



Machine Learning per l'analisi del Turnover





Speaker

Andrea Bergonzi

Data Scientist @ Dataskills srl



andrea.bergonzi@dataskills.it

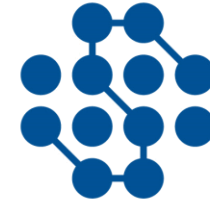


<https://www.linkedin.com/in/andrea-bergonzi-5a1390103/>

Sponsor & Org



UNIVERSITÀ DEGLI STUDI DI PARMA



DATA SKILLS
UNDERSTANDING THE WORLD



Dataskills

Specializzati nella creazione di soluzioni innovative nelle quattro aree principali della Data Science.

BUSINESS INTELLIGENCE

- Trasformare dati e informazioni in conoscenza

PREDICTIVE ANALYTICS

- Utilizzare i dati per offrire previsioni sul futuro

BIG DATA

- Gestire, immagazzinare e analizzare immense moli di dati

IOT ANALYTICS

- Estrarre e sfruttare i dati provenienti da device interconnessi

25

Anni di
esperienza

90+

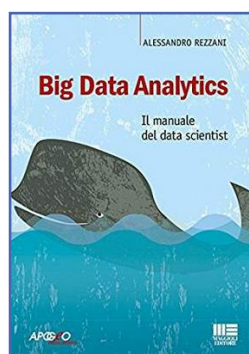
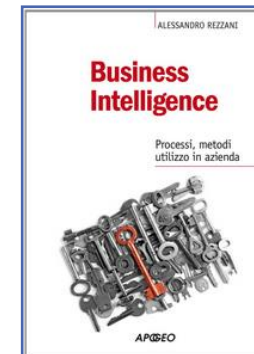
Progetti
realizzati

3

Libri di Data
Science
pubblicati

2

Professori
all'Università
Bocconi

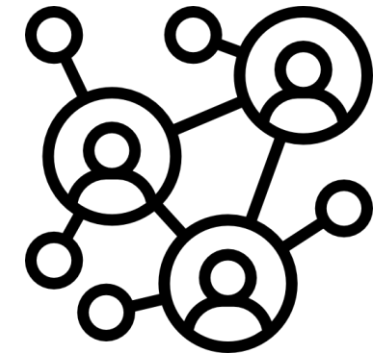
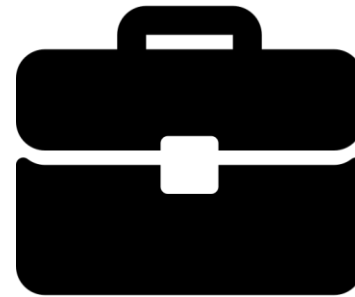


Agenda

1. Business Context
2. Metodologia adottata: **XGBOOST** & **SHAP**
3. Data preparation & Feature Engineering
4. Model Pipeline Implementation
5. Lettura e interpretazione dei risultati
6. Business outcome & conclusioni

Business Context

- Azienda organizzata in filiali su tutto il territorio italiano
- Modelli di business ibridi
- Rete di collaboratori
- Alti tassi di turnover



Business Requirements

Necessità di diagnosticare il Turnover attraverso **strumenti quantitativi** per:

- Identificare i **fattori chiave** che scatenano l'abbandono
- Stilare un **profilo tipico del soggetto** interessato dal Turnover
- Determinare i **contesti** con maggiori tassi di abbandono



I dati

- Dati anagrafici dei venditori
- Dati di produttività
- Dati relativi alle filiali
- Dati relativi al Tutor di filiale
- Dati di contesto
- Dati esterni



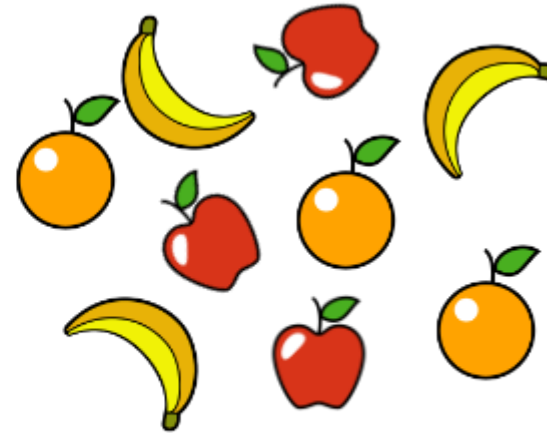
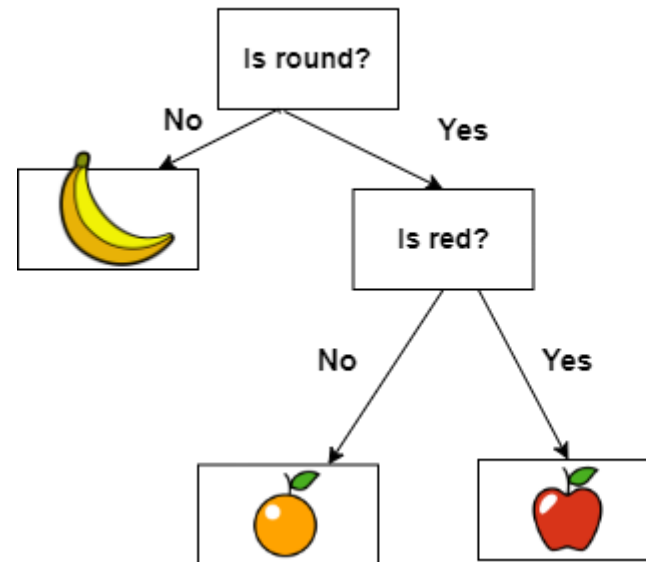
Approccio metodologico

XGBOOST

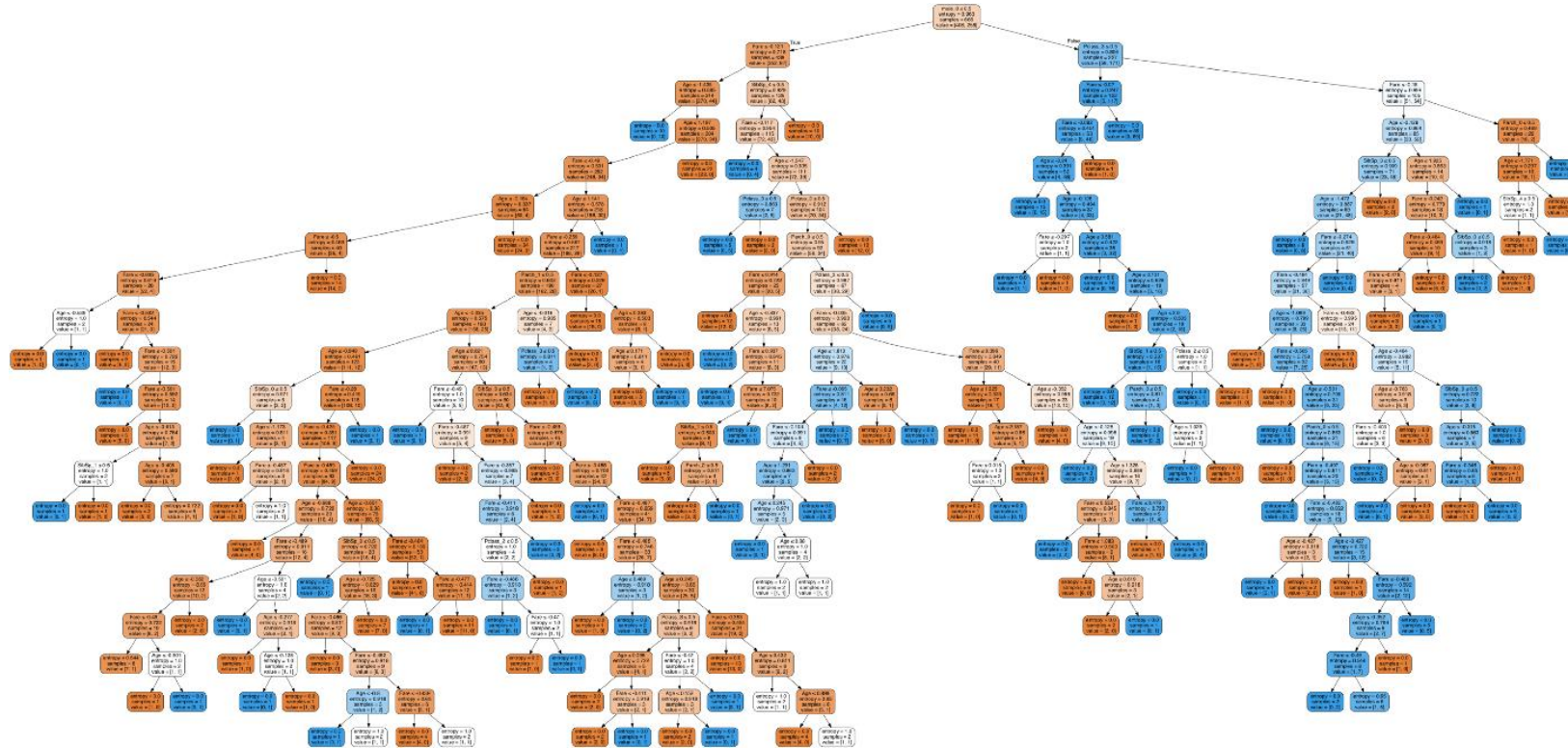
- Ad oggi considerato uno dei modelli più performanti per lavorare su dati strutturati
- «*Ensemble*»: utilizza diversi modelli per ottenere un risultato migliore.
- Permette di massimizzare la capacità predittiva senza incorrere in problemi di *overfitting*

Classificatore XGBOOST

- Basato sugli Alberi Decisionali (Decision Trees)



Classificatore XGBOOST



Molto potente..

...Ma non facilmente interpretabile!

Approccio metodologico - SHAP

SHAP (SHapley Additive exPlanations)

- Approccio coniato nell'ambito della Teoria dei Giochi Cooperativi
- «*Model Agnostic*»: può essere applicato a diversi modelli ML
- Restituisce come output il contributo di una feature ad una particolare prediction

Approccio SHAP (1/3)

- Basato sugli **Shapley values**, coniati da Lloyd Stowell Shapley (1953) nell'ambito della **Teoria dei Giochi Cooperativi**

In questo caso:

- Gioco → Prediction del modello
- Giocatori che cooperano → Features adoperate dal modello

Quindi:

Uno Shapley Value è il contributo marginale medio di una feature tra tutte le possibili coalizioni.



Approccio SHAP (2/3)

Uno Shapley value soddisfa le seguenti condizioni:

- 1) Tutti i guadagni ottenuti dalla cooperazione sono distribuiti tra i giocatori (feature), nessuno è sprecato.
- 2) I giocatori (features) che presentano un contributo uguale ricevono un payoff uguale.
- 3) Il gioco (prediction) non può essere diviso in un set di giochi più piccoli che insieme raggiungono lo stesso risultato.
- 4) Un giocatore che ha contributo marginale pari a zero riceve zero payoff.

Approccio SHAP (3/3)

Come viene calcolato lo Shapley value?

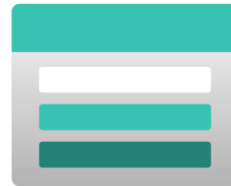
- Per ciascuna prediction si calcola il contributo marginale della singola feature per tutte le possibili coalizioni di variabili che spiegano la prediction.
- Si ripete il processo su ogni variabile e si ottengono gli Shapley values per una singola prediction
- Si ripete su tutte le variabili per tutte le prediction del dataset e si ottiene la matrice di Shapley values

Data preparation & Feature Engineering

Data preparation & Feature Engineering

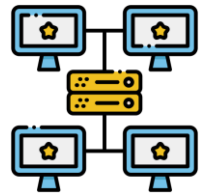
Risorse utilizzate:

- Notebook su Azure Databricks
- Storage Account (V2)



Data preparation & Feature Engineering

Cluster utilizzato



UI | [JSON](#)

Summary

1 Driver	14 GB Memory, 4 Cores
Runtime	10.4.x-scala2.12

Standard_DS3_v2

0.75 DBU/h

single_node_cluster

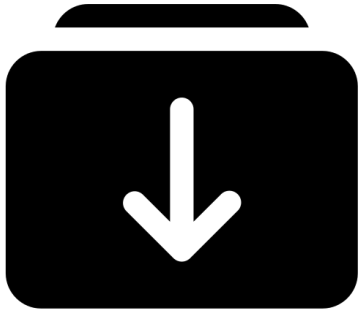
Configuration Notebooks (0) Libraries Event log Spark U

Uninstall

Install new

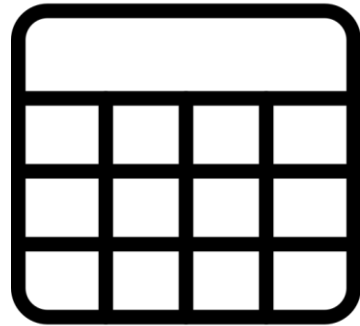
<input type="checkbox"/>	Name	Type
<input type="checkbox"/>	xlrd	PyPI
<input type="checkbox"/>	openpyxl	PyPI
<input type="checkbox"/>	skope-rules	PyPI
<input type="checkbox"/>	scikit-learn==0.22	PyPI
<input type="checkbox"/>	treeinterpreter	PyPI
<input type="checkbox"/>	shap	PyPI
<input type="checkbox"/>	xgboost	PyPI

Data preparation & Feature Engineering



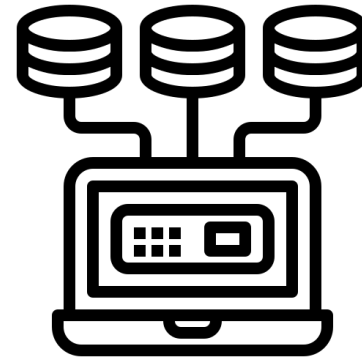
Fase 1

Import dei dati
grezzi



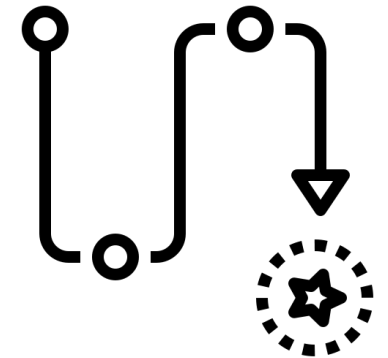
Fase 2

Costruzione
delle tabelle



Fase 3

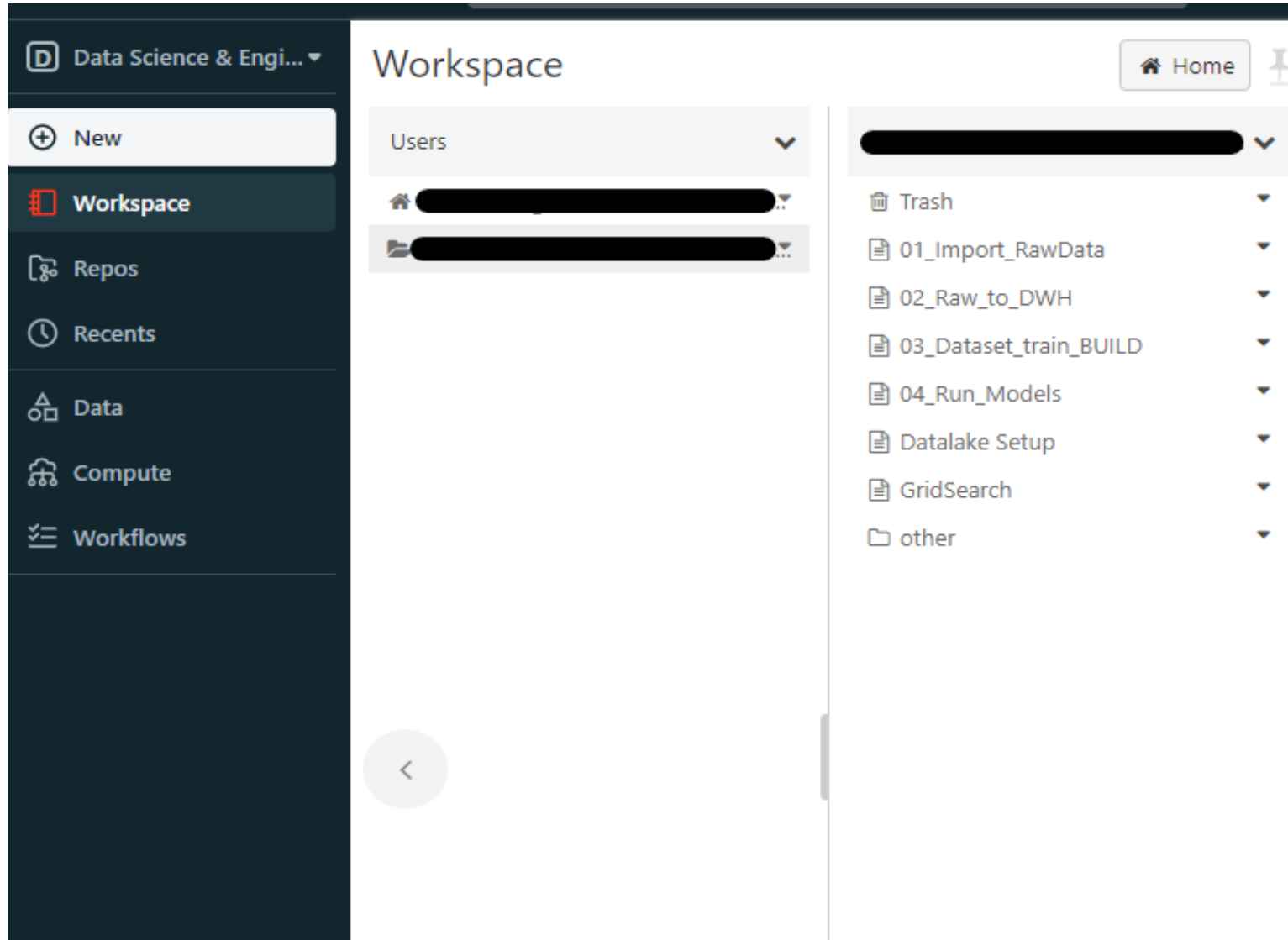
Costruzione
del dataset



Fase 4

Run dei
modelli

Data preparation & Feature Engineering



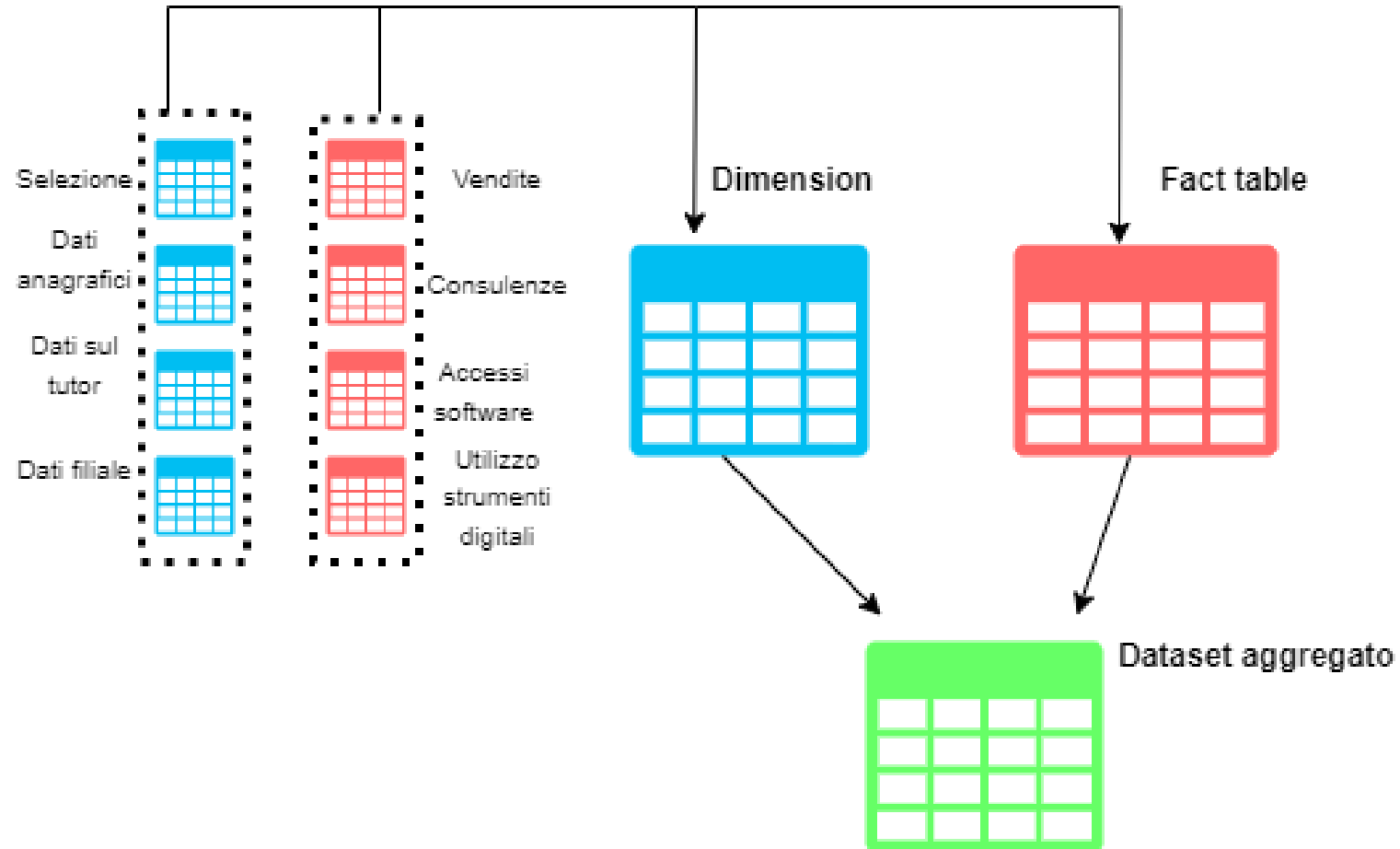
Data preparation & Feature Engineering

- Import dei dati

```
1 def import_excel(PATH, table_destination, header_row=0, sheet_name=0, single_file=False, single_file_name=None, stop_at=slice(0, None, None)):
2     if single_file:
3         df = pd.read_excel(PATH+single_file_name, header=header_row, sheet_name=sheet_name)
4         sparkDF=spark.createDataFrame(df)
5         for column in sparkDF.columns:
6             sparkDF = sparkDF.withColumnRenamed(column, column.replace(" ", "_"))
7         print("Writing SparkDF...")
8         sparkDF.write.mode("overwrite").saveAsTable(table_destination)
9         print(f"Writing completed at {table_destination}")
10    if not single_file:
11        file_names = glob.glob(PATH + "*.xlsx")
12        for file in file_names[stop_at]:
13            df = pd.read_excel(file, header=header_row, sheet_name=sheet_name)
14            print("pandas df created. Creating SparkDF...")
15            sparkDF=spark.createDataFrame(df)
16            for column in sparkDF.columns:
17                sparkDF = sparkDF.withColumnRenamed(column, column.replace(" ", "_"))
18            print("Writing SparkDF...")
19            sparkDF.write.mode("append").saveAsTable(table_destination)
20            print(f"Writing completed at {table_destination}")
```

Data preparation & Feature Engineering

Costruzione Dimensione e Fact Tables



Data preparation & Feature Engineering

- Costruzione Dimensione e Fact Tables

02_Raw_to_DWH Python ▾

File Edit View Run Help Last edit was 7 minutes ago Give feedback

Run all Connect ▾


Cmd 11

```
1 %sql
2 SELECT
3 ROW_NUMBER() OVER (PARTITION BY 1 ORDER BY 1 DESC) sk_agente
4 ,idAgente
5 ,nominativo
6 ,codice_fiscale
7 ,comune_di_residenza
8 ,provincia
9 ,CAST(data_di_nascita AS DATE) AS data_di_nascita
10 ,sexo
11 ,titolo_di_studio
12 ,cast(data_caricamento_gestionale as DATE) AS data_caricamento_gestionale
13 ,data_completamento_formazione
14 ,CAST(data_abbandono AS DATE) AS data_abbandono
15 ,ultima_qualifica
16 ,Regione_inserimento
17 ,Filiale_inserimento
18 ,idTrainer
19 ,nominativo_Trainer
20 ,data_inizio_del_Trainer
21 ,%_completamento_obbligatori
22 ,tot_iniziativa_incentivanti
23 ,DIMENSIONE_Filiale
24 ,ETA_MEDIA_Filiale
25 ,Tasso_Junior_Filiale
26 ,ULTIMO_Portafogli
27 ,MEDIA_Portafogli
28 ,Tasso_vendite_digitali_trainer
29 ,Tasso_vendite_digitali_filiale
30 ,Utilizzo_software_trainer
31 ,Utilizzo_software_filiale
32 .tasso_disoccupazione
```

Data preparation & Feature Engineering

- Costruzione Dimensione e Fact Tables

```
214  
215 dim_agente = spark.sql(queryDimAgente)  
216 dim_agente.write.mode("overwrite").option("overwriteSchema", "true").saveAsTable("dwh.dim_agente")
```

▶  dim_agente: pyspark.sql.dataframe.DataFrame = [redacted] 43 more fields]

Command took 21.61 seconds -- by andrea.bergonzi@dataskills.onmicrosoft.com at 14/9/2022, 15:23:56 on unknown cluster

Cmd 17

```
1 %sql  
2 SELECT  
3   idAgente  
4   ,REPLACE(pp.annomese, "/", "") ANNOMESE  
5   --,CAST(REPLACE(pp.annomese, "/", "") AS INT) ANNOMESE  
6   ,nr_contratti  
7   ,nr_macro_prodotto  
8   ,nr_prodotto  
9   ,tot_fatturato  
10  ,SK_AGENTE  
11 FROM dwh.[redacted] pp  
12 LEFT JOIN [redacted]  
13 ON [redacted]  
14 WHERE SK_AGENTE IS NOT NULL
```

▶  _sqldf: pyspark.sql.dataframe.DataFrame = [redacted] 5 more fields]

Data preparation & Feature Engineering

- Costruzione Dataset finale

Cmd 1

```
1 import pandas as pd
2 import numpy as np
3 from pyspark.sql.functions import col, lit, when, substring, datediff, year, current_timestamp, regexp_replace, coalesce
4 from pyspark.sql.types import *
5 DO_ALL = False
6 from datetime import datetime
```

Command took 0.39 seconds -- by andrea.bergonzi@dataskills.onmicrosoft.com at 26/9/2022, 10:07:41 on unknown cluster

Cmd 2

```
1 dataset = spark.sql("""select * from dwh.dataset""")
2 target_profiles = [REDACTED]
3 dataset = dataset.filter(col("ultima_qualifica").isin(target_profiles))
4 dataset = (dataset.withColumn("hasLeft", when(col("data_cancellazione").isNotNull(), lit(True)).otherwise(lit(False)))
5             .withColumn("ETA_ANAGRAFICA", substring(coalesce(col("data_richiesta_cancellazione"), lit("2022-05-30")).cast(StringType()), 0, 4).cast(IntegerType()) -
6             substring(col("data_di_nascita").cast(StringType()), 0, 4).cast(IntegerType()))
7             .withColumn("delta_days_formazione", datediff(col("data_completamento_formazione"), col("data_caricamento_gestionale")))
8             .withColumn("data_inizio_del_TS", col("data_inizio_del_TS").cast(TimestampType()))
9             .withColumn("Anzianita_Trainer", substring(col("ANNOMESE").cast(StringType()), 0, 4).cast(IntegerType()) - year(col("data_inizio_del_trainer"))) )
10
11 dataset.write.mode("overwrite").option("overwriteSchema", "true").saveAsTable("dwh.dataset")
12 dataset.createOrReplaceTempView("df")
```

▶ dataset: pyspark.sql.dataframe.DataFrame = [REDACTED]double, ANNOMESE: string ... 59 more fields]

Command took 14.34 seconds -- by andrea.bergonzi@dataskills.onmicrosoft.com at 25/9/2022, 19:26:52 on unknown cluster

Data preparation & Feature Engineering

- Costruzione Dataset finale

```
mqa = spark.sql("""select * from stage [redacted]""")
mqa = (mqa.withColumn("PERC_ANNUA", regexp_replace(col('PERC_ANNUA'), " %", ""))
      .withColumn("PERC_AL_MESE", regexp_replace(col('PERC_AL_MESE'), " %", ""))
      )
mqa = (mqa.withColumn("PERC_ANNUA", regexp_replace(col('PERC_ANNUA'), ",", ".").cast(DoubleType()))
      .withColumn("PERC_AL_MESE", regexp_replace(col('PERC_AL_MESE'), ",", ".").cast(DoubleType()))
      )
for c in mqa.columns:
    mqa = mqa.withColumnRenamed(c, c.replace("%", ""))
mqa = (mqa.withColumn("gennaio", regexp_replace(col('gennaio'), " %", ""))
      .withColumn("febbraio", regexp_replace(col('febbraio'), " %", ""))
      .withColumn("marzo", regexp_replace(col('marzo'), " %", ""))
      .withColumn("aprile", regexp_replace(col('aprile'), " %", ""))
      .withColumn("maggio", regexp_replace(col('maggio'), " %", ""))
      .withColumn("giugno", regexp_replace(col('giugno'), " %", ""))
      .withColumn("luglio", regexp_replace(col('luglio'), " %", ""))
      .withColumn("agosto", regexp_replace(col('agosto'), " %", ""))
      .withColumn("settembre", regexp_replace(col('settembre'), " %", ""))
      .withColumn("ottobre", regexp_replace(col('ottobre'), " %", ""))
      .withColumn("novembre", regexp_replace(col('novembre'), " %", ""))
      .withColumn("dicembre", regexp_replace(col('dicembre'), " %", ""))
      )
mqa = (mqa.withColumn("gennaio", regexp_replace(col('gennaio'), ",", ".").cast(DoubleType()))
      .withColumn("febbraio", regexp_replace(col('febbraio'), ",", ".").cast(DoubleType()))
      .withColumn("marzo", regexp_replace(col('marzo'), ",", ".").cast(DoubleType()))
      .withColumn("aprile", regexp_replace(col('aprile'), ",", ".").cast(DoubleType()))
      .withColumn("maggio", regexp_replace(col('maggio'), ",", ".").cast(DoubleType()))
      .withColumn("giugno", regexp_replace(col('giugno'), ",", ".").cast(DoubleType()))
      .withColumn("luglio", regexp_replace(col('luglio'), ",", ".").cast(DoubleType()))
      .withColumn("agosto", regexp_replace(col('agosto'), ",", ".").cast(DoubleType()))
      .withColumn("settembre", regexp_replace(col('settembre'), ",", ".").cast(DoubleType()))
      .withColumn("ottobre", regexp_replace(col('ottobre'), ",", ".").cast(DoubleType()))
      .withColumn("novembre", regexp_replace(col('novembre'), ",", ".").cast(DoubleType()))
      )
```

Data preparation & Feature Engineering

- Costruzione Dataset finale

Cmd 15

```
1 dataset.write.mode("overwrite").option("overwriteSchema", "true").saveAsTable("dwh.dataset_enriched")
```

Command took 3.76 minutes -- by andrea.bergonzi@dataskills.onmicrosoft.com at 26/9/2022, 10:46:51 on unknown cluster

Implementazione del modello

Come?

- Pipeline automatizzata con selezione manuale del "Filtro"
- Filtri relativi al periodo di interesse o variabili
- Run ed esame dei risultati in un processo iterativo per escludere determinate variabili e apportare modifiche

Implementazione del modello

04_Run_Models (1) Python ▾

File Edit View Run Help [Last edit was 1 minute ago](#) [Give feedback](#)

Run Models

Cmd 8

```
1 # SELEZIONARE IL FILTRO DA APPLICARE
2 FILTRO = "FULL"
3 #FILTRO = "Over_12_Months"
4 #FILTRO = "Between_6_and_12_Months"
5 #FILTRO = "Under_6_Months"
6
7
8 Only2020 = False # con/senza variabili che hanno dati dal 2020 in poi
9 isAccount = False
10 noRidondanze = True
```

Cancelled

Command took 14.19 seconds -- by andrea.bergonzi@dataskills.onmicrosoft.com at 21/10/2022, 14:16:08 on unknown cluster

Cmd 9

```
1 dataset = spark.sql("""select * from dwh.dataset_enriched""")
2
3 if isAccount == True:
4     dataset = dataset.filter(col(██████████).isNotNull()) # ONLY ACCOUNT
5 if Only2020 == True:
6     dataset=dataset.filter((col("data_caricamento_gestionale") >= '2020-01-01'))
```

Cancelled

Command took 11.86 seconds -- by andrea.bergonzi@dataskills.onmicrosoft.com at 21/10/2022, 14:16:11 on unknown cluster

Implementazione del modello

Cmd 10

Python



```
1 dataset = (dataset.withColumn("Under_6_Months", when(
2     (col("data_caricamento_gestionale") < '2021-11-23') &
3     (
4         ((col("hasLeft") == True) & (col("NR_MESI_ATTIVITA") < 6))
5         | (col("hasLeft") == False)
6     ), lit(True)).otherwise(lit(False))
7
8     )
9     .withColumn("Over_12_Months", when(
10         (col("data_caricamento_gestionale") < '2021-05-23') &
11         (col("NR_MESI_ATTIVITA") >= 12), lit(True)).otherwise(lit(False))
12     )
13     .withColumn("Between_6_and_12_Months", when(
14         (col("data_caricamento_gestionale") < '2021-11-23') &
15         (((col("hasLeft") == True) & (col("NR_MESI_ATTIVITA") < 12) & (col("NR_MESI_ATTIVITA") >= 6))
16         | (col("hasLeft") == False)) &
17         (col("NR_MESI_ATTIVITA") >= 6)), lit(True)).otherwise(lit(False))
18     )
19 )
20 if FILTRO == "Under_6_Months":
21     dataset = dataset.filter(col("Under_6_Months") == True)
22 if FILTRO == "Between_6_and_12_Months":
23     dataset = dataset.filter(col("Between_6_and_12_Months") == True)
24 if FILTRO == "Over_12_Months":
25     dataset = dataset.filter(col("Over_12_Months") == True)
```

Cmd 11

Implementazione del modello

Ritocchi finali di Feature Engineering...

Cmd 13



Python



```
1 dataset = dataset.filter(col("Ratio_Regionale_Last_3_Months") <= 1) # Clean bad data on Ratio_Regionale_Last_3_Months
2 dataset = dataset.withColumn("Trainer_cartificato", when(col("Fascia_di_certificazione").isNotNull(), lit(1)).otherwise(lit(0)))
3 dataset = dataset.withColumn("date_to_filter", when(col("Ratio_Regionale_Last_3_Months") == 0, lit('2022-05-01')).otherwise(lit('2022-06-30')))
4 dataset = dataset.filter(col("data_caricamento_gestionale") < col("date_to_filter")) # Se la performance è pari a 0, prendo solo quelli inseriti prima di maggio 2022
5 dataset = (dataset.withColumn("HasLaureaOrMaster", when(
6                                     (col("titolo_di_studio") == "laurea") |
7                                     (col("titolo_di_studio") == "Master")
8                                     ,lit(1)).otherwise(lit(0))
9                                     ))
10 dataset = dataset.withColumn("DIGITAL_OVER_80", when((col("DIGITAL_RATE_TRAINER") >= 80.0), lit(1)).otherwise(lit(0)))
```

Implementazione del modello

Ritocchi finali di Feature Engineering...

Cmd 23

```
1 def replace_special_char(txt):  
2     t=txt.replace(' ', '_').replace('/', '').replace('(', '_').replace(')', '_').replace('/', '').replace('\\', '').replace('; ', '').replace('.', '')  
3     return t
```

Command took 0.02 seconds -- by andrea.bergonzi@dataskills.onmicrosoft.com at 17/10/2022, 09:11:43 on unknown cluster

Cmd 24

```
1 train_encoded = pd.get_dummies(train_raw[feature_names], drop_first = True)  
2 feature_names_encoded = [replace_special_char(c) for c in train_encoded.columns]  
3 # Fill nulls with average of the column  
4 numeric_cols = [c for c in train_raw.columns if train_raw[c].dtype in ("int32", "int64", "float") and c in feature_names]  
5 for column in numeric_cols:  
6     train_encoded[column] = train_encoded[column].fillna(train_encoded[column].mean())  
7 X=train_encoded.values
```

Python



Command took 0.05 seconds -- by andrea.bergonzi@dataskills.onmicrosoft.com at 17/10/2022, 09:11:43 on unknown cluster

Implementazione del modello

- Fit & Predict

split and predict for confusion matrix

Cmd 38

```
1 X_train, X_test, y_train, y_test = train_test_split(train_encoded, y, test_size=0.25)
2 model = xgboost.XGBClassifier()
3 model.fit(X_train, y_train)
4 expected_y = y_test
5 predicted_y = model.predict(X_test)
6 print(metrics.classification_report(expected_y, predicted_y))
7 print(metrics.confusion_matrix(expected_y, predicted_y))
```

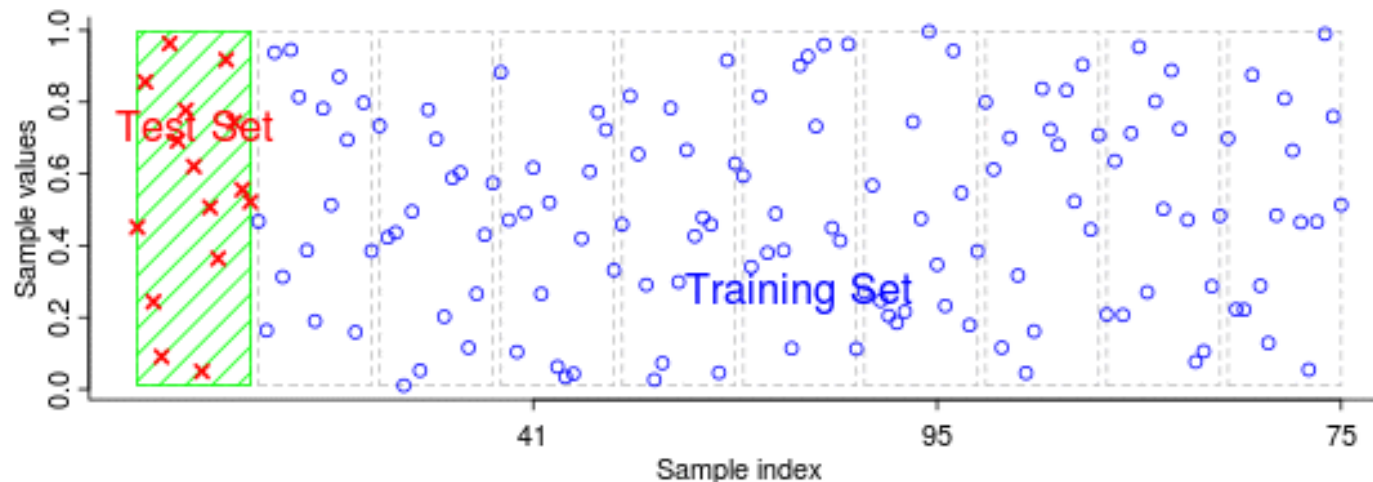
Python ▶ ▼ - ✕

precision recall f1-score support

Implementazione del modello

Cross Validation:

- Il dataset è suddiviso in **k** sottoinsiemi determinati casualmente (k-fold).
- A turno si utilizzano **k-1** sottoinsiemi per il training e 1 sottoinsieme per il test
- Infine sono calcolati i valori medi delle metriche.



Implementazione del modello

Ricerca degli Iperparametri, tra cui:

n_estimators: Numero di alberi utilizzati

learning_rate: Tasso di apprendimento del modello

max_depth: profondità massima degli alberi

Come?

«**Randomized Search**» massimizzando l'f1-score.

Implementazione del modello

Cross Validation, ricerca degli iperparametri

Cmd 31

```
1 from sklearn.model_selection import RandomizedSearchCV, GridSearchCV
2 from sklearn.metrics import roc_auc_score
3 from sklearn.model_selection import StratifiedKFold
4
5 params = {
6     'n_estimators' : np.arange(500, 10000, 500),
7     'learning_rate' : np.arange(0.01, 0.05, 0.005),
8     'max_depth' : np.arange(2, 8, 2),
9     'min_child_weight': np.arange(1, 10, 1),
10    'gamma': np.arange(0.0, 2, 0.1),
11    'subsample': np.arange(0.5, 1, 0.1),
12    'colsample_bytree': np.arange(0.5, 1, 0.1),
13    'lambda': np.arange(0.5, 5, 0.2),
14    'alpha' : np.arange(0.0, 3, 0.2)
15 }
16
17 folds = 5
18 param_combinations = 20
19
20 skf = StratifiedKFold(n_splits=folds, shuffle = True, random_state=95)
21 random_search = RandomizedSearchCV(xgb, param_distributions=params, n_iter=param_combinations, scoring='f1', n_jobs=4, cv=skf.split(train_encoded,y), verbose=3,
22    random_state=95 )
23 random_search.fit(train_encoded, y)
```

```
Fitting 5 folds for each of 20 candidates, totalling 100 fits
[Parallel(n_jobs=4)]: Using backend LokyBackend with 4 concurrent workers.
[Parallel(n_jobs=4)]: Done 24 tasks | elapsed: 17.2min
[Parallel(n_jobs=4)]: Done 100 out of 100 | elapsed: 66.2min finished
```

Implementazione del modello

Iperparametri «ottimali»

Best estimator:

```
XGBClassifier(alpha=0.2, base_score=0.5, booster='gbtree', callbacks=None,
              colsample_bylevel=1, colsample_bynode=1,
              colsample_bytree=0.7999999999999999, early_stopping_rounds=None,
              enable_categorical=False, eval_metric=None,
              gamma=1.9000000000000001, gpu_id=-1, grow_policy='depthwise',
              importance_type=None, interaction_constraints='',
              lambda=2.6999999999999993, learning_rate=0.01, max_bin=256,
              max_cat_to_onehot=4, max_delta_step=0, max_depth=6, max_leaves=0,
              min_child_weight=8, missing=nan, monotone_constraints='()',
              n_estimators=9500, n_jobs=1, nthread=1, num_parallel_tree=1,
              objective='binary:logistic', ...)
```

Best score:

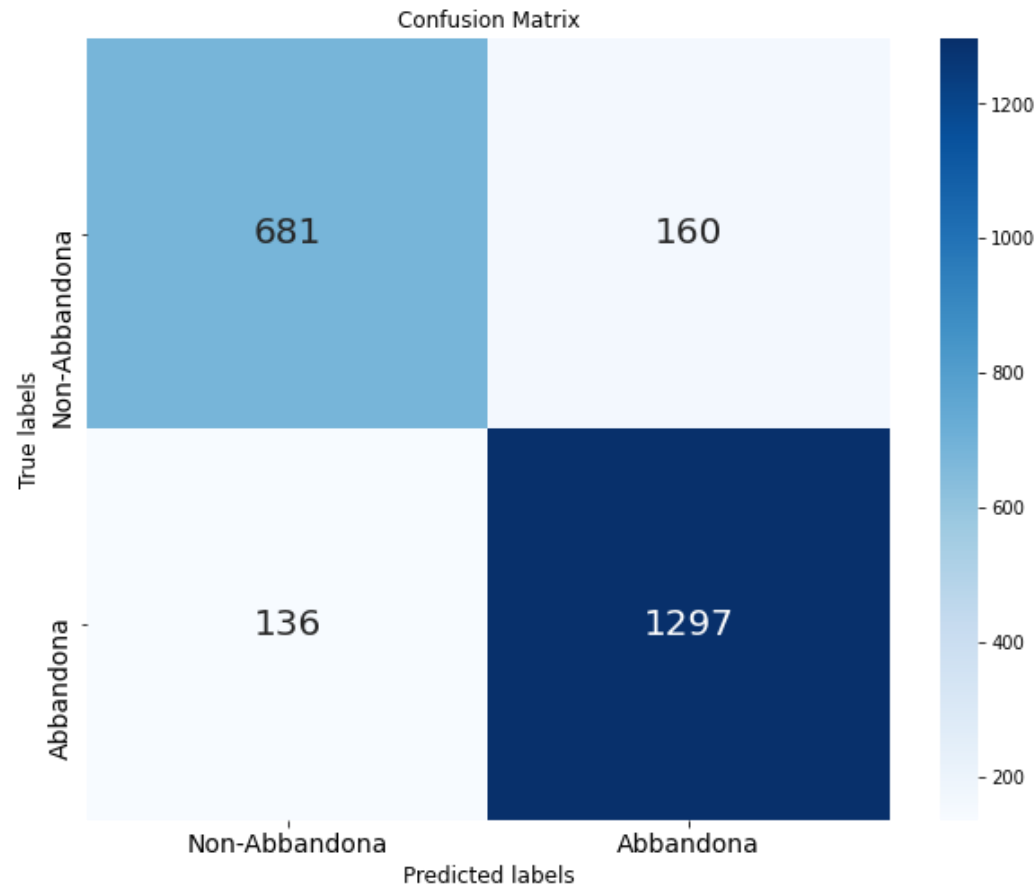
0.8636028959726294

Best hyperparameters:

```
{'subsample': 0.7999999999999999, 'min_child_weight': 8, 'max_depth': 6, 'lambda': 2.6999999999999993, 'gamma': 1.9000000000000001, 'colsample_bytree': 0.7999999999999999, 'alpha': 0.2}
```

Implementazione del modello

Matrice di confusione



	precision	recall	f1-score	support
0	0.83	0.81	0.82	841
1	0.89	0.91	0.90	1433
accuracy			0.87	2274
macro avg	0.86	0.86	0.86	2274
weighted avg	0.87	0.87	0.87	2274

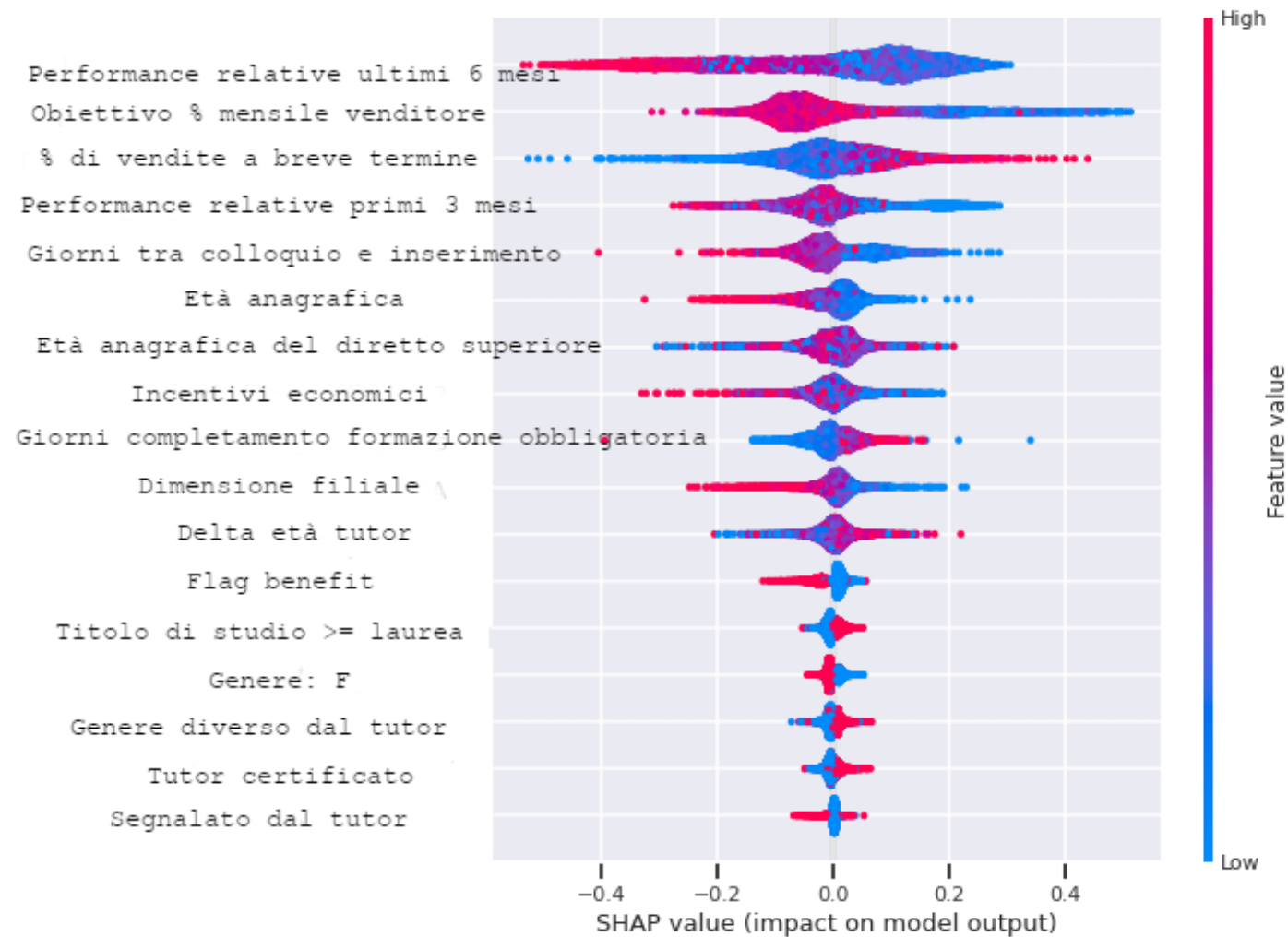
Lettura e Interpretazione dei risultati

Calcolo SHAP values e plot

```
Cmd 47
1 import matplotlib.pyplot as plt
2 from random import randrange
3 explainer = shap.Explainer(model)
4 shap_values = explainer(train_encoded)
5 shap.plots.beeswarm(shap_values, max_display=100, show=False)
6 current_timestamp = datetime.now().strftime("%Y%m%d_%H%M%S")
7 random_integer = randrange(1000000000)
8 plt.savefig(f"/dbfs/mnt/plots/{FILTRO}_{current_timestamp}_{random_integer}.png", dpi=300, bbox_inches = "tight")
9 dbutils.fs.cp(f'dbfs:/mnt/plots/{FILTRO}_{current_timestamp}_{random_integer}.png',
f'wasbs://{redacted}.windows.net/{FILTRO}_{current_timestamp}_{random_integer}.png')
```

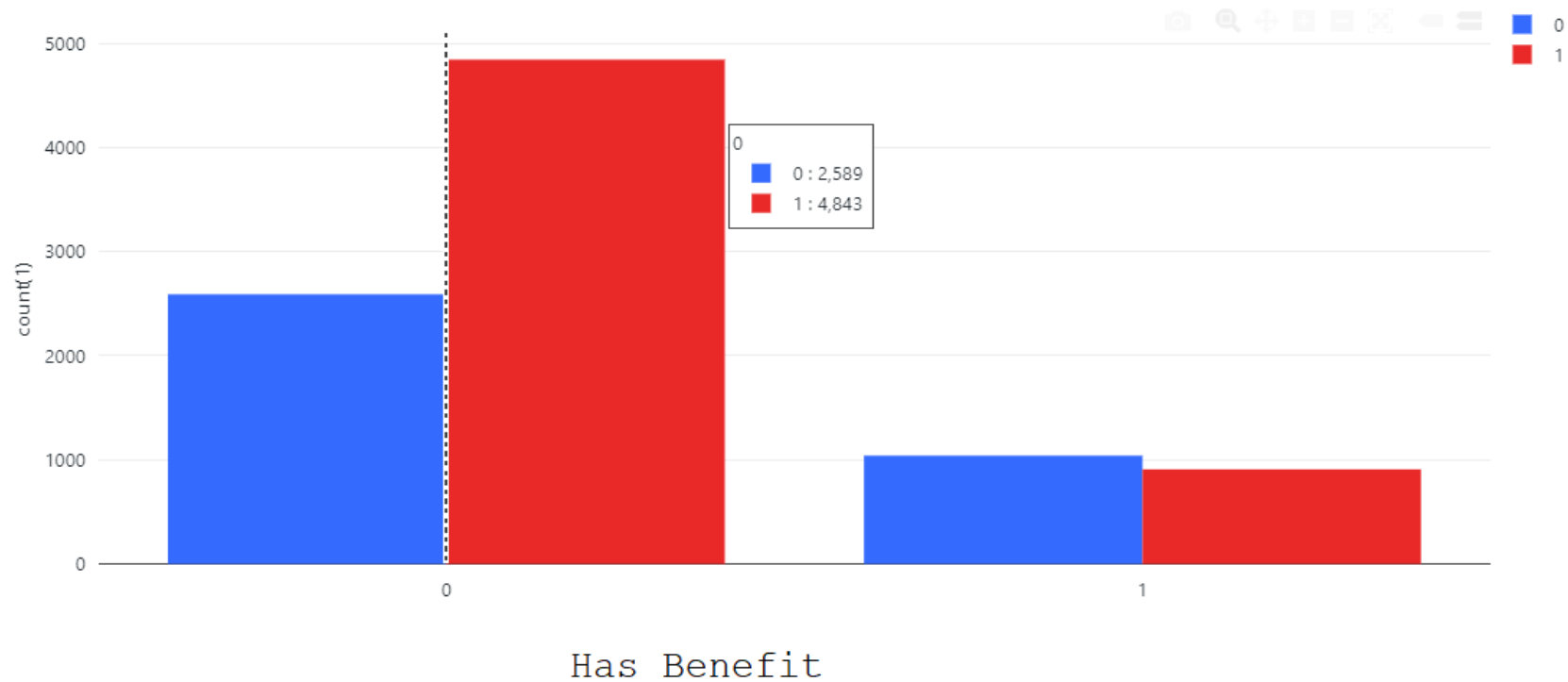
Lettura e Interpretazione dei risultati

Beeswarm Plot



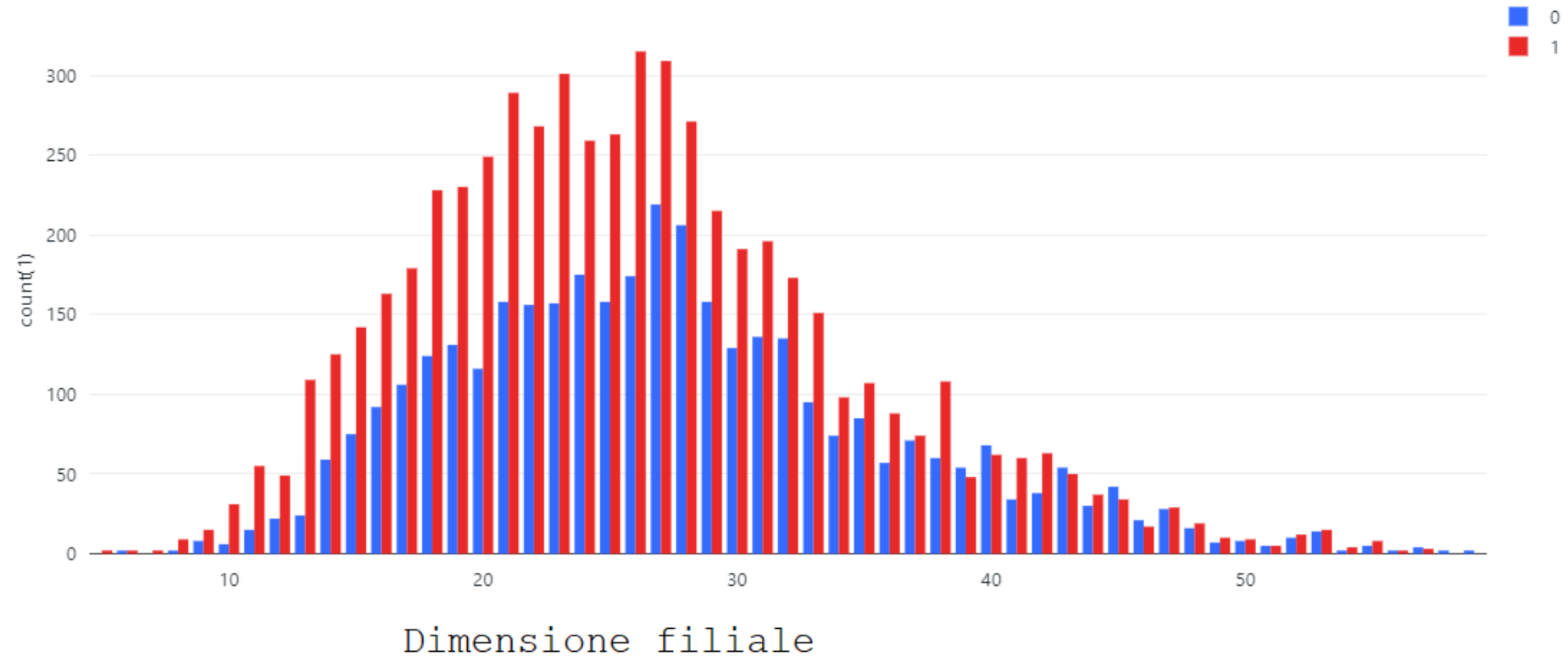
Lettura e Interpretazione dei risultati

Storytelling e contesto delle variabili



Lettura e Interpretazione dei risultati

Storytelling e contesto delle variabili



Business outcome – Alcuni insights

Fattori maggiormente associati al Turnover:

- Scarse performance nei mesi precedenti all'abbandono
- Difficoltà in fase di avviamento
- Mancato raggiungimento degli obiettivi mensili a livello di filiale
- Inserimento in Filiale votata alla vendita a breve termine



Business outcome

- Dataset con variabili aggregate riguardanti i venditori
- Individuazione dei fattori chiave associati all'abbandono
- Definizione di profili e contesti che favoriscono il Turnover
- Possibilità di analizzare nel dettaglio i tassi di abbandono



Next Steps



- Standardizzazione e automatizzazione del flusso dati (es. Data Factory)
- Creazione di un "Tableau de Bord" per identificare i soggetti a rischio (es. Power BI)
- Trigger su azioni mirate a contrastare il turnover sui soggetti a rischio



Grazie!!!

