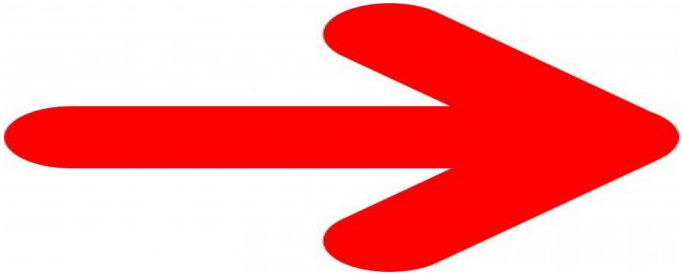
Présentation

Vivien Silvert
Farid Berrabah

La mission

Suite à un Turn-over trop important et grâce au machine learning, nous devons proposer aux salariés les plus susceptibles de quitter l'entreprise grâce un parcours d'accompagnement.



Notre Plan

1. Exploration et cleaning
2. Portrait robot de l'employé sur le départ
3. Modèle de machine learning
4. Evaluation du modèle et recommandation

1ere étape: Information sur la data

Nous disposons de 2 jeux de données :

- Train.csv : qui comprend 1100 lignes et 34 colonnes

- Test.csv : qui comprend 370 lignes et 34 colonnes

1ere étape: Le jeu de données

Les colonnes

Informations générales:

- Gender
- Age
- MaritalStatus
- Education
- EducationField
- NumCompaniesWorked
- TotalWorkingYears

Entreprise:

- DistanceFromHome
- StockOptionLevel
- TrainingTimesLastYear
- YearsAtCompany
- YearsSinceLastPromotion
- YearsWithCurrManager

Poste:

- BusinessTravel
- Department
- JobLevel
- OverTime
- WorkLifeBalance
- YearsInCurrentRole

Performance et satisfaction:

- JobInvolvement
- PerformanceRating
- EnvironmentSatisfaction
- RelationshipSatisfaction

Salaire:

- DailyRate
- HourlyRate
- MonthlyIncome
- PercentSalaryHike

1ere étape: Data Cleaning

Concernant la data elles sont plutôt propre.

Suppressions des colonnes EmployéeCount, Over18, StandardHours dans le train et le test et suppression d'index1 dans le test.

```
train_df.duplicated().sum()
test_df.duplicated().sum()
```

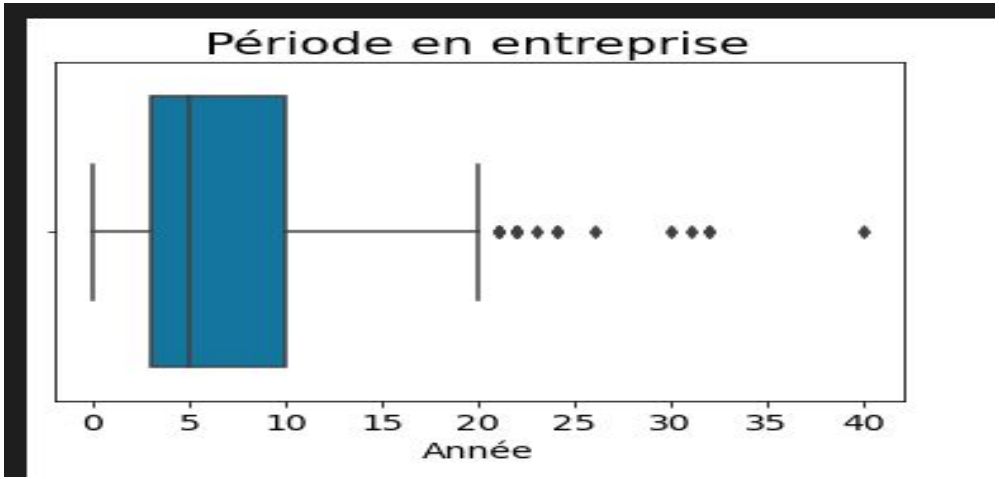
```
0
```

```
Valeurs manquantes par colonne:
Age                         0
Attrition                   0
BusinessTravel              0
DailyRate                   0
Department                  0
DistanceFromHome            0
Education                   0
EducationField              0
EmployeeCount               0
EnvironmentSatisfaction     0
Gender                      0
HourlyRate                  0
JobInvolvement              0
```

1ere étape: Donnée Aberrante

Concernant les valeurs aberrantes nous avons remarqué qu'elles étaient présente dans le test et le train.

On décide de les garder car la méthode de machine learning diminuera l'importance sur les outliers.



1ere étape: Encodage

Transformation des variables catégoriel “Attrition”et “Overtime”

c’est 3 colonnes nous les transformons en booléen (“PerformanceRating”)

Suite à cela on constate que dans la variable “Attrition” il y a 178 oui, soit environs 16% de personnes qui souhaite partir de l’entreprise.

1ere étape: Code cleaning

```
#Drop colonnes inutiles
train.drop(columns=["EmployeeCount", "Over18", "StandardHours"], inplace=True)
test.drop(columns=["index1", "EmployeeCount", "Over18", "StandardHours"], inplace=True)

#Scale et encondage
num_cols = [col for col in train if train[col].dtype == "int64"]
string_cols = [col for col in train if train[col].dtype == "object"]
must_be_categorical = ["Education", "WorkLifeBalance", "StockOptionLevel", "RelationshipSatisfaction", "PerformanceRating", "JobSatisfaction", "JobLevel", "JobInvolvement", "Environmen
must_be_scaled = list(set(num_cols) - set(must_be_categorical))
str_to_bool = {'Yes': 1, 'No': 0}
num_to_bool = {4: 1, 3: 0}

train[must_be_scaled] = StandardScaler().fit_transform(train[must_be_scaled])
train.Attrition = train.Attrition.map(str_to_bool)
train.OverTime = train.OverTime.map(str_to_bool)
train.PerformanceRating = train.PerformanceRating.map(num_to_bool)
test.OverTime = test.OverTime.map(str_to_bool)
```

1ere étape: En résumé

Bonne qualité de la donnée (pas de valeurs manquantes ni doublons...) mais quantité plutôt faible pour réaliser un modèle de Machine Learning.
Suppression de 4 colonnes inutiles: index1, EmployeeCount, Over18, StandardHours = 30 colonnes restantes pour prédire l'attrition
train.csv: Historiques des employés soit 1100 employés uniques
test.csv: Effectif présent: 370 employés uniques
L'attrition : + 178 - 922

1ere étape terminé passons maintenant au portrait robot de l’employé

2éme étape: Corrélations

On retrouve plusieurs corrélations très fortes, notamment entre le niveau hiérarchique et la rémunération (0.95)

JobLevel - MonthlyIncome 0.950980

JobLevel - TotalWorkingYears 0.782401

PercentSalaryHike - PerformanceRating 0.775132

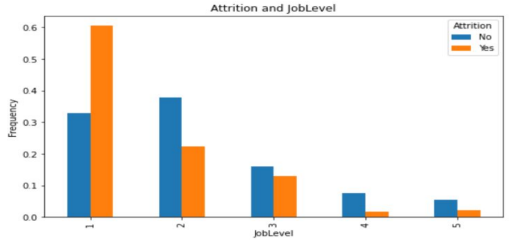
TotalWorkingYears - MonthlyIncome 0.768259

Les corrélations avec l'attrition sont assez faibles

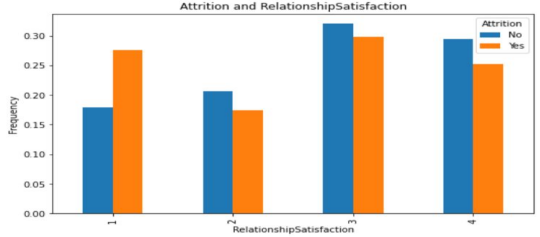
	corr with attrition
OverTime	0.265441
JobRole_Sales Representative	0.199775
TotalWorkingYears	-0.174393
JobLevel	-0.173708
MonthlyIncome	-0.167138
YearsInCurrentRole	-0.161772
YearsWithCurrManager	-0.159436
MaritalStatus_Single	0.158854
JobInvolvement	-0.154985
Age	-0.148381
YearsAtCompany	-0.148020
EnvironmentSatisfaction	-0.141538
BusinessTravel_Travel_Frequently	0.138366
StockOptionLevel	-0.133322
JobSatisfaction	-0.100901

2éme étape: Portrait robot

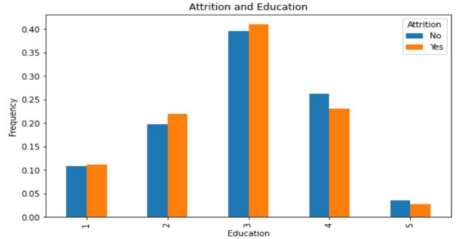
JobLevel



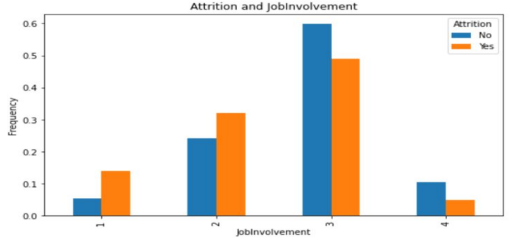
RelationshipSatisfaction



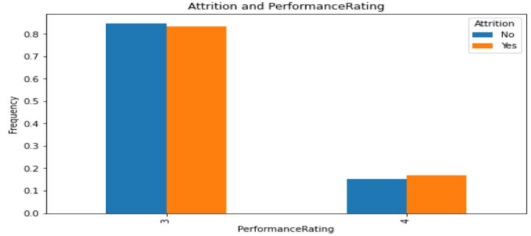
Education



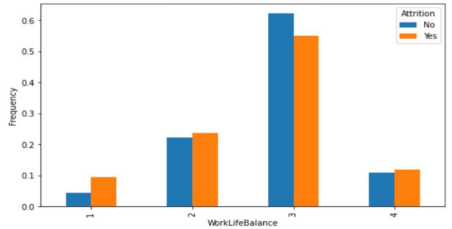
JobInvolvement



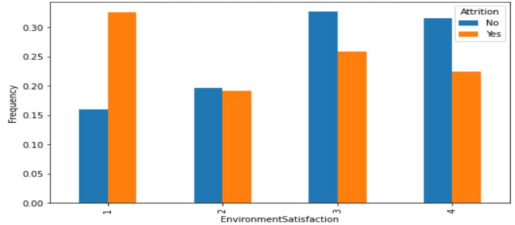
PerformanceRating



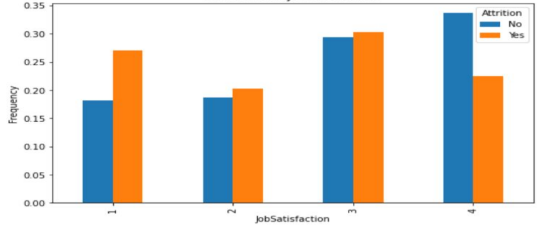
WorkLifeBalance



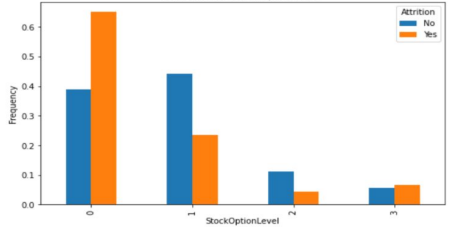
EnvironmentSatisfaction



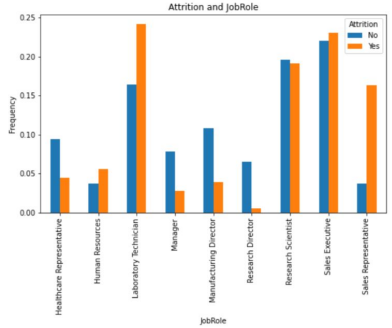
JobSatisfaction



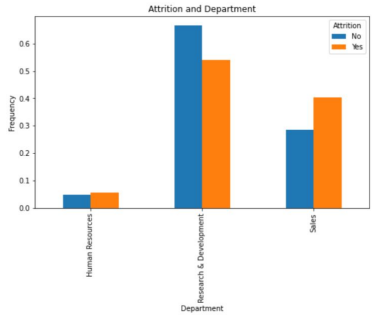
StockOptionLevel



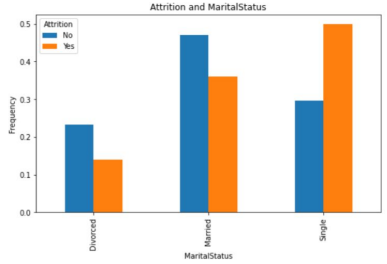
JobRole



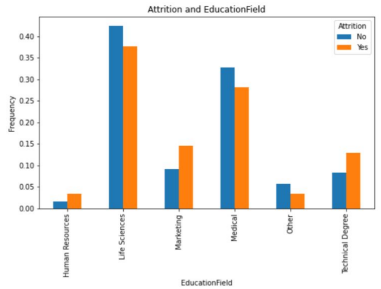
Department



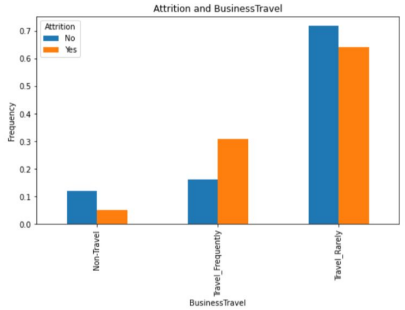
MaritalStatus



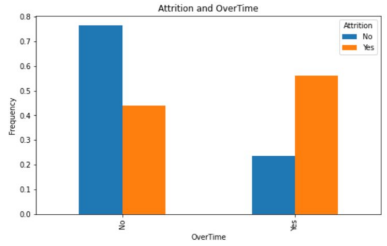
EducationField



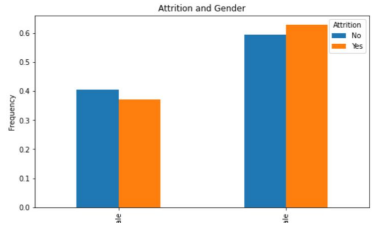
BusinessTravel



OverTime



Gender



Le fait de normaliser les colonnes nous permet d'observer les catégories ou l'attrition est surepresentée ou sous représenté à savoir:

Attrition -

- JobRole_Research Director
- JobRole_Manager
- JobRole_Manufacturing Director
- BusinessTravel_Non-Travel
- JobRole_Healthcare Representative
- EducationField_Other
- MaritalStatus_Divorced
- StockOptionLevel
- YearsInCurrentRole
- YearsWithCurrManager
- YearsAtCompany
- TotalWorkingYears
- MonthlyIncome

Attrition +

- JobRole_Sales Representative
- WorkLifeBalance
- OverTime
- EducationField_Human Resources (seulement 21)
- BusinessTravel_Travel_Frequently
- MaritalStatus_Single
- EducationField_Marketing
- EducationField_Technical Degree
- JobRole_Human Resources
- JobRole_Laboratory Technician

3éme étape:Machine Learning

Il s’appuie sur la régression logistique.
Etape de réalisation:

. Encodage des colonnes catégorielles (dummies) et normalisation des données numériques (RobustScaler) pour être sur même échelle et diminuer le poids des outliers (on va éviter de supprimer des outliers vu le peu de données).

. Séparation de notre dataset en 2 jeux de données train/test (0.3)

. Pour le train, simulation d’employés attrition+ avec SMOTE (770 -> 1288)

. Sélection des variables ayant le plus grand pouvoir explicatif (RFE -> 25 features)

. Pour ces variables vérification du p-value < 0.05, pour vérifier que les variations de ces variables ne sont pas du au hasard et que l’effectif associé est suffisant pour avoir une valeur statistique. (-> 16 features)

. GridSearchCV pour trouver les meilleurs paramètres : on désactive la régularisation

4éme étape: Evaluation Machine Learning

[145]:
```
X_train, X_valid, y_train, y_valid = train_test_split(X_train_plus_best, y_train_plus, test_size=0.3, random_state=0)
log_reg = LogisticRegression(penalty="none")
log_reg.fit(X_train, y_train)
y_pred = log_reg.predict(X_valid)
accuracy = accuracy_score(y_valid, y_pred)
f1 = f1_score(y_valid, y_pred)
tn, fp, fn, tp = confusion_matrix(y_valid, y_pred).ravel()
conf_ma = [[tp, tn], [fp,fn]]
df_conf_ma = pd.DataFrame(conf_ma, index=["Sucess", "Error"], columns=["Positive", "Negative"])
display(df_conf_ma)
```

	Positive	Negative
Sucess	164	166
Error	21	36

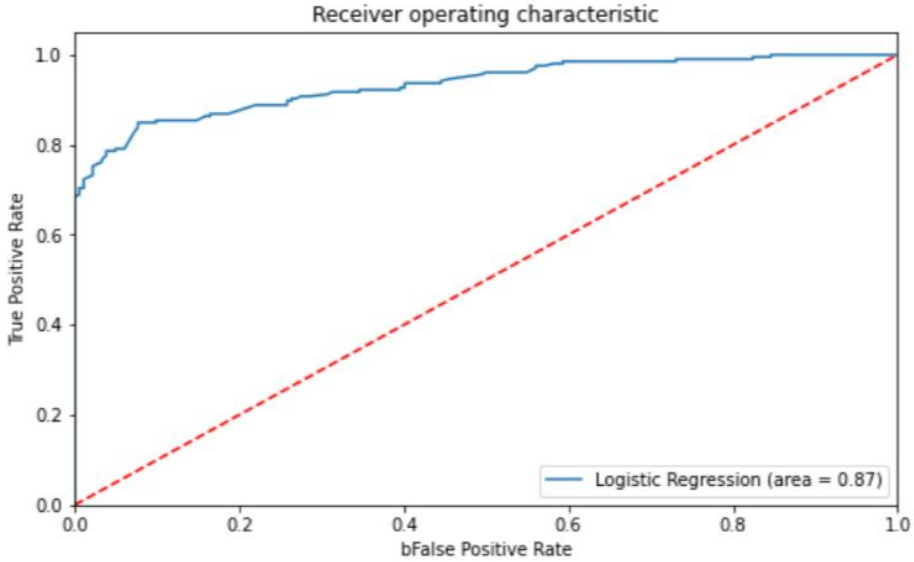
[146]:
```
from sklearn.metrics import classification_report
print(classification_report(y_valid, y_pred))
```

```
              precision    recall  f1-score   support

           0       0.82      0.89      0.85       187
           1       0.89      0.82      0.85       200

    accuracy                           0.85       387
   macro avg       0.85      0.85      0.85       387
weighted avg       0.86      0.85      0.85       387
```

Courbe ROC


Receiver operating characteristic
True Positive Rate
bFalse Positive Rate
Logistic Regression (area = 0.87)
0.0
0.2
0.4
0.6
0.8
1.0

Amélioration du modèle

En supprimant la sélection des features (RFE+p-features) on peut augmenter le f-score tout en ayant une meilleure matrice de confusion.
Cependant cela entraîne un risque d'overfitting que l'on peut difficilement quantifier car il nous faudrait davantage de données d'apprentissage. On conservera donc le modèle moins performant mais certainement plus discriminant sur des nouvelles données.

[224]:
```
X_train, X_valid, y_train, y_valid = train_test_split(X_train_plus_best, y_train_plus, test
log_reg = LogisticRegression(penalty="none")
log_reg.fit(X_train, y_train)
y_pred = log_reg.predict(X_valid)
accuracy = accuracy_score(y_valid, y_pred)
f1 = f1_score(y_valid, y_pred)
tn, fp, fn, tp = confusion_matrix(y_valid, y_pred).ravel()
conf_ma = [[tp, tn], [fp,fn]]
df_conf_ma = pd.DataFrame(conf_ma, index=["Sucess", "Error"], columns=["Positive", "Negativ
display(df_conf_ma)
```

	Positive	Negative
Sucess	178	177
Error	10	22

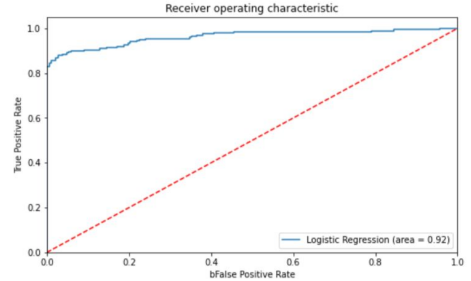
[225]:
```
from sklearn.metrics import classification_report
print(classification_report(y_valid, y_pred))
```

```
              precision    recall  f1-score   support

           0       0.89      0.95      0.92       187
           1       0.95      0.89      0.92       200

    accuracy                           0.92       387
   macro avg       0.92      0.92      0.92       387
weighted avg       0.92      0.92      0.92       387
```

[226]:
```
from sklearn.metrics import roc_auc_score
from sklearn.metrics import roc_curve
logit_roc_auc = roc_auc_score(y_valid, log_reg.predict(X_valid))
fpr, tpr, thresholds = roc_curve(y_valid, log_reg.predict_proba(X_valid)[:,1])
plt.figure()
plt.plot(fpr, tpr, label='Logistic Regression (area = %0.2f)' % logit_roc_auc)
plt.plot([0, 1], [0, 1],'r--')
plt.xlim([0.0, 1.0])
plt.ylim([0.0, 1.05])
plt.xlabel('bFalse Positive Rate')
plt.ylabel('True Positive Rate')
plt.title('Receiver operating characteristic')
plt.legend(loc="lower right")
plt.savefig('Log_ROC')
plt.show()
```



Prédictions et recommandations

51 salariés ont été sélectionnés (>0.5).

Cela ne veut pas dire que le groupe doit accueillir 51 personnes.

Nous recommandons la construction d'un indicateur permettant de connaître l'investissement optimal par salarié. Par exemple en prenant en compte les performances, le poste, l'ancienneté… car au delà de savoir si le salarié souhaite partir il faudrait aussi savoir à quel point son départ peut être ou non néfaste à l'entreprise.

Par ailleurs nous recommandons une action ciblée auprès du service vente qui est fortement touché par le turn-over.

	no	yes
305	0.009	0.991
341	0.014	0.986
267	0.078	0.922
221	0.078	0.922
315	0.121	0.879
170	0.137	0.863
329	0.161	0.839
97	0.166	0.834
115	0.175	0.825
23	0.195	0.805
306	0.214	0.786
80	0.229	0.771
25	0.233	0.767
82	0.280	0.720
152	0.292	0.708
111	0.292	0.708
351	0.301	0.699
79	0.308	0.692
95	0.312	0.688
17	0.323	0.677
178	0.328	0.672
76	0.328	0.672
0	0.330	0.670
265	0.330	0.670
353	0.330	0.670
184	0.330	0.670
41	0.353	0.647
300	0.353	0.647
117	0.375	0.625
247	0.399	0.601
60	0.399	0.601
10	0.421	0.579
36	0.429	0.571
334	0.431	0.569
179	0.434	0.566
141	0.454	0.546
54	0.454	0.546
34	0.459	0.541
174	0.459	0.541
216	0.459	0.541
266	0.459	0.541
271	0.462	0.538
128	0.471	0.529
343	0.471	0.529
263	0.482	0.518
244	0.488	0.512