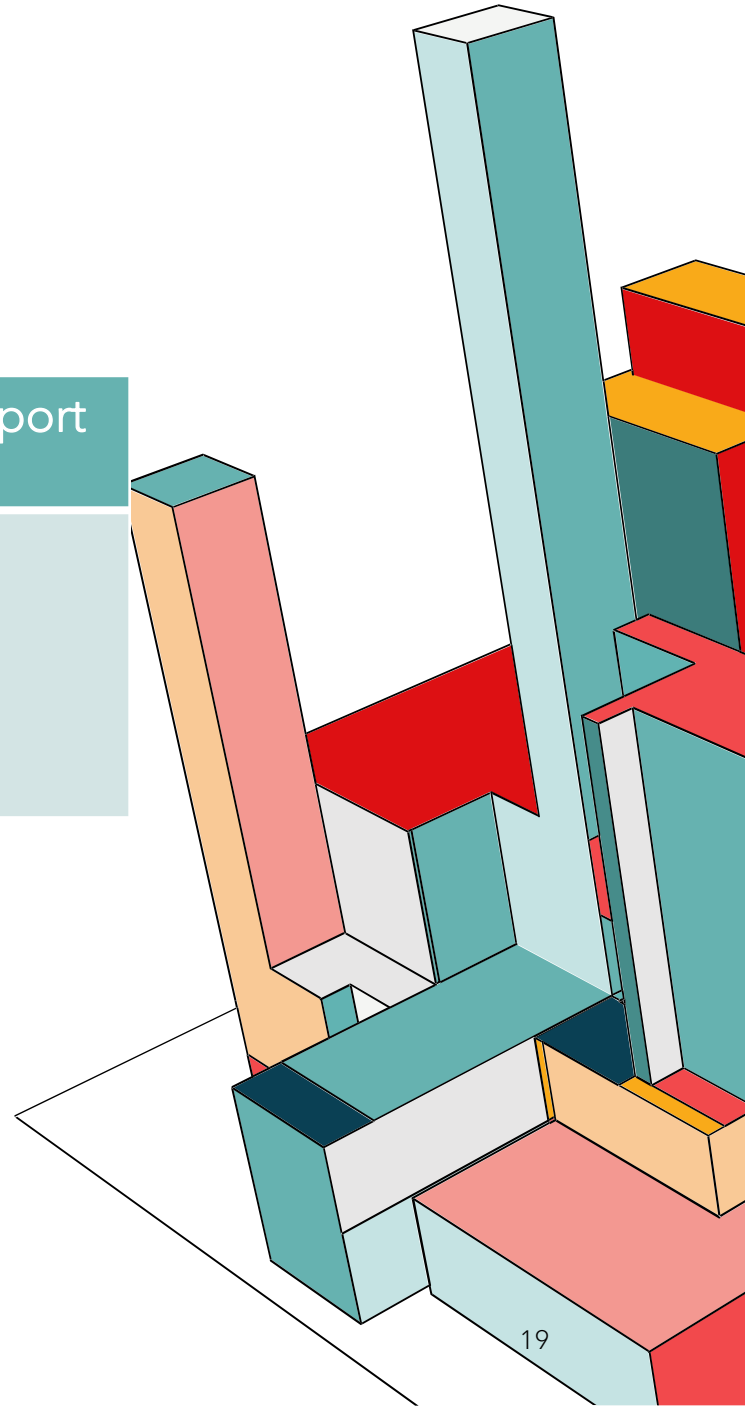
Split the entire dataset into a train and validation set.





CLASSIFICATION ALGORITHMS

Algorithms used that support categorical features	Algorithms used that do not support categorical features
CatBoost	KNeighborsClassifier Decision trees Random Forest AdaBoostClassifier SVM



CATBOOST IS CHOSEN TO BE DEPLOYED

```
print(classification_report(y_val, y_pred))
```

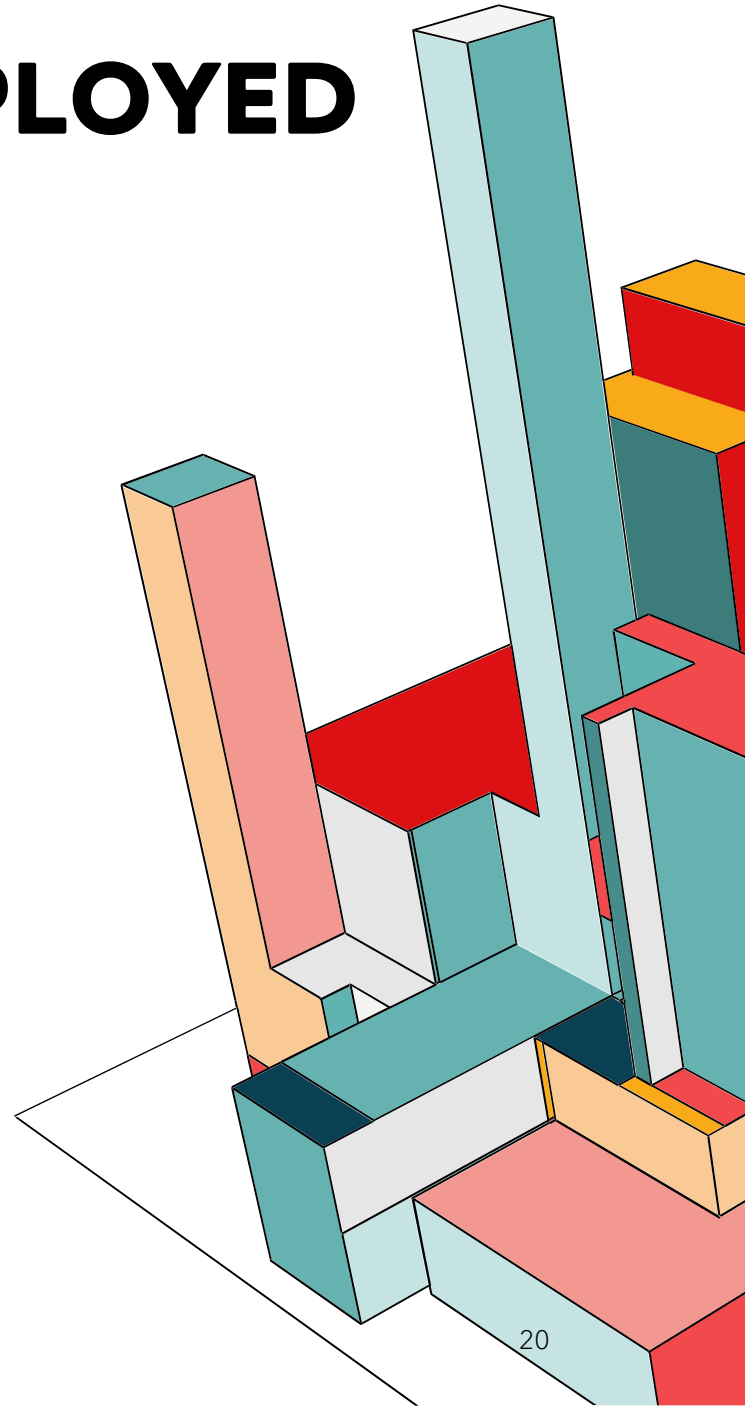
```
                    precision    recall  f1-score   support

    Data architecte       0.98      0.98      0.98       650
      Data engineer       1.00      1.00      1.00       682
     Data scientist       0.83      0.91      0.87      1171
Lead data scientist       0.61      0.42      0.49       366

           accuracy                           0.88      2869
          macro avg       0.86      0.83      0.84      2869
       weighted avg       0.88      0.88      0.88      2869
```

```
confusion_matrix(y_val, y_pred)
```

```
array([[ 638,    0,   12,    0],
       [   0,  682,    0,    0],
       [   7,    0, 1067,   97],
       [   3,    0,  211,  152]], dtype=int64)
```



CREATION OF AN ML WEB APP WITH STREAMLIT

Data profiles Predictor

Enter the characteristics of the profile:

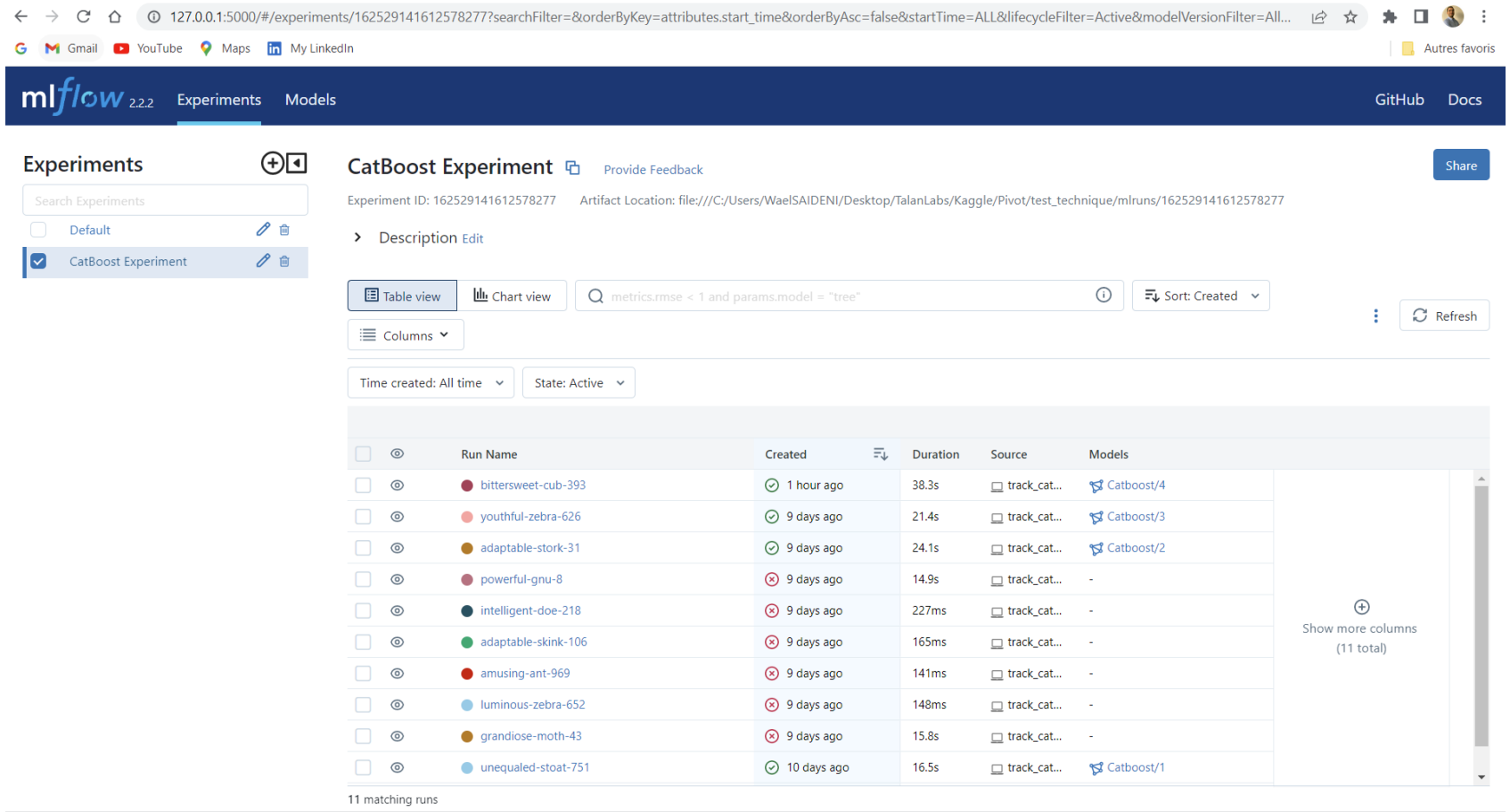
Entreprise

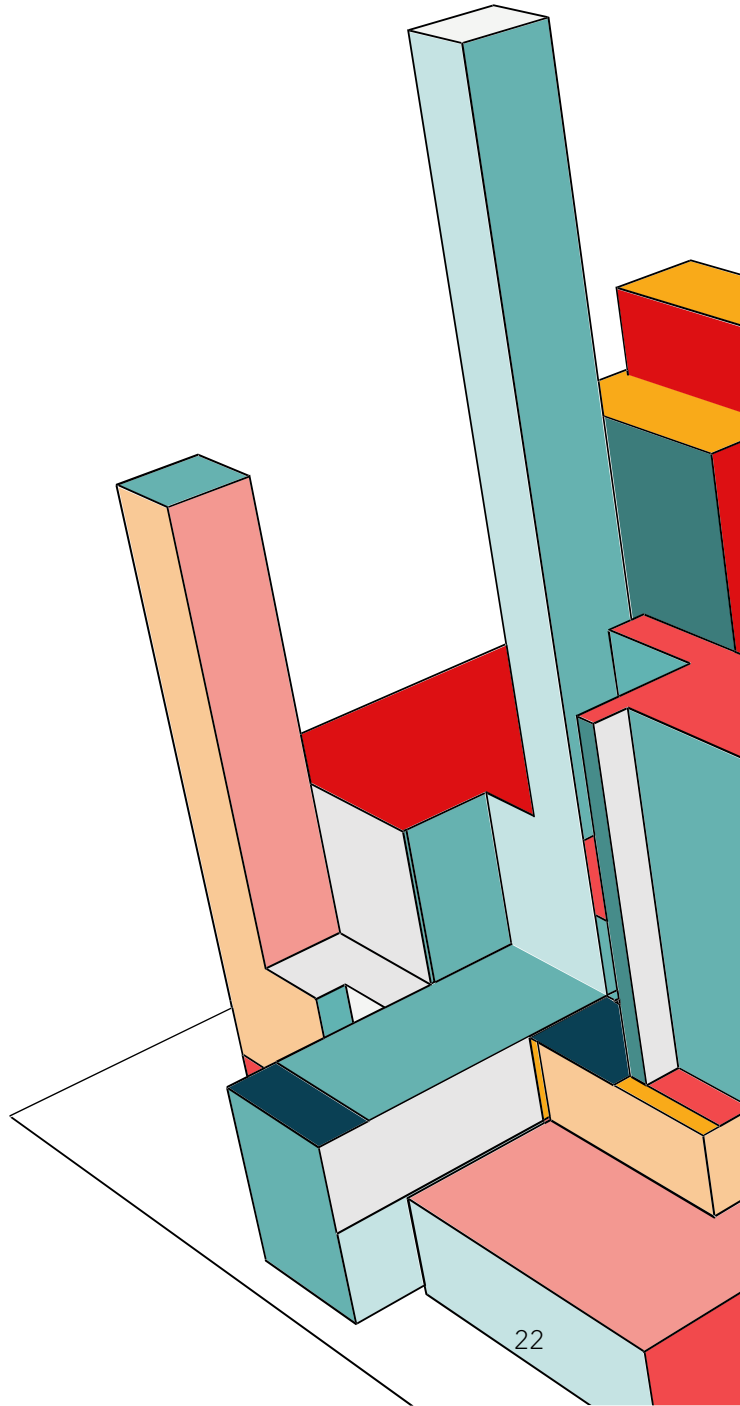
Diplome:

No diploma

MLFLOW

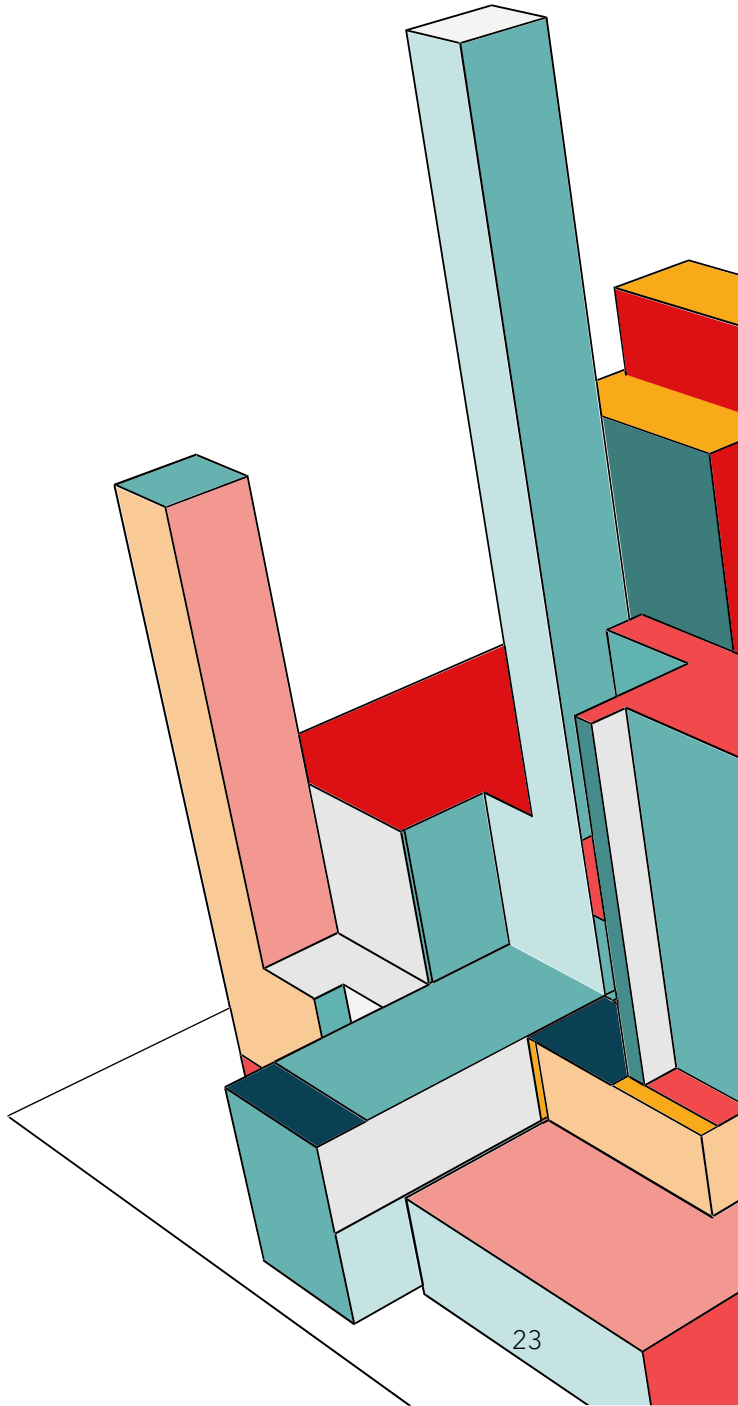
IT IS USED TO MANAGE THE ML LIFECYCLE, INCLUDING EXPERIMENTATION, REPRODUCIBILITY, DEPLOYMENT, AND A CENTRAL MODEL REGISTRY.



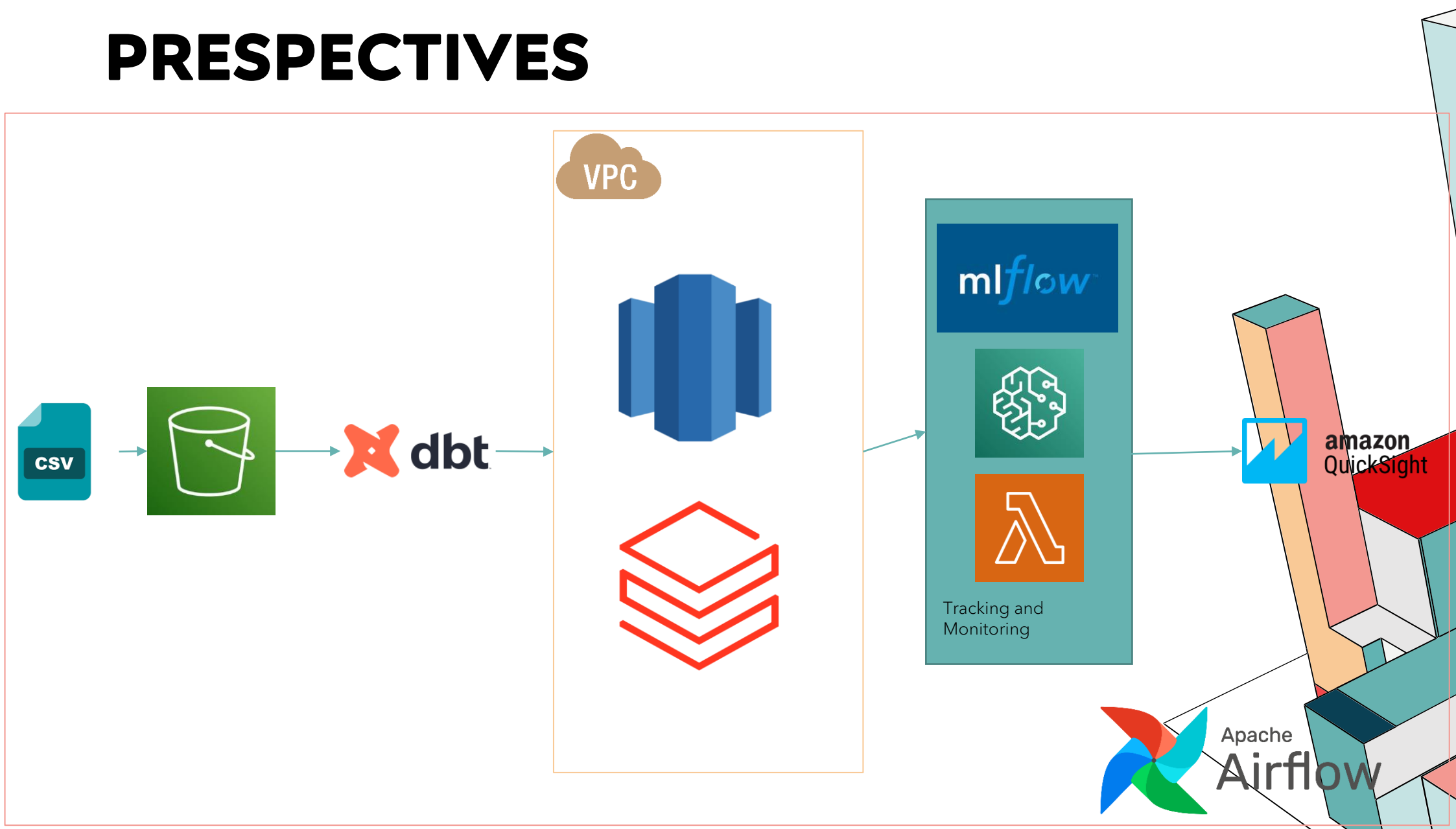


PERSPECTIVES

- Set up a cross validation strategy based on Stratified-Kfold since we are dealing with an imbalanced data
- Use MLFlow to track the hyperparameters tunning experimentations
- Design and deploy a complete data pipeline for the web app



PRESPECTIVES



THANK YOU

Wael SAIDENI