HULT
INTERNATIONAL
BUSINESS SCHOOL

Data Mining with Mysql, Nosql, Hadoop, Spark, and Hive

IBM Team Assignment

Team2: HongSeop, Lynda, Roger, Yashasv, Megumu

HULT INTERNATIONAL BUSINESS SCHOOL

DATA AND OBJECTIVE

Dataset

- IBM HR Attrition Dataset
- The dataset has 35 variable columns and 1,470 rows

Age	Attrition	BusinessTravel	DailyRate	Department	DistanceFromHome	Education	EducationField	EmployeeCount	EmployeeNumber	EnvironmentSatisfaction	Gender	HourlyRate	JobInvolvement	JobLevel
bigint	string	string	bigint	string	tinyint	tinyint	string	tinyint	bigint	tinyint	string	int	tinyint	tinyint
Integer	Boolean	Text	Integer	Text	Integer	Integer	Text	Integer	Integer	Integer	Gender	Integer	Integer	Integer
41	Yes	Travel_Rarely	1102	Sales	1	2	Life Sciences	1	1	2	Female	94	3	2
49	No	Travel_Frequently	279	Research & Development	8	1	Life Sciences	1	2	3	Male	61	2	2
37	Yes	Travel_Rarely	1373	Research & Development	2	2	Other	1	4	4	Male	92	2	1
33	No	Travel_Frequently	1392	Research & Development	3	4	Life Sciences	1	5	4	Female	56	3	1
27	No	Travel_Rarely	591	Research & Development	2	1	Medical	1	7	1	Male	40	3	1
32	No	Travel_Frequently	1005	Research & Development	2	2	Life Sciences	1	8	4	Male	79	3	1
59	No	Travel_Rarely	1324	Research & Development	3	3	Medical	1	10	3	Female	81	4	1
30	No	Travel_Rarely	1358	Research & Development	24	1	Life Sciences	1	11	4	Male	67	3	1
38	No	Travel_Frequently	216	Research & Development	23	3	Life Sciences	1	12	4	Male	44	2	3
36	No	Travel_Rarely	1299	Research & Development	27	3	Medical	1	13	3	Male	94	3	2
35	No	Travel_Rarely	809	Research & Development	16	3	Medical	1	14	1	Male	84	4	1
29	No	Travel_Rarely	153	Research & Development	15	2	Life Sciences	1	15	4	Female	49	2	2
31	No	Travel_Rarely	670	Research & Development	26	1	Life Sciences	1	16	1	Male	31	3	1

Objective

How the company decrease leaving employees

How to analyze

Finding features(columns) related to Attrition: “Yes”

HULT INTERNATIONAL BUSINESS SCHOOL

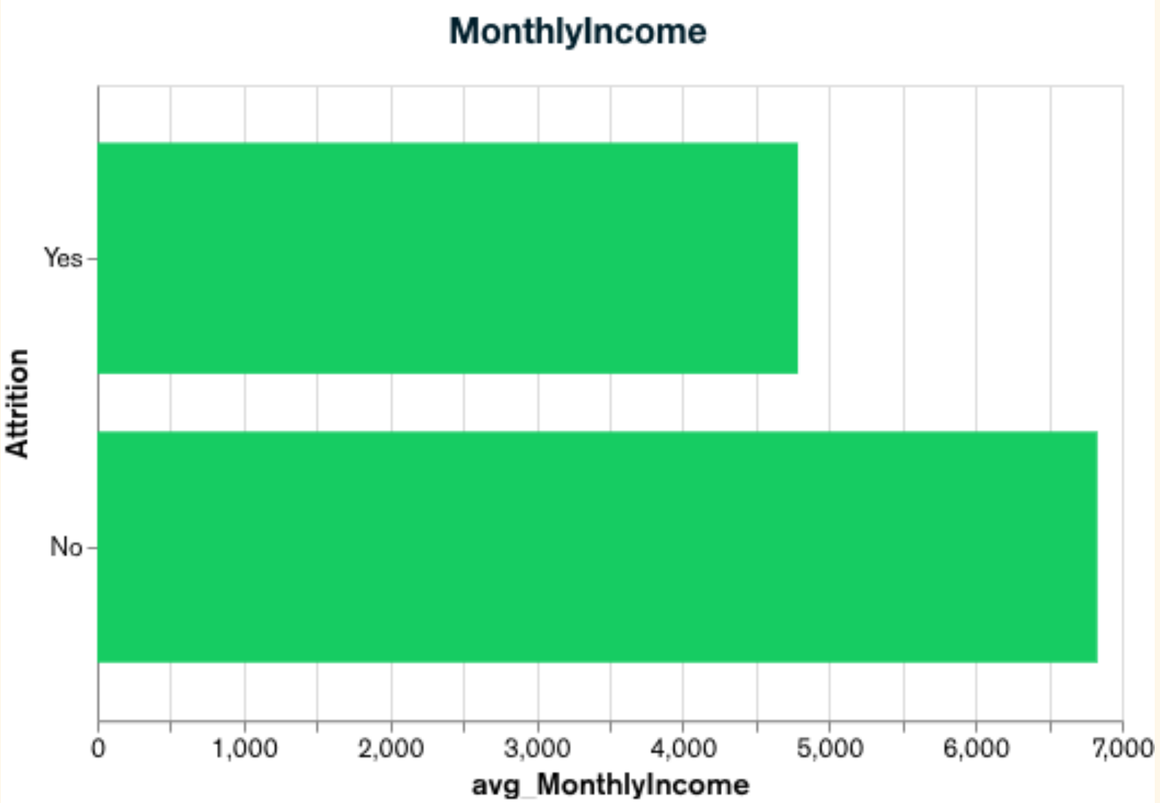
DATA ANALYZING FLOW

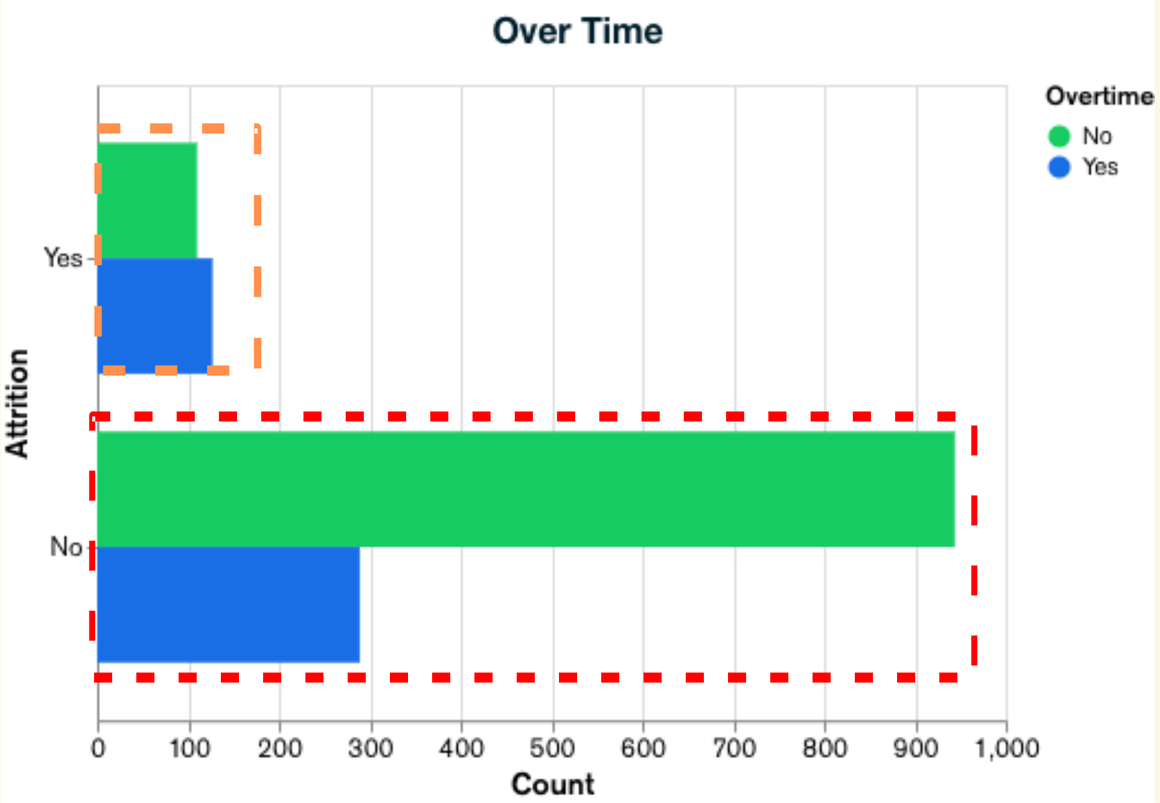
	Data Visualizing	AI Analysis	Business Insight Recommendation
Contents	• Visualizing the data • Finding the factor related to Attrition on the visualized charts	• Making the AI model • Checking the coefficients to confirm the factors related to Attrition or not	• Gaining the business insights based on our visualizing and AI analysis • Recommendations to IBM
DB	Mongo DB	Spark	-
Environment	Mongo DB Atlas Dashboard	PySpark on Dataiku	-

HULT
INTERNATIONAL
BUSINESS SCHOOL

DATA VISUALIZING IN MONGODB

We visualized almost of all data. In these data, average of **“MonthlyIncome”** and **“OverTime”** has a big difference between Attrition YES and No.





PREDICTIVE MODELING IN DFS - PYSPARK

AI prediction results

features	label	rawPrediction	probability	prediction
(5,[0,3],[1.0,246...]	0	[1.95241127441409...]	[0.87570932879953...]	0.0
(5,[0,3],[1.0,254...]	0	[1.96248345523969...]	[0.87680146802390...]	0.0
(5,[0,3],[1.0,261...]	1	[1.97123034911454...]	[0.87774320340243...]	0.0
(5,[0,3],[1.0,261...]	0	[1.97162793519976...]	[0.87778586198671...]	0.0
(5,[0,3],[1.0,268...]	0	[1.98103747254999...]	[0.87879171383867...]	0.0
(5,[0,3],[1.0,294...]	0	[2.01522987587897...]	[0.88238686655610...]	0.0
(5,[0,3],[1.0,455...]	0	[2.22913118972775...]	[0.90283517032261...]	0.0
(5,[0,3],[1.0,472...]	0	[2.25139601050011...]	[0.90477088409279...]	0.0
(5,[0,3],[1.0,506...]	0	[2.29632323813006...]	[0.90857207303412...]	0.0
(5,[0,3],[1.0,513...]	0	[2.30533518939506...]	[0.90931793422989...]	0.0
(5,[0,3],[1.0,525...]	0	[2.32203380497434...]	[0.91068550334029...]	0.0
(5,[0,3],[1.0,560...]	0	[2.36815379085995...]	[0.91436641205485...]	0.0
(5,[0,3],[1.0,596...]	0	[2.41626170717166...]	[0.91805896423421...]	0.0
(5,[0,3],[1.0,639...]	0	[2.47311651735823...]	[0.92223556685983...]	0.0
(5,[0,3],[1.0,894...]	0	[2.81053457501559...]	[0.94324244505347...]	0.0
(5,[0,3],[1.0,121...]	1	[3.23807214532299...]	[0.96224212864583...]	0.0
(5,[0,3],[1.0,138...]	0	[3.45767219305993...]	[0.96945911998152...]	0.0
(5,[0,3],[1.0,152...]	0	[3.6400316774812,...]	[0.97442000128206...]	0.0
(5,[0,3],[1.0,178...]	0	[3.99176283420650...]	[0.98186772026493...]	0.0
(5,[0,3],[1.0,184...]	0	[4.06783430517875...]	[0.98317356161897...]	0.0

only showing top 20 rows

AUC ROC:0.8008662508662475

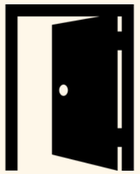
Coefficients

No	Features	Feature Weight	Note
1	BTFreq	0.473	"BusinessTravel" Travel-Frequently: 1 Travel-Rarely: 0 No-Travel: 0
2	MSSingle	0.899	"MaritalStatus" Single:1 Discovered:0 Married:0
3	OverTime	**1.190**	row data
4	MonthlyIncome	**-1.115**	row data
5	Department	0.561	"Department" Sales:1 HR:1 Researcher:0

Model intercept: -2.124

PREDICTIVE MODELING INSIGHTS

AUC ROC = 0.80 -> Predicts Y variables with high confidence



Single, Overtime -> Likely to leave

Monthly Income -> Likely to stay

Overall, it aligns with the previous MongoDB analysis

(Bock, 2022)

HULT
INTERNATIONAL
BUSINESS SCHOOL

CONCLUSIONS

- Overtime work has a significant direct relationship with Attrition: the more overtime employees work, the more likely they leave IBM.

- Salary increases and bonuses are not features that significantly impact Attrition, although there is always room for improvement, this can slightly reduce the chances of leaving.

- Mini vacations in between trips and organizing in-house social networking events reduce the odds of Attrition=yes.

- Offer career development opportunities has a relationship with Attrition and reduces the odds that employees -especially singles- leave IBM.

HULT
INTERNATIONAL
BUSINESS SCHOOL

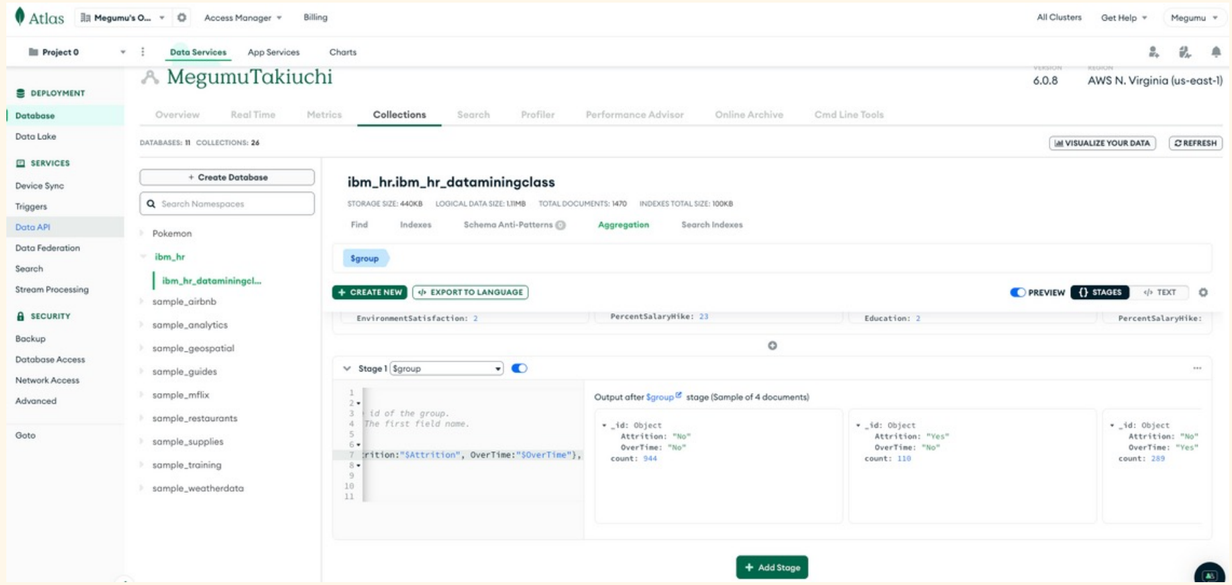
REFERENCES

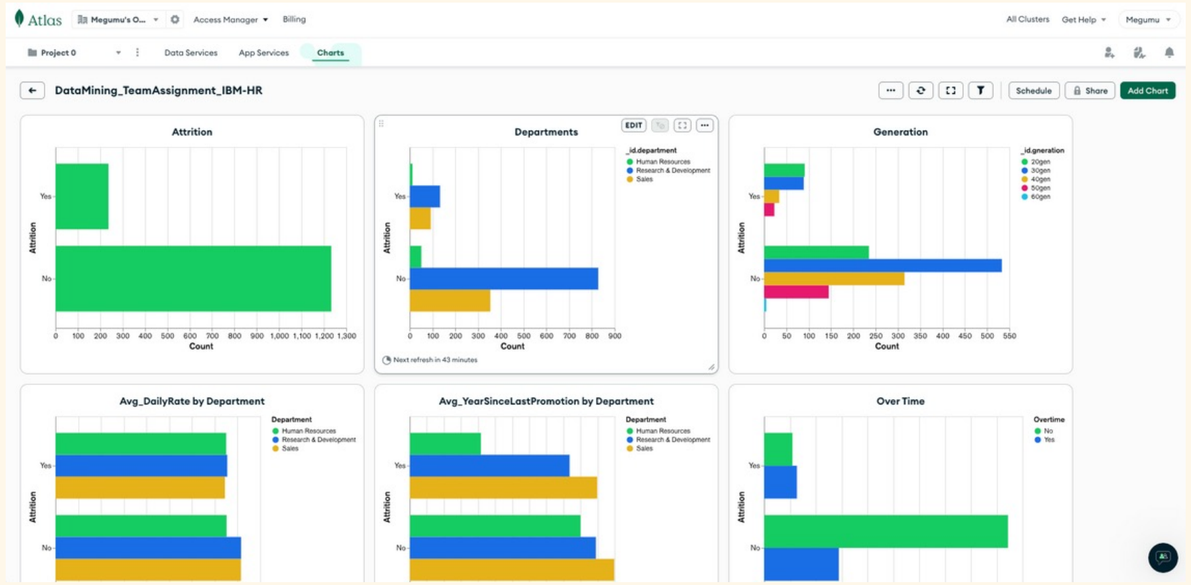
Bock, L. (2022, July). It's Time to Reimagine Employee Retention. Harvard Business Review. Retrieved from https://hbr.org/2022/07/its-time-to-reimagine-employee-retention

HULT INTERNATIONAL BUSINESS SCHOOL

APPENDIX

- We used MongoDB and Spark to practice visualizing and creating AI models.
- Visualizing environment: Mongo DB Atlas





HULT INTERNATIONAL BUSINESS SCHOOL

PySpark Code (1)

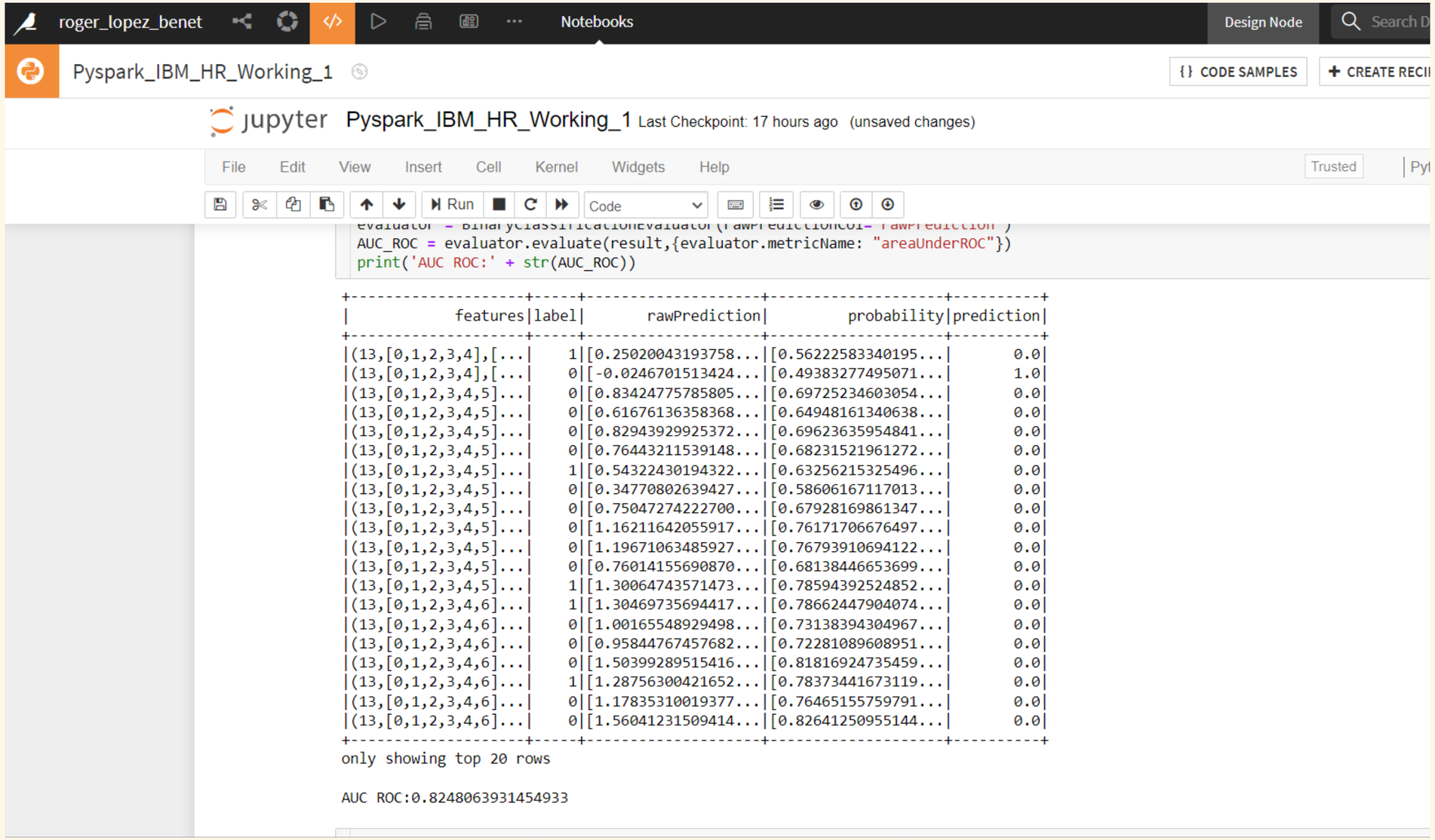
```
from pyspark.ml import Pipeline
from pyspark.ml.classification import LogisticRegression
from pyspark.ml.feature import HashingTF, Tokenizer
from pyspark.sql import Row
from pyspark.sql.functions import UserDefinedFunction
from pyspark.sql.types import *
from pyspark.ml.feature import VectorAssembler
from pyspark.mllib.evaluation import *

#creating vectors with names of variables
vecAssembler = VectorAssembler(inputCols = ['BTFreq', 'MSSingle', 'OverTime', 'MonthlyIncome', 'HRSales'], output
v_df = vecAssembler.transform(df)
vhouse_df = v_df.select(['features', 'Attrition'])
vhouse_df = vhouse_df.withColumnRenamed("Attrition", "label")# We have to rename our output variable to 'label'

#splitting the dataset
splits = vhouse_df.randomSplit([0.7, 0.3])
train_df = splits[0]
test_df = splits[1]

#creating an object with the logistic regression engine
lr = LogisticRegression(maxIter=20)
pipeline = Pipeline(stages=[lr])

#fitting the model
model = lr.fit(train_df)

#evaluating the model using testing data
result = model.transform(test_df)
result.prediction
result.show()

from pyspark.ml.evaluation import BinaryClassificationEvaluator
evaluator = BinaryClassificationEvaluator(rawPredictionCol="rawPrediction")
AUC_ROC = evaluator.evaluate(result,{evaluator.metricName: "areaUnderROC"})
print('AUC ROC:' + str(AUC_ROC))
```

HULT INTERNATIONAL BUSINESS SCHOOL

PySpark Code (2)

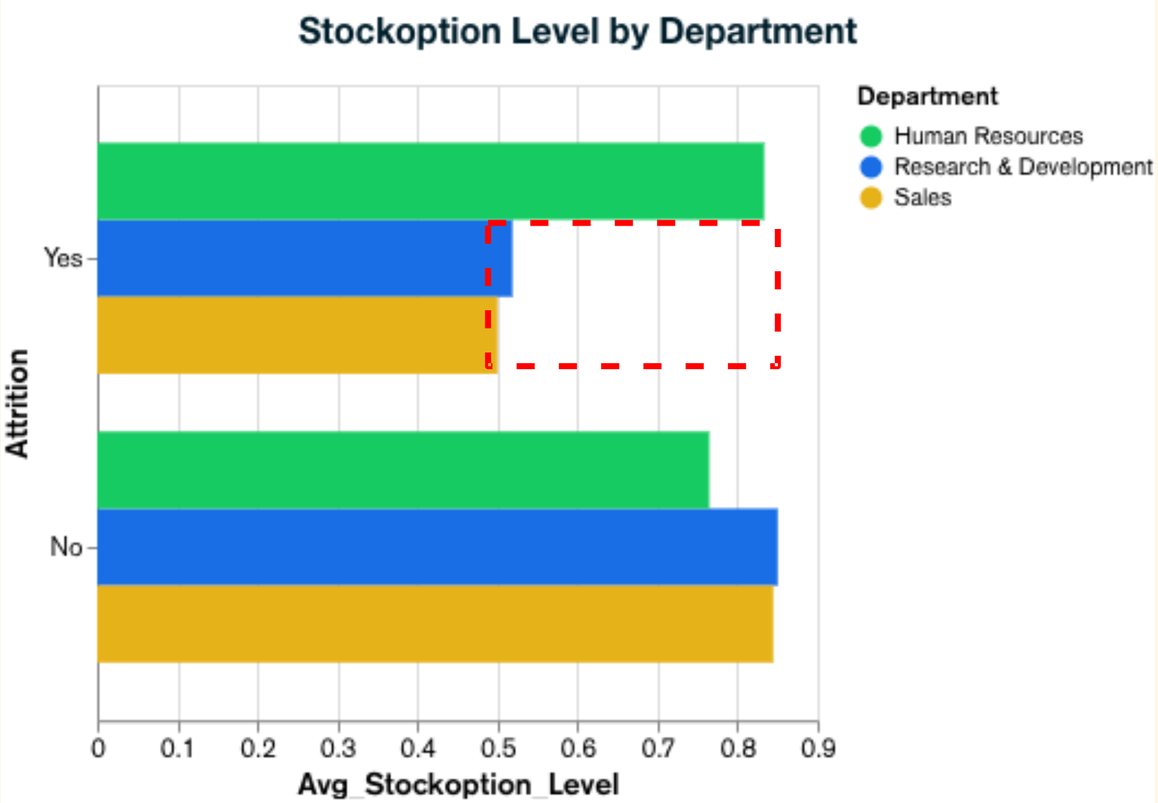
```
from pyspark.ml.evaluation import BinaryClassificationEvaluator

# Evaluate model
evaluator = BinaryClassificationEvaluator(rawPredictionCol="rawPrediction")
evaluator.evaluate(result)

from pyspark.ml.tuning import ParamGridBuilder, CrossValidator

# Create ParamGrid for Cross Validation
paramGrid = (ParamGridBuilder()
             .addGrid(lr.regParam, [0.01, 0.5, 2.0])
             .addGrid(lr.elasticNetParam, [0.0, 0.5, 1.0])
             .addGrid(lr.maxIter, [1, 5, 10])
             .build())
# Create 5-fold CrossValidator
cv = CrossValidator(estimator=lr, estimatorParamMaps=paramGrid, evaluator=evaluator, numFolds=5)

# Run cross validations
cvModel = cv.fit(train_df)
# this will likely take a fair amount of time because of the amount of models that we're creating and testing

# Use test set to measure the accuracy of our model on new data
predictions = cvModel.transform(test_df)

# cvModel uses the best model found from the Cross Validation
# Evaluate best model
evaluator.evaluate(predictions)

print('Model Intercept: ', cvModel.bestModel.intercept)
weights = cvModel.bestModel.coefficients
weights = [(float(w),) for w in weights]  # convert numpy type to float, and to tuple
weightsDF = sqlContext.createDataFrame(weights, ["Feature Weight"])
weightsDF.show()
# View best model's predictions and probabilities of each prediction class
selected = predictions.select("label", "prediction", "probability", "features")
selected.show()
```

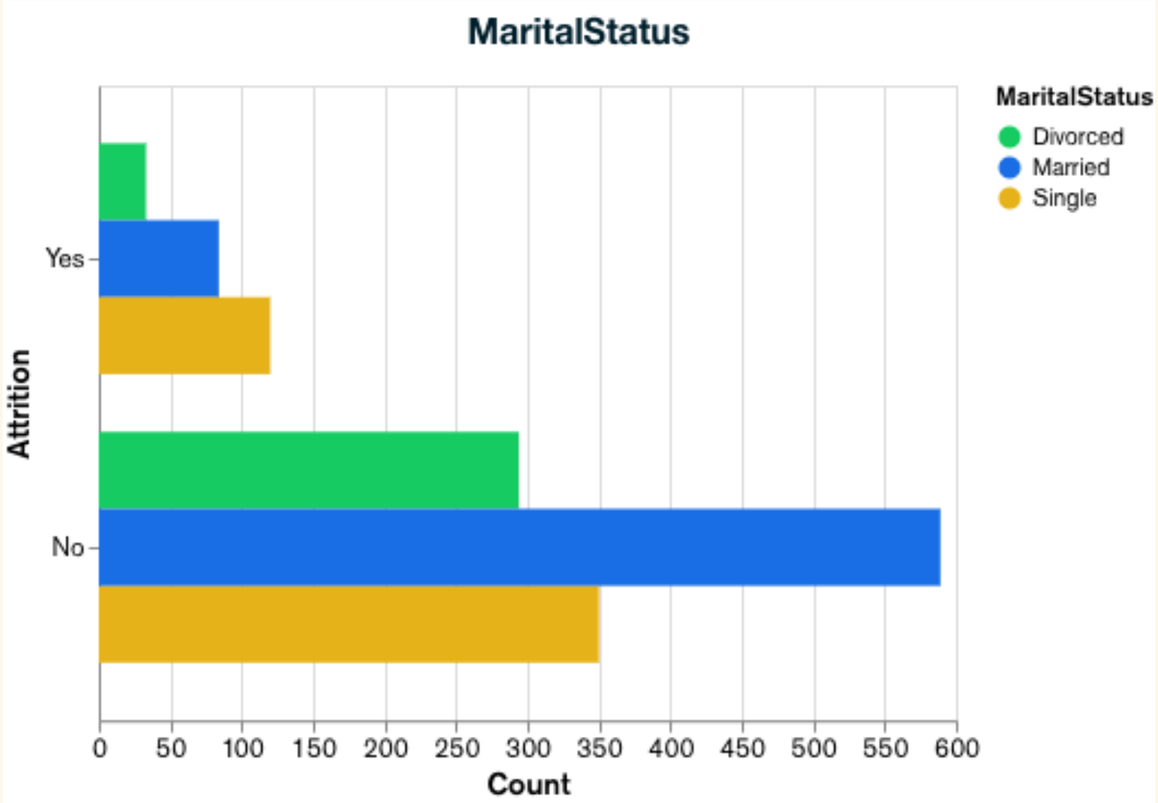
HULT
INTERNATIONAL
BUSINESS SCHOOL



HULT INTERNATIONAL BUSINESS SCHOOL

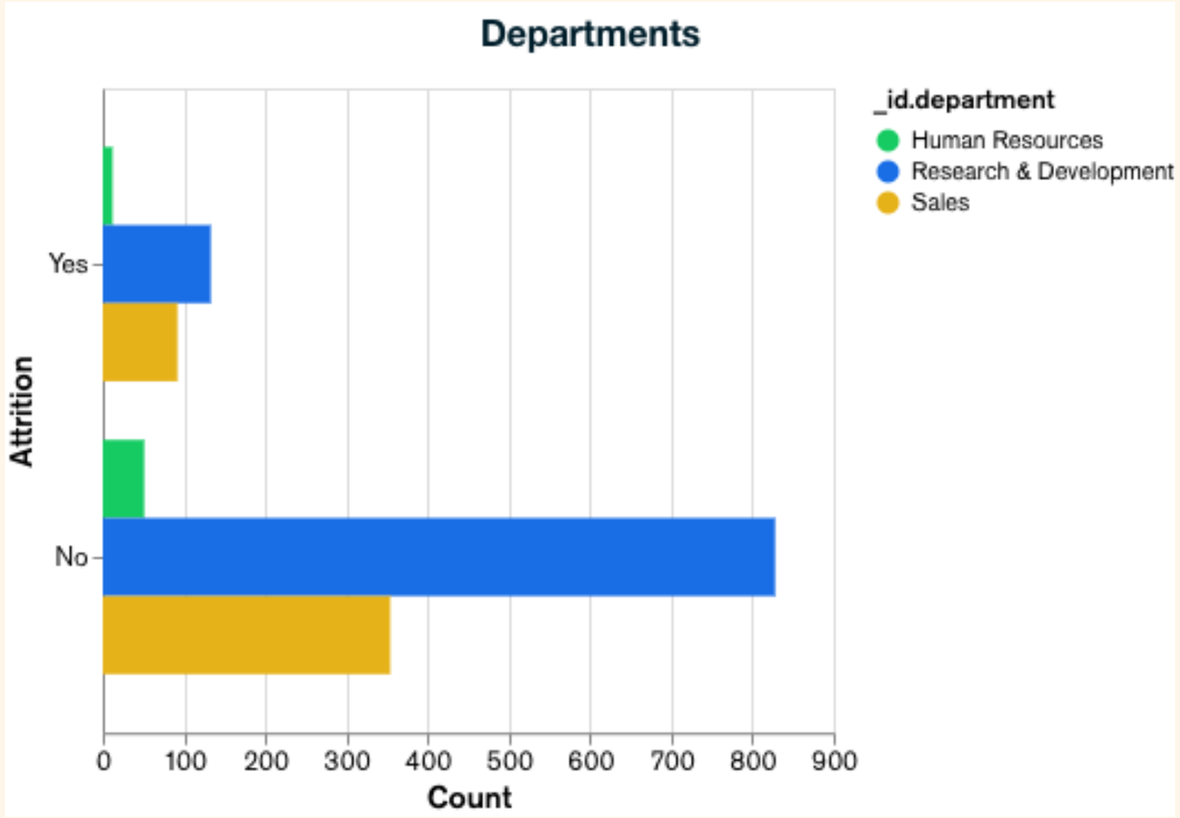
Data Visualizing in MongoDB

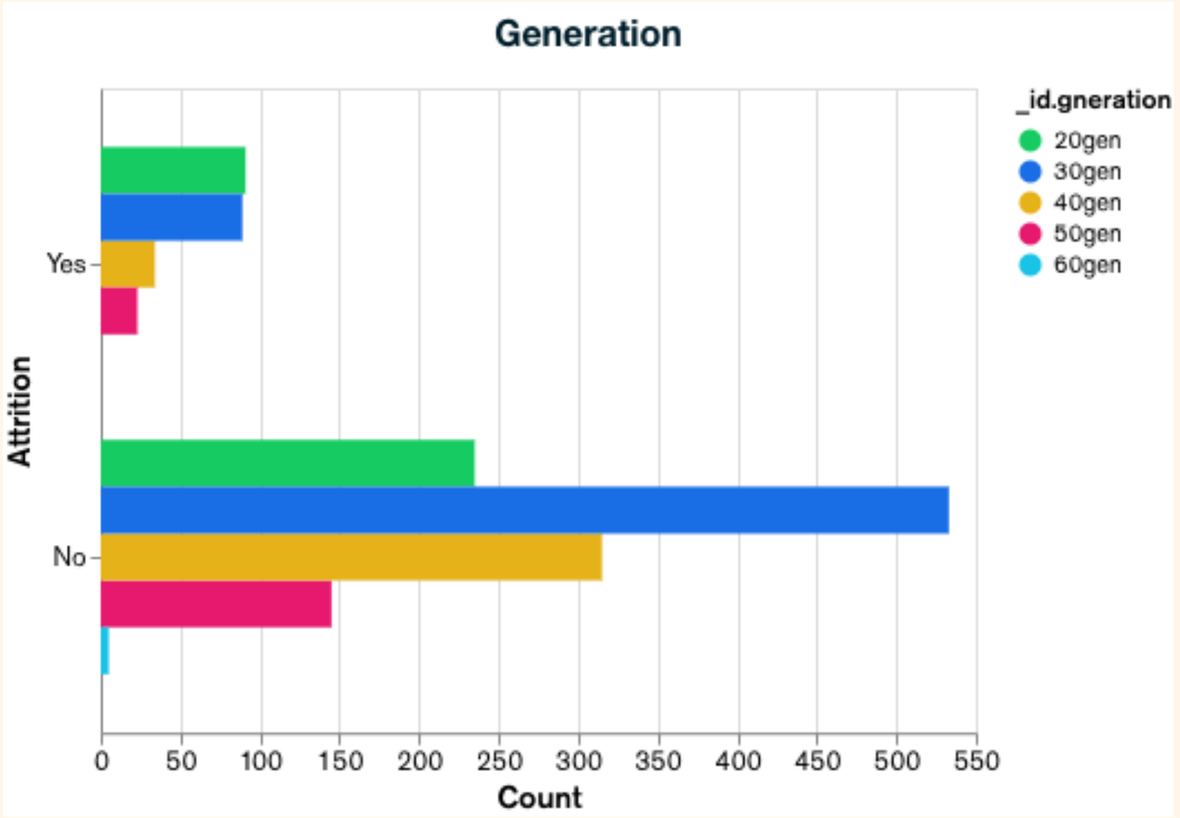




HULT INTERNATIONAL BUSINESS SCHOOL

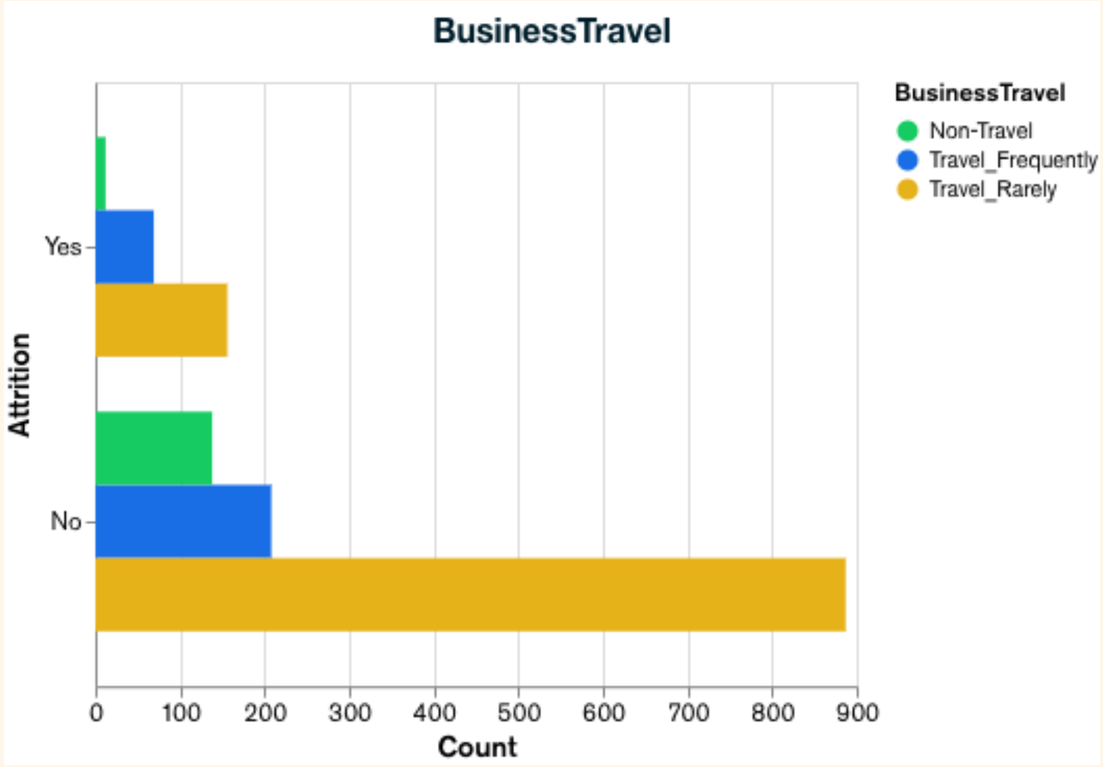
Data Visualizing in MongoDB

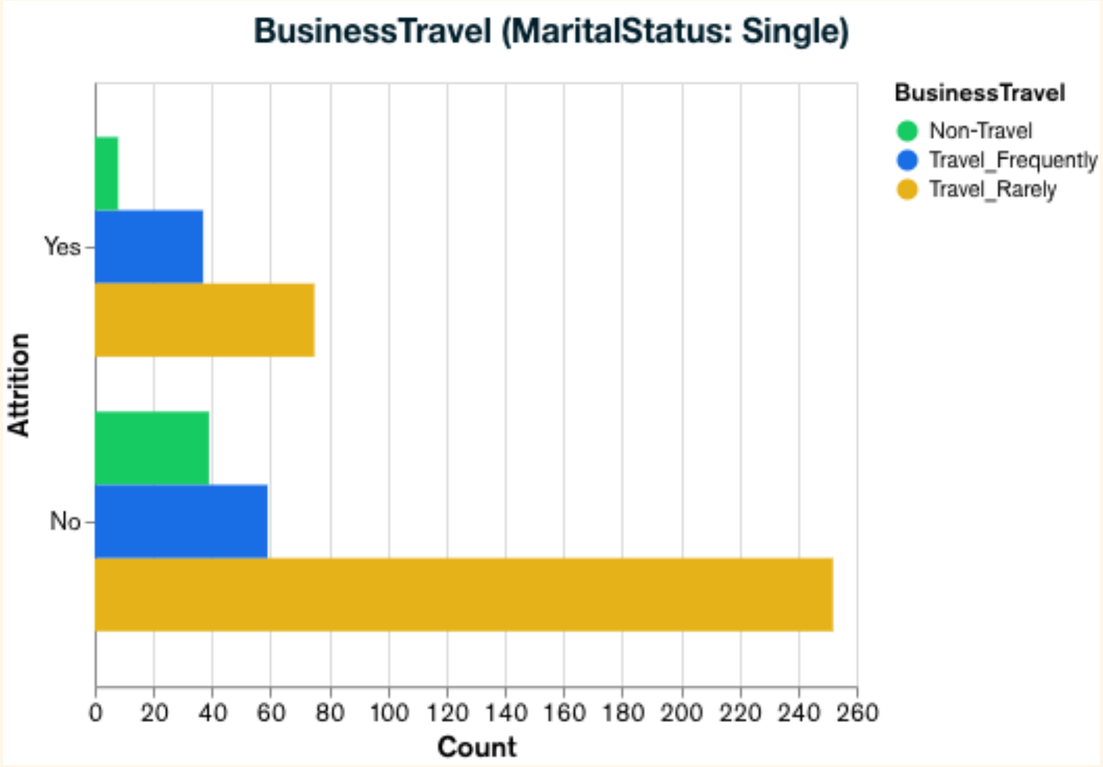




HULT INTERNATIONAL BUSINESS SCHOOL

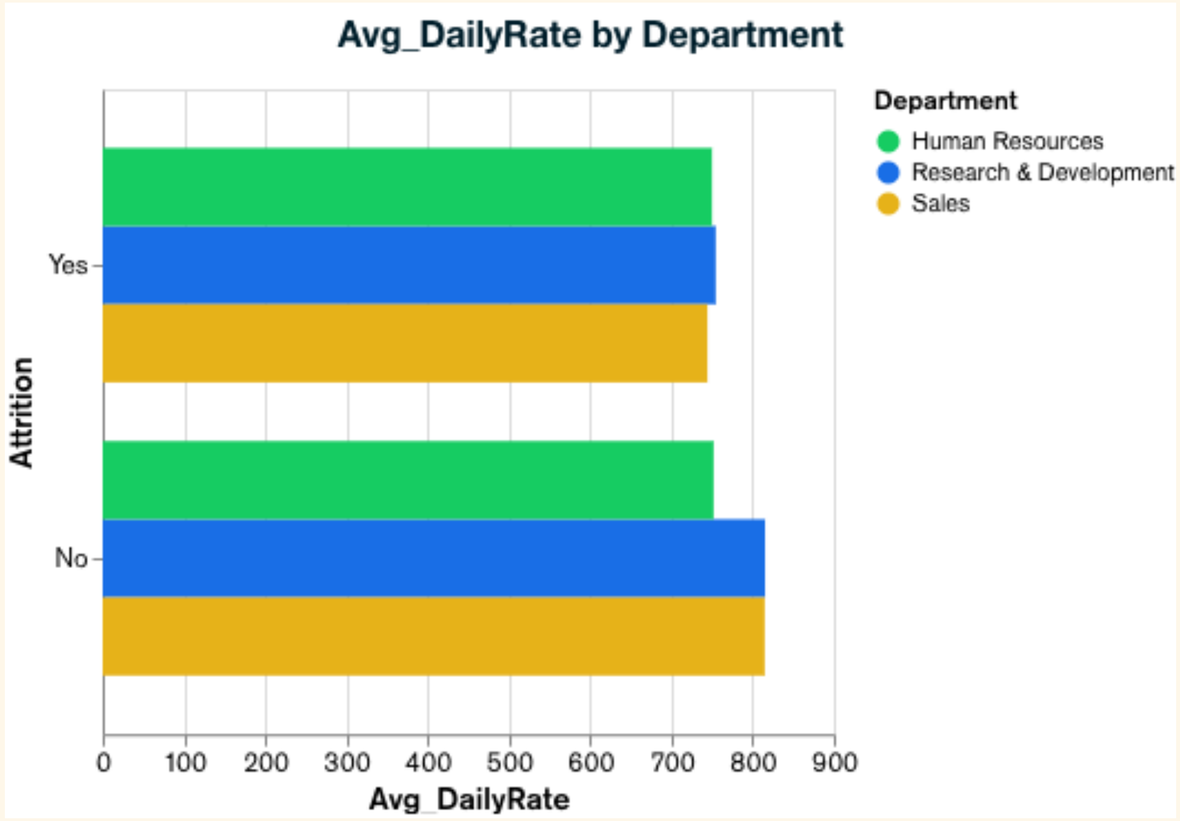
Data Visualizing in MongoDB

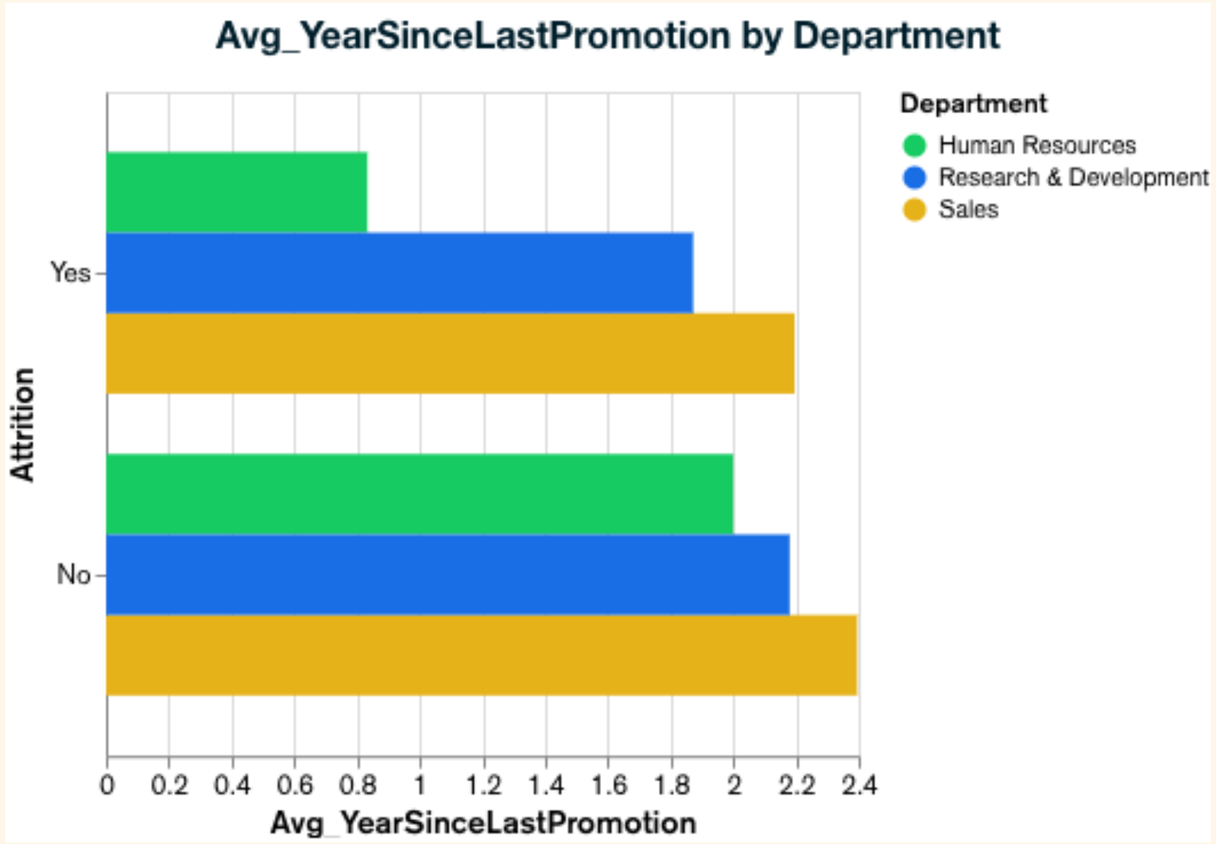




HULT INTERNATIONAL BUSINESS SCHOOL

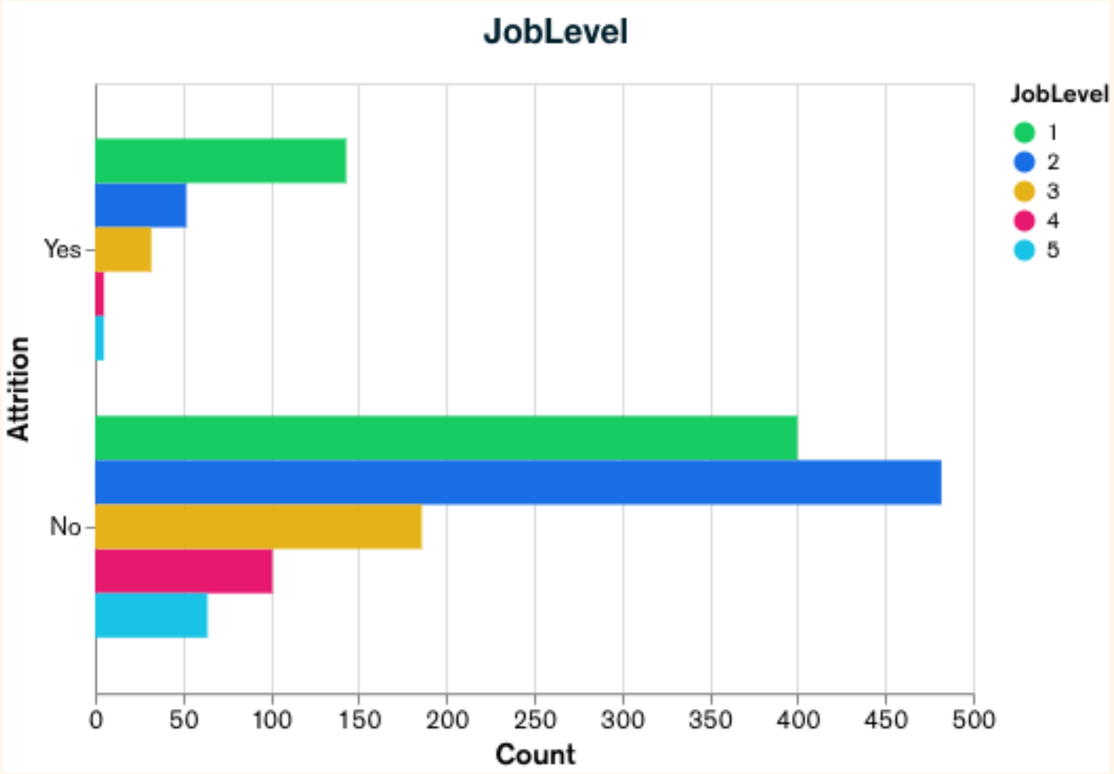
Data Visualizing in MongoDB

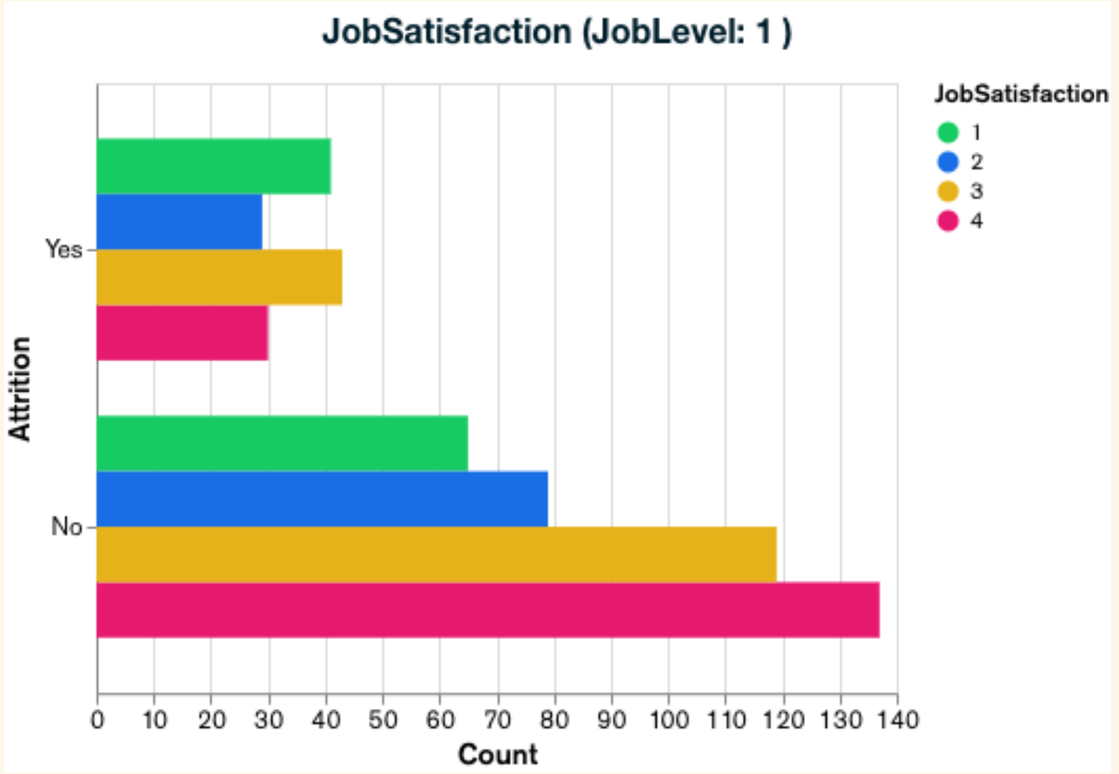




HULT INTERNATIONAL BUSINESS SCHOOL

Data Visualizing in MongoDB





HULT INTERNATIONAL BUSINESS SCHOOL

Data Visualizing in MongoDB