In [1]:
```
# @hidden_cell
# The project token is an authorization token that is used to access project r
esources like data sources, connections, and used by platform APIs.
from project_lib import Project
project = Project(spark.sparkContext, '700d3b18-c5e9-4e66-a15a-07f6ac88f776',
'p-e8ed794dd3515b31f89e141b36e18d4c86b7f5ef')
pc = project.project_context
```

```
Waiting for a Spark session to start...
Spark Initialization Done! ApplicationId = app-20210519164324-0001
KERNEL_ID = ed46d11b-2e05-426a-887e-3dc2478ba505
```

Predicting Beneficiary Claims Income


```
Extended by Dr. H. Völlinger, DHBW Stuttgart, 12.05.2021
```

In this notebook, you will learn to train a model that takes four sets of beneficary-related data sets to predict beneficiary claims income for customers. This notebook runs on the latest Python version and Spark.

This sample project contains a notebook and four data assets that offer hands-on experience with the Watson Health Analytics Workbench. The notebook includes live code that accesses the sample data assets stored in COS, prepares those data, and builds a machine-learning model using the PySpark interface. The sample data assets include synthetic healthcare beneficiary claims data publicly available from the Agency for Healthcare Research and Quality (HRAC). The data records contain ICD9 diagnosis for claims data that follow the FHIR specification. One of the data assets contains clinical classification categories to which the diagnostic codes from the claims data are converted. The notebook trains a linear regression model from a subset of the data, and validates it on another, validation subset. The model predicts future medical service costs. The notebook runs on the latest Python version and Spark.

Input data - We use the following four files. The structure and formats of the files are known medical data formats for documenting of medical content:

1. $dxref_2015.csv: contains 6 data fields used for the classification of medical conditions. Number of data records (#) = 15075
2. Patient.ndjson: information about the patient, such as age, place of residence, etc; #=50
3. Coverage.ndjson: patient, payment, disease, etc; #=200
4. ExplanationOfBenefit.ndjson: payment information, diagnoses, etc; #=1588

Table of Contents

Step1: Pre-requisites

Insert a project token

When you import this project from the Watson Studio Gallery, a token should be automatically generated and inserted at the top of this notebook as a code cell such as the one below:

```
# @hidden_cell
# The project token is an authorization token that is used to access project resour
ces like data sources, connections, and used by platform APIs.
from project_lib import Project
project = Project(project_id='YOUR_PROJECT_ID', project_access_token='YOUR_PROJECT_
TOKEN')
pc = project.project_context
```

If you do not see the cell above, follow these steps to enable the notebook to access the dataset from the project's resources:

- Click on `More -> Insert project token` in the top-right menu section

- This should insert a cell at the top of this notebook similar to the example given above.

> If an error is displayed indicating that no project token is defined, follow these instructions (https://dataplatform.cloud.ibm.com/docs/content/wsj/analyze-data/token.html?audience=wdp&context=data).

- Run the newly inserted cell before proceeding with the notebook execution below

Start Spark session

In [2]:
```
spark = SparkSession.builder \
        .appName("spark-model-training") \
        .getOrCreate()
```

Authentication and Import of Libraries

In [3]:
```
import ibm_boto3
from botocore.client import Config
import ibmos2spark

# to check execution time we import this library
import time

properties = project.get_storage_metadata().get("properties")
credentials = properties.get("credentials").get("editor")
credentials["endpoint"] = properties.get("endpoint_url")

client = ibm_boto3.client(
    service_name='s3',
    ibm_api_key_id=credentials["api_key"],
    ibm_auth_endpoint='https://iam.ng.bluemix.net/identity/token',
    config=Config(signature_version='oauth'),
    endpoint_url=credentials["endpoint"]
)

project_cos = ibmos2spark.CloudObjectStorage(spark, credentials, "project-cos"
,'bluemix_cos')

print("**** End of Step1 ******")
print("Execution-Date & -Time of Step1:",time.strftime("%d.%m.%Y %H:%M:%S"))
```

```
**** End of Step1 ******
Execution-Date & -Time of Step1: 19.05.2021 16:43:27
```

Step2: Load the data sets

This synthetic dataset only has ICD9 codes for diagnoses. So we're using AHRQ CCS categories to bucket ICD9 codes into ~250 CCS buckets

https://www.hcup-us.ahrq.gov/toolssoftware/ccs/ccs.jsp (https://www.hcup-us.ahrq.gov/toolssoftware/ccs/ccs.jsp)
The file "$dxref 2015.csv" is loaded into the project.

We filter and select the three columns: dx-code (ICD-9-CM), ccs-code and ontolgy(ICD9)

In [4]:
```
from pyspark.sql.functions import col, lit, trim

ccs_df = (
    spark.read
    .format('org.apache.spark.sql.execution.datasources.csv.CSVFileFormat') \
    .option("header", "true") \
    .option("comment", "N") \
    .option("quote", "'") \
    .load(project_cos.url("$dxref 2015.csv", project.get_project_bucket_name
())) \
    .withColumn("dx_code", trim(col("ICD-9-CM CODE"))) \
    .withColumn("ccs_code", trim(col("CCS CATEGORY"))) \
    .withColumn("ontology", lit("ICD9")) \
    .select("dx_code", "ontology", "ccs_code")
)
ccs_df.show()
```

```
+-------+--------+--------+
|dx_code|ontology|ccs_code|
+-------+--------+--------+
|       |    ICD9|       0|
|  01000|    ICD9|       1|
|  01001|    ICD9|       1|
|  01002|    ICD9|       1|
|  01003|    ICD9|       1|
|  01004|    ICD9|       1|
|  01005|    ICD9|       1|
|  01006|    ICD9|       1|
|  01010|    ICD9|       1|
|  01011|    ICD9|       1|
|  01012|    ICD9|       1|
|  01013|    ICD9|       1|
|  01014|    ICD9|       1|
|  01015|    ICD9|       1|
|  01016|    ICD9|       1|
|  01080|    ICD9|       1|
|  01081|    ICD9|       1|
|  01082|    ICD9|       1|
|  01083|    ICD9|       1|
|  01084|    ICD9|       1|
+-------+--------+--------+
only showing top 20 rows
```

The synthetic data was taken from CMS https://bcda.cms.gov/data.html#sample-files (https://bcda.cms.gov/data.html#sample-files)

- The coverage data doesn't have periods so we've added artifical forever periods
- All of the patients are age 1 at event so we artifically changed their birthdates

We import the three ndjson files.

In [5]:

```
from datetime import datetime
from pyspark.sql.functions import lit, struct

#load the first ndjson file
eob = spark.read.json(project_cos.url("ExplanationOfBenefit.ndjson", project.g
et_project_bucket_name()))
eob.show(5)

#load the second ndjson file
#change the start and end date and show results
coverage = spark.read.json(project_cos.url("Coverage.ndjson", project.get_proj
ect_bucket_name()))
coverage = coverage.withColumn("period", struct(
    lit(datetime(1900, 1, 1)).alias("start"),
    lit(datetime(2099, 1, 1)).alias("end")
))
coverage.show(5)

#load the third ndjson file
#change birthDate value ans show results
patient = spark.read.json(project_cos.url("Patient.ndjson", project.get_projec
t_bucket_name()))
patient = patient.withColumn("birthDate", lit(datetime(1960, 1, 1)))
patient.show(5)

print("**** End of Step2 ******")
print("Execution-Date & -Time of Step2:",time.strftime("%d.%m.%Y %H:%M:%S"))
```

```
+--------------+--------------------+--------+--------------------+----------
----------+--------+--------------+--------------------+--------------------+
-----------+--------------------+--------------------+--------------------+--
------------------+---------+--------+--------------------+------+---------+-
-------------------+
|benefitBalance|      billablePeriod|careTeam|           diagnosis|
extension|facility|hospitalization|                  id|          identifier|i
nformation|           insurance|                item|             patient|
payment|procedure|provider|        resourceType|status|totalCost|
type|
+--------------+--------------------+--------+--------------------+----------
----------+--------+--------------+--------------------+--------------------+
-----------+--------------------+--------------------+--------------------+--
------------------+---------+--------+--------------------+------+---------+-
-------------------+
|          null|[2000-10-01,, 200...|    null|[[[[[4011, BENIGN...|[[https://
bluebut...|    null|           null|carrier-10335267275|[[https://bluebut...|
null|[[Coverage/part-b...|[[[[, [[[https://...|[Patient/19990000...|[[USD, ur
n:iso:st...|     null|    null|ExplanationOfBenefit|active|     null|[[[71, L
ocal carr...|
|          null|[2000-11-01,, 200...|    null|[[[[[185, MALIGN ...|[[https://
bluebut...|    null|           null|carrier-10339756766|[[https://bluebut...|
null|[[Coverage/part-b...|[[[[, [[[https://...|[Patient/19990000...|[[USD, ur
n:iso:st...|     null|    null|ExplanationOfBenefit|active|     null|[[[71, L
ocal carr...|
|          null|[2000-06-01,, 200...|    null|[[[[[25000, DMII ...|[[https://
bluebut...|    null|           null|carrier-10344372825|[[https://bluebut...|
null|[[Coverage/part-b...|[[[[, [[[https://...|[Patient/19990000...|[[USD, ur
n:iso:st...|     null|    null|ExplanationOfBenefit|active|     null|[[[71, L
ocal carr...|
|          null|[2000-02-01,, 200...|    null|[[[[[25000, DMII ...|[[https://
bluebut...|    null|           null|carrier-10348851446|[[https://bluebut...|
null|[[Coverage/part-b...|[[[[, [[[https://...|[Patient/19990000...|[[USD, ur
n:iso:st...|     null|    null|ExplanationOfBenefit|active|     null|[[[71, L
ocal carr...|
|          null|[2000-08-01,, 200...|    null|[[[[[71516, LOC P...|[[https://
bluebut...|    null|           null|carrier-10506301927|[[https://bluebut...|
null|[[Coverage/part-b...|[[[[, [[[https://...|[Patient/19990000...|[[USD, ur
n:iso:st...|     null|    null|ExplanationOfBenefit|active|     null|[[[71, L
ocal carr...|
+--------------+--------------------+--------+--------------------+----------
----------+--------+--------------+--------------------+--------------------+
-----------+--------------------+--------------------+--------------------+--
------------------+---------+--------+--------------------+------+---------+-
-------------------+
only showing top 5 rows

+--------------------+--------------------+--------------------+------------
-----+--------------------+------------+------+--------------------+---------
-----------+
|         beneficiary|            contract|           extension|          gro
uping|                  id|resourceType|status|                type|
period|
+--------------------+--------------------+--------------------+------------
-----+--------------------+------------+------+--------------------+---------
-----------+
|[Patient/19990000...|[[ptc-contract1,]...|[[https://bluebut...|[Medicare, Pa
```

```
rt A]|part-a-1999000000...|    Coverage|active|[[[Part A, Medica...|[1900-01-
01 00:00...|
|[Patient/19990000...|               null|[[https://bluebut...|[Medicare, Pa
rt B]|part-b-1999000000...|    Coverage|active|[[[Part B, Medica...|[1900-01-
01 00:00...|
|[Patient/19990000...|               null|                null|[Medicare, Pa
rt C]|part-c-1999000000...|    Coverage|active|[[[Part C, Medica...|[1900-01-
01 00:00...|
|[Patient/19990000...|               null|[[https://bluebut...|[Medicare, Pa
rt D]|part-d-1999000000...|    Coverage|active|[[[Part D, Medica...|[1900-01-
01 00:00...|
|[Patient/19990000...|[[ptc-contract1,]...|[[https://bluebut...|[Medicare, Pa
rt A]|part-a-1999000000...|    Coverage|active|[[[Part A, Medica...|[1900-01-
01 00:00...|
+--------------------+--------------------+--------------------+------------
-----+--------------------+------------+------+--------------------+---------
-----------+
only showing top 5 rows

+------------------+--------------------+--------------------+------+---------
-----+--------------------+--------------------+------------+
|           address|          birthDate|           extension|gender|
id|          identifier|                name|resourceType|
+------------------+--------------------+--------------------+------+---------
-----+--------------------+--------------------+------------+
|[[999, 99999, 05]]|1960-01-01 00:00:00|[[https://bluebut...|  male|199900000
00140|[[https://bluebut...|[[Doe, [John, X],...|     Patient|
|[[999, 99999, 22]]|1960-01-01 00:00:00|[[https://bluebut...|  male|199900000
00141|[[https://bluebut...|[[Doe, [John, X],...|     Patient|
|[[999, 99999, 22]]|1960-01-01 00:00:00|[[https://bluebut...|  male|199900000
00142|[[https://bluebut...|[[Doe, [John, X],...|     Patient|
|[[999, 99999, 22]]|1960-01-01 00:00:00|[[https://bluebut...|female|199900000
00144|[[https://bluebut...|[[Doe, [Jane, X],...|     Patient|
|[[999, 99999, 22]]|1960-01-01 00:00:00|[[https://bluebut...|female|199900000
00145|[[https://bluebut...|[[Doe, [Jane, X],...|     Patient|
+------------------+--------------------+--------------------+------+---------
-----+--------------------+--------------------+------------+
only showing top 5 rows

**** End of Step2 ******
Execution-Date & -Time of Step2: 19.05.2021 16:44:08
```

Step3: Extract and transform the data

Pull out age and gender from the patient resource file and load these data in a new file with the name demographics.

We chose 2000-01-01 because thats what this synthetic dataset supports.

Show the first 20 rows of the resulting file demographics.

In [6]:
```
from pyspark.sql.functions import floor, lit, months_between

demographics = (
    patient.select(
        col("id").alias("patient"),
        "gender",
        floor(months_between(lit(datetime(2000, 1, 1)), "birthDate") / 12).ali
as("age")
    )
)
demographics.show()
```

```
+--------------+------+---+
|       patient|gender|age|
+--------------+------+---+
|19990000000140|  male| 40|
|19990000000141|  male| 40|
|19990000000142|  male| 40|
|19990000000144|female| 40|
|19990000000145|female| 40|
|19990000000146|  male| 40|
|19990000000147|  male| 40|
|19990000000149|  male| 40|
|19990000000150|female| 40|
|19990000000151|female| 40|
|19990000000153|female| 40|
|19990000000154|female| 40|
|19990000000155|female| 40|
|19990000000156|  male| 40|
|19990000000158|female| 40|
|19990000000159|female| 40|
|19990000000160|female| 40|
|19990000000162|  male| 40|
|19990000000163|  male| 40|
|19990000000164|  male| 40|
+--------------+------+---+
only showing top 20 rows
```

We're going to be predicting the total paid $ year n+1 given features calculated on data from year n.

So here we're splitting the data into `features_period` (year n) and `outcome_period` (year n+1) for our ETL

In [7]:
```
from pyspark.sql.functions import col

features_period = eob.where(
    (col("billablePeriod.start") >= datetime(1999, 1, 1)) &
    (col("billablePeriod.start") < datetime(2000, 1, 1))
)

outcome_period = eob.where(
    (col("billablePeriod.start") >= datetime(2000, 1, 1)) &
    (col("billablePeriod.start") < datetime(2001, 1, 1))
)
```

Create a row/vector of CCS codes that each patient has during the `features_period`

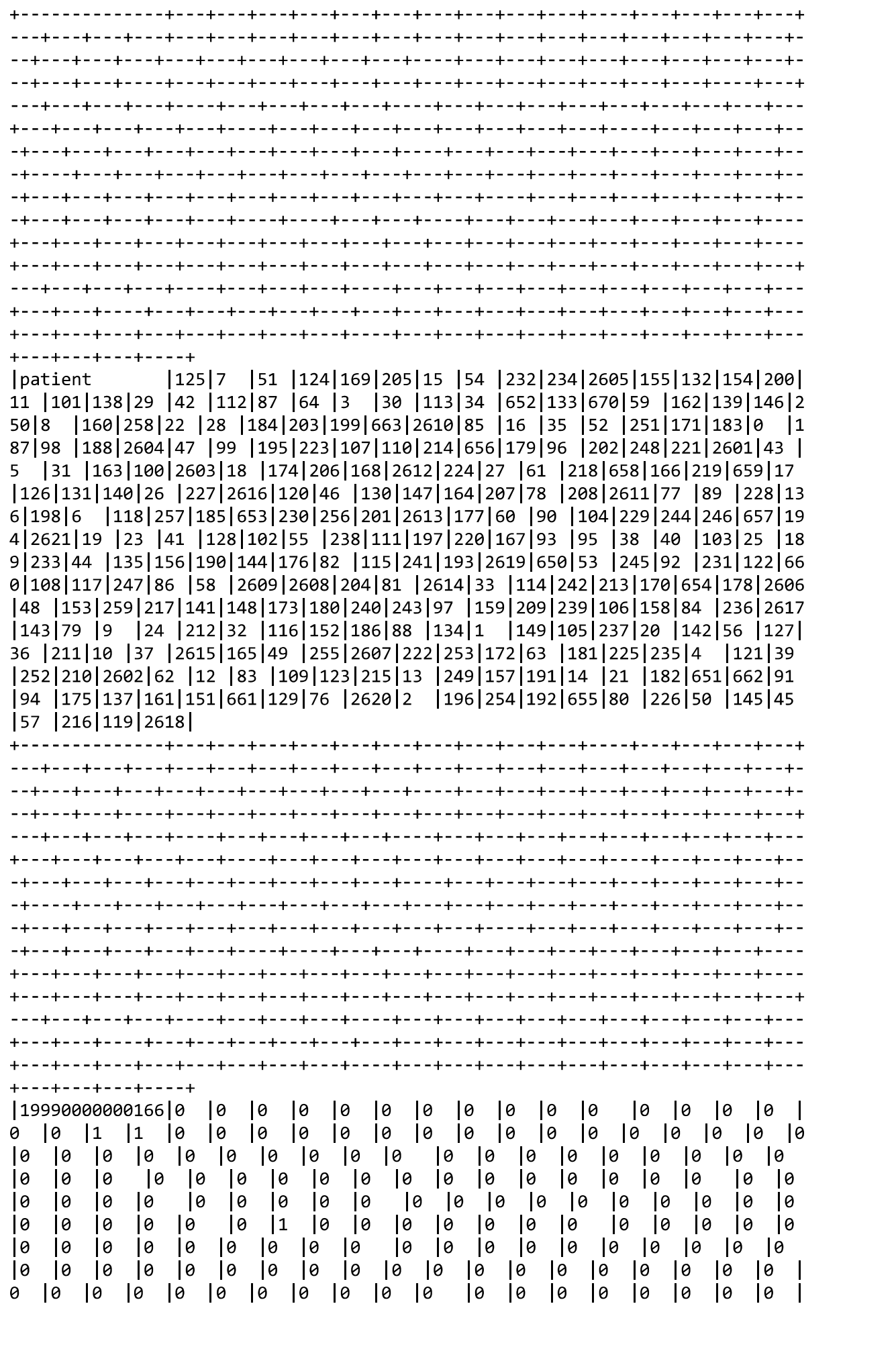
In [8]:

```
from pyspark.sql.functions import countDistinct, explode, substring

ccs_categories = ccs_df.select("ccs_code").distinct().rdd.flatMap(lambda x: x)
.collect()

ccs_codes = (
    # Explode arrays to get to codes
    features_period.select(
        substring("patient.reference", 9, 255).alias("patient"),
        explode("diagnosis").alias("diagnosis")
    ).select(
        "patient",
        col("diagnosis.sequence").alias("sequence"),
        explode("diagnosis.diagnosisCodeableConcept.coding").alias("coding")
    ).select(
        "patient",
        "sequence",
        col("coding.system").alias("system"),
        col("coding.code").alias("dx_code")
    )
    # Map ICD -> CCS
    .join(ccs_df, on="dx_code", how="inner")
    .select("patient", "ccs_code")
    .distinct()
    # Groupby pivot to get 1/0 columns for CCS codes
    .groupby("patient")
    .pivot("ccs_code", ccs_categories)
    .agg(countDistinct("patient"))
    .fillna(0)
)
ccs_codes.show(truncate=False)


print("**** End of Step3 ******")
print("Execution-Date & -Time of Step3:",time.strftime("%d.%m.%Y %H:%M:%S"))
```



0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0
|0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0
|0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |
0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0
|0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |1
|0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0
|0 |0 |
|19990000000138|0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |
0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0
|0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0
|0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0
|0 |0 |0 |0 |0 |0 |1 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0
|0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0
|0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0
|0 |1 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |
1 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |1 |0 |0 |0 |0 |0 |0 |
0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0
|0 |0 |0 |0 |1 |0 |0 |0 |0 |0 |0 |0 |1 |0 |0 |0 |0 |0 |0
|0 |1 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |
0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0
|0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0
|0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0
|0 |0 |
|19990000000151|0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |
0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0
|0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0
|0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0
|0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0
|0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0
|0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0
|0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |
0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |
0 |0 |0 |0 |0 |0 |1 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0
|0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0
|0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |
0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0
|0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0
|0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0
|0 |0 |
|19990000000142|0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |
0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0
|0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |1 |0 |0 |0 |0
|0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0
|0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0
|0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0
|0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0
|0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |
0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |
0 |0 |1 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0
|0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0
|0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |
0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0
|0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0
|0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |1 |0 |0 |0
|0 |0 |
|19990000000181|0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |
0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0

|0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0
|0 |0 |0 |1 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0
|0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0
|0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0
|0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0
|0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |1 |0 |0 |0 |0 |0 |0 |0 |
0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |
0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0
|1 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0
|0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |
0 |0 |0 |0 |0 |0 |0 |0 |1 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0
|0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0
|0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0
|0 |0 |
|19990000000141|0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |
0 |0 |0 |0 |0 |0 |1 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0
|0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0
|1 |0 |0 |0 |0 |0 |0 |0 |1 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0
|0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0
|0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |1 |0 |0 |0
|0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0
|0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |1 |0 |0 |0 |0 |0 |0 |0 |
0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |1 |0 |0 |0 |0 |0 |0 |
0 |0 |1 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0
|0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0
|0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |
0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |1 |0
|0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0
|0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0
|0 |0 |
|19990000000155|0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |
0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0
|0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0
|1 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0
|0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0
|0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0
|0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0
|0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |
0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |1 |0 |0 |0 |0 |0 |0 |
0 |0 |0 |0 |0 |0 |1 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0
|1 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0
|0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |1 |
0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0
|0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0
|0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0
|0 |0 |
|19990000000174|0 |0 |0 |0 |0 |1 |0 |1 |0 |0 |0 |0 |0 |0 |0 |
0 |0 |0 |0 |0 |0 |1 |0 |0 |0 |0 |0 |0 |1 |0 |1 |0 |0 |0 |0
|0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0
|1 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0
|0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0
|0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |1 |0 |0 |0
|0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |1 |0 |0 |0 |0
|0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |
0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |1 |0 |0 |0 |0 |0 |1 |
0 |0 |1 |1 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |1
|1 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0

```
|0 |0 |1 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |1 |0 |0 |
0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0
|0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |1
|0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0
|0 |0 |
|19990000000161|0 |0 |0 |0 |0 |1 |0 |0 |0 |0 |0 |0 |0 |0 |0 |
0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0
|0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0
|0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0
|0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0
|0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0
|0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0
|0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |
0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |
0 |0 |1 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0
|0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0
|0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |
0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0
|0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0
|0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0
|0 |0 |
|19990000000137|0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |1 |
0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0
|0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0
|0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0
|0 |1 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0
|0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0
|0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0
|0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |
0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |
0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0
|0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |1 |0 |0 |0 |0 |0 |0 |0 |0
|0 |0 |1 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |1 |
0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0
|0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0
|0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0
|0 |0 |
|19990000000182|0 |0 |0 |0 |0 |1 |0 |0 |0 |0 |0 |0 |0 |0 |0 |
0 |0 |1 |0 |0 |0 |1 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0
|0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0
|1 |0 |0 |0 |0 |0 |0 |0 |1 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0
|0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0
|0 |0 |0 |0 |0 |0 |0 |0 |0 |1 |0 |0 |0 |0 |0 |0 |0 |0 |0
|0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0
|0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |
0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |
1 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |1 |0
|0 |0 |0 |0 |0 |0 |0 |0 |0 |1 |0 |0 |0 |0 |0 |0 |0 |0 |0
|0 |0 |1 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |
0 |0 |0 |0 |1 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0
|0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0
|0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0
|0 |0 |
|19990000000157|0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |
0 |1 |1 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0
|0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0
|1 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0
```

```
|0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0
|0 |0 |0 |0 |0 |0 |0 |0 |0 |1 |0 |0 |0 |0 |0 |0 |0 |0 |0
|0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0
|0 |0 |0 |0 |1 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |
0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |1 |0 |0 |0 |0 |0 |0 |
0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0
|1 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0
|0 |0 |0 |0 |0 |0 |0 |0 |1 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |
0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0
|0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0
|0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0
|0 |0 |
|19990000000176|0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |1 |
0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0
|0 |0 |0 |0 |0 |0 |1 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0
|0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0
|0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |1
|0 |0 |0 |0 |0 |1 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0
|0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |1 |0 |0 |0 |0 |0 |0 |0
|0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |1 |0 |0 |0 |0 |0 |0 |0 |0 |
1 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |
0 |0 |1 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0
|0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0
|0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |1 |
0 |0 |0 |0 |0 |0 |0 |0 |1 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0
|0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |1 |0
|0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0
|0 |0 |
|19990000000169|0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |
0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0
|0 |0 |0 |0 |0 |0 |1 |0 |0 |0 |0 |0 |0 |1 |0 |0 |0 |0 |0
|0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0
|0 |1 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0
|0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0
|0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0
|0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |1 |0 |0 |0 |0 |0 |0 |
0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |
0 |0 |1 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |1 |0
|0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0
|0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |1 |
0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0
|0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0
|0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0
|0 |0 |
|19990000000170|1 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |
0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0
|0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0
|0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0
|0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0
|0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0
|0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0
|0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |
0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |
0 |0 |1 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0
|0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0
|0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |
0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0
```

|0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0
|0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0
|0 |0 |
|19990000000145|0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |1 |0 |1 |0 |
0 |0 |0 |1 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0
|0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |1 |0 |0 |0 |0
|1 |0 |0 |1 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0
|0 |1 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |1
|0 |0 |0 |0 |0 |0 |0 |0 |0 |1 |0 |0 |0 |0 |0 |0 |0 |0 |0
|0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0
|0 |0 |0 |0 |1 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |
0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |
0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0
|0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0
|0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |1 |0 |0 |0 |0 |0 |0 |0 |0 |
0 |0 |0 |0 |1 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0
|0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0
|0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |1 |0 |0 |0
|0 |0 |
|19990000000183|0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |
0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0
|0 |0 |0 |0 |0 |0 |1 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0
|1 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0
|0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0
|0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0
|0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0
|0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |
0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |1 |0 |0 |0 |0 |0 |0 |
0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0
|0 |0 |0 |0 |1 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0
|0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |
0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0
|0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0
|0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0
|0 |0 |
|19990000000152|0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |
0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0
|0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0
|0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0
|0 |1 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0
|0 |0 |0 |0 |0 |0 |0 |0 |0 |1 |0 |0 |0 |0 |0 |0 |0 |0 |0
|0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0
|0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |
0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |
0 |0 |1 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0
|0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0
|0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |1 |
0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0
|0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0
|0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0
|0 |0 |
|19990000000144|0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |1 |0 |0 |0 |
0 |0 |1 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |1 |0 |0 |0 |0 |0 |0
|0 |0 |0 |0 |0 |0 |1 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0
|1 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0
|0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0
|0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0 |0

```
|0  |0  |0  |0  |0  |0  |0  |0  |0   |0  |0  |0  |0  |0  |0  |0  |0  |0  |0  |0
|0  |0  |0  |0  |0  |0  |0  |0  |0  |0  |0  |0  |0  |0  |0  |0  |0  |0  |0  |0  |
0  |0  |0  |0  |0  |0  |0  |0  |0  |0  |0   |0  |1  |0  |0  |0  |0  |0  |0  |0  |
0  |0  |0  |0  |0   |0   |0  |0  |0   |0  |1  |0  |0  |0  |0  |0  |0   |0  |0
|0  |0  |0  |0  |0  |0  |0  |0  |0  |0  |0  |0  |0  |0  |0  |0  |0   |0  |0
|0  |0  |0  |0  |0  |0  |0  |0  |0  |0  |0  |0  |0  |0  |0  |0  |0  |0  |0  |0  |
0  |0  |0   |0  |0  |0  |0   |0  |0  |0  |0  |0  |0  |0  |0  |0  |0  |0  |0
|0   |0  |0  |0  |0  |0  |0  |0  |0  |0  |0  |0  |0  |0  |0  |0  |0  |0  |0
|1  |0  |0  |0  |0  |0  |0   |0  |0  |0  |0  |0  |0  |0  |0  |0  |0  |0  |0
|1  |0   |
|19990000000175|0  |0  |0  |0  |0  |0  |0  |0  |0  |0  |0   |1  |0  |0  |0  |
0  |0  |0  |0  |0  |0  |0  |0  |0  |0  |0  |0  |0  |0  |0  |0  |0  |1  |0
|0  |0  |0  |0  |0  |0  |1  |0  |0  |0   |0  |0  |0  |0  |0  |0  |0  |0  |0
|1  |0  |0   |0  |0  |0  |0  |0  |0  |0  |0  |0  |0  |0  |0  |0  |0   |0  |0
|0  |1  |0  |0   |0  |0  |0  |0  |0   |0  |0  |0  |0  |0  |0  |0  |0  |0  |1
|0  |0  |0  |0  |0   |1  |0  |0  |0  |0  |0  |0  |0  |0   |0  |1  |0  |0  |0
|0  |0  |0  |0  |0  |0  |0  |0  |0   |0  |0  |0  |1  |0  |0  |0  |0  |0  |0
|0  |0  |0  |0  |0  |0  |0  |0  |0  |0  |0  |0  |0  |0  |0  |0  |0  |0  |0  |
0  |0  |0  |0  |0  |0  |0  |0  |0  |0  |0   |0  |1  |0  |0  |0  |0  |0  |0  |
0  |0  |1  |0  |0   |0   |0  |0  |0   |0  |0  |0  |0  |0  |0  |0  |0   |0  |0
|0  |0  |0  |0  |0  |0  |0  |0  |0  |0  |0  |0  |1  |0  |0  |0  |0   |0  |0
|0  |0  |0  |0  |0  |0  |0  |0  |0  |0  |0  |0  |0  |0  |0  |0  |0  |1  |0  |0  |
0  |0  |0   |0  |1  |0  |0   |0  |0  |0  |0  |0  |0  |0  |0  |0  |0  |0  |1  |0
|0   |0  |0  |0  |0  |0  |0  |0  |0  |0  |0  |0  |0  |0  |0  |0  |0  |1  |1  |0
|0  |0  |0  |0  |0  |0  |0   |0  |0  |0  |0  |0  |0  |0  |0  |0  |0  |0  |0  |0
|0  |0   |
+--------------+---+---+---+---+---+---+---+---+---+---+----+---+---+---+---+
---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+-
--+---+---+---+---+---+---+---+---+---+----+---+---+---+---+---+---+---+---+-
--+---+---+----+---+---+---+---+---+---+---+---+---+---+---+---+---+----+---+
---+---+---+---+----+---+---+---+---+----+---+---+---+---+---+---+---+---+---
+---+---+---+---+---+----+---+---+---+---+---+---+---+----+---+---+---+---+--
-+---+---+---+---+---+---+---+---+---+----+---+---+---+---+---+---+---+---+--
-+----+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+--
-+---+---+---+---+---+---+---+---+---+---+---+----+---+---+---+---+---+---+--
-+---+---+---+---+---+----+----+---+---+----+---+---+---+---+---+---+---+----
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+----
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
---+---+---+---+----+---+---+---+----+---+---+---+---+---+---+---+---+---+---
+---+---+----+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---
+---+---+---+---+---+---+---+---+----+---+---+---+---+---+---+---+---+---+---
+---+---+---+----+
only showing top 20 rows

**** End of Step3 ******
Execution-Date & -Time of Step3: 19.05.2021 16:45:32
```

Step4: Analyze the Data

Count the number of office visits in the "features_period" each patient had.

Definitions for the CPT codes we're using to define office visits can be seen in the following link:

https://www.aafp.org/journals/fpm/blogs/inpractice/entry/coding_office_visits_the_easy_way.html (https://www.aafp.org/journals/fpm/blogs/inpractice/entry/coding_office_visits_the_easy_way.html)

In [9]:
```
#select the visit codes
office_visit_em_codes = [str(i) for i in range(99201, 99206)] + [str(i) for i
in range(99211, 99216)]

#aggregate the visting dates per patient
#show the first 20 results
em_dates = (
    features_period.select(
        substring("patient.reference", 9, 255).alias("patient"),
        explode("item").alias("item")
    ).select(
        "patient",
        col("item.servicedPeriod.start").alias("service_date"),
        col("item.sequence").alias("sequence"),
        explode("item.service.coding").alias("coding")
    ).select(
        "patient",
        "service_date",
        "sequence",
        col("coding.system").alias("system"),
        col("coding.code").alias("code")
    ).where(
        (col("system") == "https://bluebutton.cms.gov/resources/codesystem/hcp
cs") &
        (col("code").isin(office_visit_em_codes))
    ).groupby("patient")
    .agg(countDistinct("service_date").alias("em_dates"))
)
em_dates.show()
```

```
+--------------+--------+
|       patient|em_dates|
+--------------+--------+
|19990000000166|       2|
|19990000000138|       1|
|19990000000151|       1|
|19990000000142|       1|
|19990000000181|       2|
|19990000000141|       1|
|19990000000155|       2|
|19990000000174|       6|
|19990000000161|       1|
|19990000000137|       4|
|19990000000182|       3|
|19990000000157|       2|
|19990000000176|       5|
|19990000000169|       2|
|19990000000170|       1|
|19990000000183|       2|
|19990000000145|       3|
|19990000000152|       2|
|19990000000144|       3|
|19990000000175|       5|
+--------------+--------+
only showing top 20 rows
```

Count the number of unique days each patient has claims on during the `features_period` .

In [10]:
```
from pyspark.sql.functions import countDistinct

num_claims = (
    features_period
    .groupby(substring("patient.reference", 9, 255).alias("patient"))
    .agg(countDistinct("billablePeriod.start").alias("unique_days"))
)
num_claims.show()
```

```
+--------------+-----------+
|       patient|unique_days|
+--------------+-----------+
|19990000000166|          4|
|19990000000138|          1|
|19990000000151|          1|
|19990000000142|          2|
|19990000000181|          2|
|19990000000141|          2|
|19990000000155|          3|
|19990000000174|          8|
|19990000000161|          2|
|19990000000137|          6|
|19990000000182|          7|
|19990000000157|          3|
|19990000000176|          7|
|19990000000169|          3|
|19990000000170|          2|
|19990000000145|          6|
|19990000000183|          3|
|19990000000152|          5|
|19990000000144|          3|
|19990000000175|          7|
+--------------+-----------+
only showing top 20 rows
```

Create the outcome variable we'll be trying to predict.

Sum the `total_paid_amt` across all claims for patients in the `outcome_period`

The parameter "total_paid_amt" will become or target parameter of the multiple Linear Regression (mLR) algorithm

For more details about mLR applications see my DHBW-ML lecture (WS2020), chapter ML05

In [11]:
```
from pyspark.sql.functions import sum

# calculation of the target parameter of mLR
# show first 20 rows of the results
total_paid = (
    outcome_period
    .groupby(substring("patient.reference", 9, 255).alias("patient"))
    .agg(sum("payment.amount.value").alias("total_paid_amt"))
)
total_paid.show()

print("**** End of Step4 ******")
print("Execution-Date & -Time of Step4:",time.strftime("%d.%m.%Y %H:%M:%S"))
```

```
+--------------+--------------+
|       patient|total_paid_amt|
+--------------+--------------+
|19990000000166|           650|
|19990000000138|          2930|
|19990000000151|           270|
|19990000000142|            90|
|19990000000181|           610|
|19990000000141|           250|
|19990000000155|           130|
|19990000000174|          1440|
|19990000000161|         10900|
|19990000000137|           230|
|19990000000182|            20|
|19990000000157|          2530|
|19990000000176|          2060|
|19990000000169|           510|
|19990000000170|           500|
|19990000000145|           360|
|19990000000183|           400|
|19990000000152|           140|
|19990000000144|           340|
|19990000000175|           920|
+--------------+--------------+
only showing top 20 rows

**** End of Step4 ******
Execution-Date & -Time of Step4: 19.05.2021 16:45:52
```

Step5: Train the Model

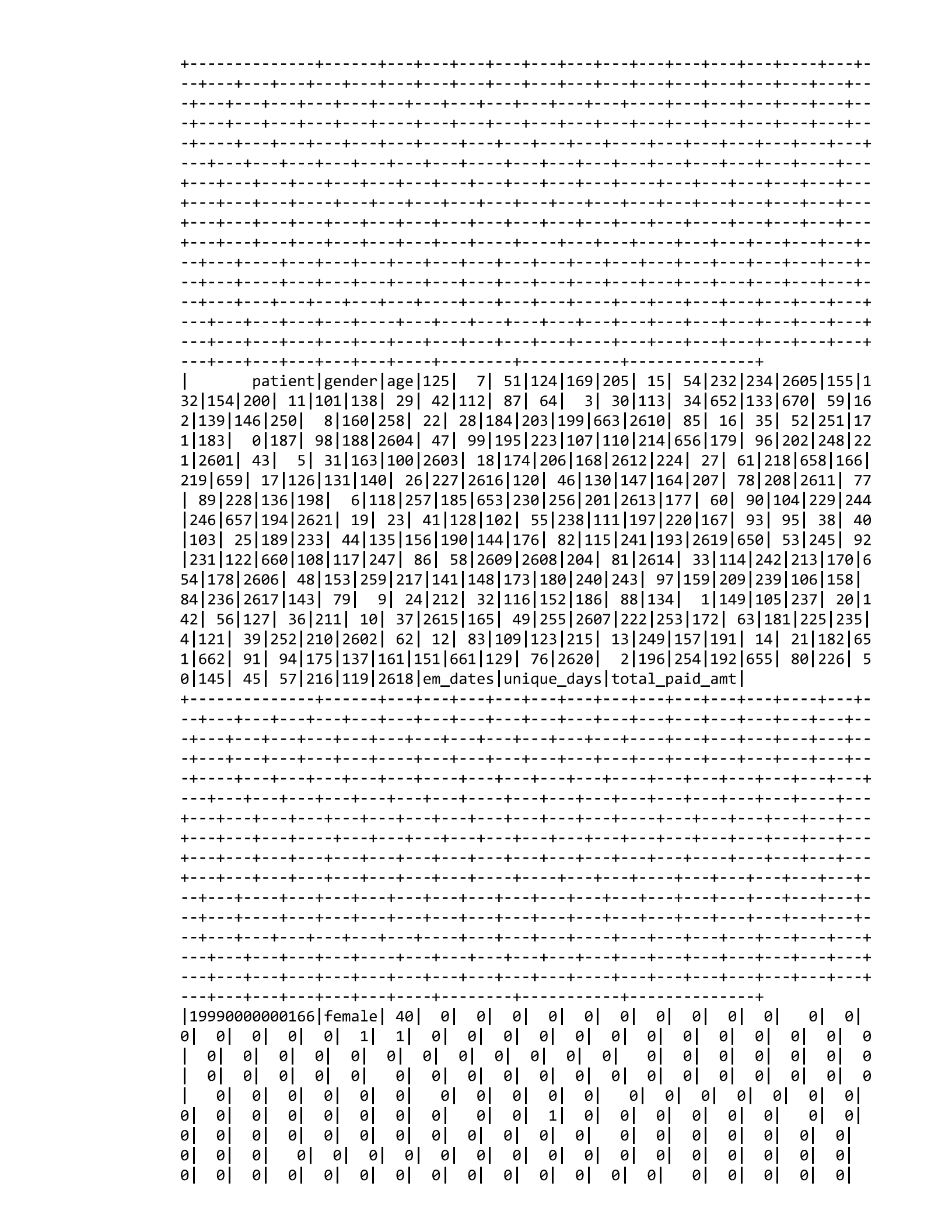
Join everything together into a "one row per prediction" dataset to feed through and train the mLR machine learning algorithm.

We're splitting the dataset into a 80% training-data and 20% validation-/test-data bucket

In [12]:

```
modeling_df = (
    demographics
    .join(ccs_codes, on="patient", how="left_outer")
    .join(em_dates, on="patient", how="left_outer")
    .join(num_claims, on="patient", how="left_outer")
    .fillna(0, subset= ccs_categories + em_dates.columns + num_claims.columns)
    .join(total_paid, on="patient", how="inner")
)
modeling_df.cache()
modeling_df.show()

# splitting of the data in a training-corpus (80%) and test-corpus (20%)
train_df, test_df = modeling_df.randomSplit([0.8, 0.2], seed=72)
```



0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0|
0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0|
0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0|
0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0|
0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0|
0| 0| 0| 1| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0|
0| 0| 0| 0| 0| 0| 2| 4| 650|
|19990000000138|female| 40| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0|
0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0
| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0
| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0
| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 1| 0| 0| 0| 0| 0| 0| 0| 0|
0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0|
0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0|
0| 0| 0| 0| 0| 1| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0|
0| 0| 0| 0| 1| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 1| 0| 0|
0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0|
0| 0| 0| 0| 0| 0| 0| 0| 1| 0| 0| 0| 0| 0| 0| 0| 1| 0| 0|
0| 0| 0| 0| 0| 1| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0|
0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0|
0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0|
0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0|
0| 0| 0| 0| 0| 0| 1| 1| 2930|
|19990000000151|female| 40| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0|
0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0
| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0
| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0
| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0|
0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0|
0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0|
0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0|
0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0|
0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 1| 0| 0| 0| 0| 0| 0| 0| 0|
0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0|
0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0|
0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0|
0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0|
0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0|
0| 0| 0| 0| 0| 0| 1| 1| 270|
|19990000000142| male| 40| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0|
0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0
| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 1| 0
| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0
| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0|
0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0|
0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0|
0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0|
0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0|
0| 0| 0| 0| 0| 0| 1| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0|
0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0|
0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0|
0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0|
0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0|
0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0|
1| 0| 0| 0| 0| 0| 1| 2| 90|
|19990000000181| male| 40| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0|
0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0| 0

```
|   0|  0|  0|  0|  0|  0|  0|   0|  0|  0|  0|  0|   0|  0|  0|  0|  0|  0|  0|
0|  0|  0|  0|  0|  0|  0|  0|  0|   0|  0|  0|  0|  0|  1|  0|  0|  0|   0|  0|
0|  0|  0|  0|  0|  0|  0|  0|  0|  0|  0|  0|  0|   0|  0|  0|  0|  0|  0|  0|
0|  0|  0|   0|  0|  0|  0|  0|  1|  0|  0|  0|  0|  0|  0|  0|  0|  0|  0|  0|
0|  0|  0|  0|  0|  0|  0|  0|  0|  0|  0|  0|  0|  0|   0|  0|  1|  0|  0|
0|  0|  0|  0|  0|  0|  0|  0|   0|   0|  0|  0|   0|  0|  0|  0|  0|  0|  0|
0|   0|  0|  0|  1|  0|  0|  0|  0|  0|  0|  0|  0|  0|  0|  0|  0|  0|  0|
0|   0|  0|  0|  0|  0|  0|  0|  0|  0|  0|  0|  0|  1|  0|  0|  0|  0|  0|
0|  0|  0|  0|  0|  0|   0|  0|  0|  0|   0|  0|  0|  0|  0|  0|  0|  0|  0|
0|  0|  0|  0|   0|  0|  0|  0|  0|  0|  0|  0|  0|  0|  0|  0|  0|  0|  0|
0|  0|  0|  0|  0|  0|  0|  0|  0|  0|   0|  0|  0|  0|  0|  0|  0|  0|  0|
0|  0|  0|  0|  0|   0|       2|          3|          2530|
|19990000000176|female| 40|  0|  0|  0|  0|  0|  0|  0|  0|  0|  0|   0|  0|
0|  0|  1|  0|  0|  0|  0|  0|  0|  0|  0|  0|  0|  0|  0|  0|  0|  0|  0|  0|
|  0|  0|  0|  0|  0|  0|  0|  0|  0|  0|  1|  0|  0|   0|  0|  0|  0|  0|  0|
|  0|  0|  0|  0|  0|  0|   0|  0|  0|  0|  0|  0|  0|  0|  0|  0|  0|  0|  0|
|   0|  0|  0|  0|  0|  0|  0|   0|  0|  0|  0|  0|   0|  0|  0|  0|  0|  0|
0|  0|  0|  1|  0|  0|  0|  0|   0|  1|  0|  0|  0|  0|  0|  0|  0|   0|  0|
0|  0|  0|  0|  0|  0|  0|  0|  0|  0|  0|  0|   0|  0|  0|  1|  0|  0|  0|
0|  0|  0|   0|  0|  0|  0|  0|  0|  0|  0|  0|  0|  0|  0|  1|  0|  0|  0|
0|  0|  0|  0|  1|  0|  0|  0|  0|  0|  0|  0|  0|  0|  0|   0|  0|  0|  0|
0|  0|  0|  0|  0|  0|  1|  0|   0|   0|  0|  0|   0|  0|  0|  0|  0|  0|  0|
0|   0|  0|  0|  0|  0|  0|  0|  0|  0|  0|  0|  0|  0|  0|  0|  0|  0|  0|
0|   0|  0|  0|  0|  0|  0|  0|  0|  0|  0|  0|  0|  0|  0|  0|  0|  0|  0|
0|  0|  0|  1|  0|  0|   0|  0|  0|  0|   0|  0|  1|  0|  0|  0|  0|  0|  0|
0|  0|  0|  0|   0|  0|  0|  0|  0|  0|  0|  0|  0|  0|  0|  0|  0|  0|  0|
0|  0|  1|  0|  0|  0|  0|  0|  0|  0|   0|  0|  0|  0|  0|  0|  0|  0|  0|
0|  0|  0|  0|  0|   0|       5|          7|          2060|
|19990000000169|female| 40|  0|  0|  0|  0|  0|  0|  0|  0|  0|  0|   0|  0|
0|  0|  0|  0|  0|  0|  0|  0|  0|  0|  0|  0|  0|  0|  0|  0|  0|  0|  0|  0|
|  0|  0|  0|  0|  0|  0|  0|  0|  0|  0|  1|  0|  0|   0|  0|  0|  0|  1|  0|
|  0|  0|  0|  0|  0|  0|   0|  0|  0|  0|  0|  0|  0|  0|  0|  0|  0|  0|  0|
|   0|  0|  0|  0|  1|  0|   0|  0|  0|  0|  0|   0|  0|  0|  0|  0|  0|  0|
0|  0|  0|  0|  0|  0|  0|  0|   0|  0|  0|  0|  0|  0|  0|  0|  0|   0|  0|
0|  0|  0|  0|  0|  0|  0|  0|  0|  0|  0|  0|   0|  0|  0|  0|  0|  0|  0|
0|  0|  0|   0|  0|  0|  0|  0|  0|  0|  0|  0|  0|  0|  0|  0|  1|  0|  0|
0|  0|  0|  0|  0|  0|  0|  0|  0|  0|  0|  0|  0|  0|  0|   0|  0|  0|  0|
0|  0|  0|  0|  0|  0|  1|  0|   0|   0|  0|  0|  0|   0|  0|  0|  0|  0|  0|
0|   0|  1|  0|  0|  0|  0|  0|  0|  0|  0|  0|  0|  0|  0|  0|  0|  0|  0|
0|   0|  0|  0|  0|  0|  0|  0|  0|  0|  0|  0|  0|  0|  0|  0|  0|  0|  0|
0|  0|  0|  1|  0|  0|   0|  0|  0|  0|   0|  0|  0|  0|  0|  0|  0|  0|  0|
0|  0|  0|  0|   0|  0|  0|  0|  0|  0|  0|  0|  0|  0|  0|  0|  0|  0|  0|
0|  0|  0|  0|  0|  0|  0|  0|  0|  0|   0|  0|  0|  0|  0|  0|  0|  0|  0|
0|  0|  0|  0|  0|   0|       2|          3|           510|
|19990000000170|female| 40|  1|  0|  0|  0|  0|  0|  0|  0|  0|  0|   0|  0|
0|  0|  0|  0|  0|  0|  0|  0|  0|  0|  0|  0|  0|  0|  0|  0|  0|  0|  0|  0|
|  0|  0|  0|  0|  0|  0|  0|  0|  0|  0|  0|  0|  0|   0|  0|  0|  0|  0|  0|
|  0|  0|  0|  0|  0|  0|   0|  0|  0|  0|  0|  0|  0|  0|  0|  0|  0|  0|  0|
|   0|  0|  0|  0|  0|  0|  0|   0|  0|  0|  0|  0|   0|  0|  0|  0|  0|  0|
0|  0|  0|  0|  0|  0|  0|  0|   0|  0|  0|  0|  0|  0|  0|  0|  0|   0|  0|
0|  0|  0|  0|  0|  0|  0|  0|  0|  0|  0|  0|   0|  0|  0|  0|  0|  0|  0|
0|  0|  0|   0|  0|  0|  0|  0|  0|  0|  0|  0|  0|  0|  0|  0|  0|  0|  0|
0|  0|  0|  0|  0|  0|  0|  0|  0|  0|  0|  0|  0|  0|  0|  0|  0|  0|  0|
0|  0|  0|  0|  0|  0|  1|  0|   0|   0|  0|  0|  0|   0|  0|  0|  0|  0|  0|
0|   0|  0|  0|  0|  0|  0|  0|  0|  0|  0|  0|  0|  0|  0|  0|  0|  0|  0|
0|   0|  0|  0|  0|  0|  0|  0|  0|  0|  0|  0|  0|  0|  0|  0|  0|  0|  0|
0|  0|  0|  0|  0|  0|   0|  0|  0|  0|   0|  0|  0|  0|  0|  0|  0|  0|  0|
```

```
0|  0|  0|  0|  0|  0|  0|  0|  0|  0|  0|  0|  0|   0|  0|  0|  0|  0|  0|  0|  0|
0|  0|  0|   0|  0|  0|  0|  0|  0|  0|  0|  0|  0|  0|  0|  0|  0|  0|  0|  0|  0|
0|  0|  0|  0|  0|  0|  0|  0|  0|  0|  0|  0|  0|  0|  0|  0|   0|  0|  1|  0|  0|
0|  0|  0|  0|  0|  0|  0|  0|  0|   0|   0|  0|  0|  0|   0|  0|  1|  0|  0|  0|  0|
0|   0|  0|  0|  0|  0|  0|  0|  0|  0|  0|  0|  0|  0|  0|  0|  0|  0|  0|  0|  0|
0|   0|  0|  0|  0|  0|  0|  0|  0|  0|  0|  0|  0|  0|  0|  0|  0|  0|  0|  0|  0|
0|  0|  0|  0|  0|  0|  0|   0|  0|  0|  0|   0|  0|  0|  0|  0|  0|  0|  0|  0|  0|
0|  0|  0|  0|   0|  0|  0|  0|  0|  0|  0|  0|  0|  0|  0|  0|  0|  0|  0|  0|  0|
0|  0|  0|  0|  1|  0|  0|  0|  0|  0|   0|  0|  0|  0|  0|  0|  0|  0|  0|  0|  0|
0|  0|  0|  0|  1|   0|       3|          3|           340|
|19990000000175|female| 40|  0|  0|  0|  0|  0|  0|  0|  0|  0|  0|   0|  1|
0|  0|  0|  0|  0|  0|  0|  0|  0|  0|  0|  0|  0|  0|  0|  0|  0|  0|  0|  0|  0
|  0|  1|  0|  0|  0|  0|  0|  0|  0|  0|  1|  0|  0|   0|  0|  0|  0|  0|  0|  0
|  0|  0|  0|  1|  0|   0|  0|  0|  0|  0|  0|  0|  0|  0|  0|  0|  0|  0|  0|  0
|   0|  0|  0|  0|  1|  0|   0|  0|  0|  0|  0|   0|  0|  0|  0|  0|  0|  0|  0|
0|  0|  0|  1|  0|  0|  0|  0|   0|  1|  0|  0|  0|  0|  0|  0|  0|   0|  0|
1|  0|  0|  0|  0|  0|  0|  0|  0|  0|  0|  0|   0|  0|  0|  0|  1|  0|  0|
0|  0|  0|   0|  0|  0|  0|  0|  0|  0|  0|  0|  0|  0|  0|  0|  0|  0|  0|
0|  0|  0|  0|  0|  0|  0|  0|  0|  0|  0|  0|  0|  0|  0|  0|  1|  0|  0|
0|  0|  0|  0|  0|  0|  1|  0|   0|   0|  0|  0|   0|  0|  0|  0|  0|  0|  0|
0|   0|  0|  0|  0|  0|  0|  0|  0|  0|  0|  0|  0|  0|  0|  0|  1|  0|  0|
0|   0|  0|  0|  0|  0|  0|  0|  0|  0|  0|  0|  0|  0|  0|  0|  0|  0|  0|
0|  1|  0|  0|  0|  0|   0|  0|  1|  0|   0|  0|  0|  0|  0|  0|  0|  0|  0|
0|  0|  1|  0|   0|  0|  0|  0|  0|  0|  0|  0|  0|  0|  0|  0|  0|  0|  0|
0|  1|  1|  0|  0|  0|  0|  0|  0|  0|   0|  0|  0|  0|  0|  0|  0|  0|  0|
0|  0|  0|  0|  0|   0|       5|          7|           920|
+--------------+------+---+---+---+---+---+---+---+---+---+---+---+----+---+-
--+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+--
-+---+---+---+---+---+---+---+---+---+---+---+---+----+---+---+---+---+---+--
-+---+---+---+---+---+----+---+---+---+---+---+---+---+---+---+---+---+---+--
-+----+---+---+---+---+---+----+---+---+---+---+----+---+---+---+---+---+---+
---+---+---+---+---+---+---+---+----+---+---+---+---+---+---+---+---+----+---
+---+---+---+---+---+---+---+---+---+---+---+---+----+---+---+---+---+---+---
+---+---+---+----+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+----+---+---+---+---
+---+---+---+---+---+---+---+---+----+----+---+---+----+---+---+---+---+---+-
--+---+----+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+-
--+---+----+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+-
--+---+---+---+---+---+---+----+---+---+---+----+---+---+---+---+---+---+---+
---+---+---+---+---+----+---+---+---+---+---+---+---+---+---+---+---+---+---+
---+---+---+---+---+---+---+---+---+---+---+----+---+---+---+---+---+---+---+
---+---+---+---+---+---+----+--------+-----------+--------------+
only showing top 20 rows
```

Use the `pyspark.ml` module to convert the modeling data above into the appropriate data types and formats to feed through the `LinearRegression` training algorithm.

We use as definition of r2:= Adj.R^2 (see DHBW ML Lecture (WS2020), chapter ML05 for more details.

We know from the lecture that Adj.R^2 can only be defined when n-k-1<> 0 which is same as k<>n-1. Also it is clear that the number of observation-points(n) >= k+2. If this is not the case, you can't calculate a meaningful Regression-Hyperplane HP. We say "without loss od generality" (w.l.o.g): k<= n-2.

Rule: if K>1 choose the regression model where Adj.R^2 is maximum

Theorem (T5.3): "Properties of Adj.R^2"

(i): Adj.R^2 =< R^2 =< 1 "Limitation" (ii): Adj.R^2 can become negative "Negativity"

In our case we see K=2 ("number of variables", also called sometimes "features").

These variables are age and gender.

In [13]:
```
from pyspark.ml import Model, Pipeline
from pyspark.ml.feature import Bucketizer, OneHotEncoder, StringIndexer, Vecto
rAssembler
from pyspark.ml.regression import LinearRegression

age_splits = [
    -float("inf"),
    0,
    10,
    20,
    30,
    40,
    50,
    60,
    70,
    80,
    float("inf")
]

# Define the "feature-vector" with the two components age and gender
# Define the label parameter as "total_paid_amt"
# Call the "Linear Regression" algorithm from the PySpark library

pipeline = Pipeline(stages=[
    StringIndexer(inputCol="gender", outputCol="gender_indexed"),
    OneHotEncoder(inputCol="gender_indexed", outputCol="gender_encoded"),
    Bucketizer(splits=age_splits, inputCol="age", outputCol="age_encoded"),
    VectorAssembler(
        inputCols=["age_encoded", "em_dates", "gender_encoded", "unique_days"]
+ ccs_categories,
        outputCol="features"
    ),
    LinearRegression(featuresCol="features", labelCol="total_paid_amt", fitInt
ercept=True)
])

# calculate the r2 metric for the training data (train_df)
#
trained_model_pipeline = pipeline.fit(train_df)
print(f"Training Data r2={trained_model_pipeline.stages[-1].summary.r2}")

print("**** End of Step5 ******")
print("Execution-Date & -Time of Step5:",time.strftime("%d.%m.%Y %H:%M:%S"))
```

```
Training Data r2=0.9999999999804761
**** End of Step5 ******
Execution-Date & -Time of Step5: 19.05.2021 16:49:07
```

Evaluate the model's performance against the training dataset.

We got almost a perfect score on the training data and a negative score on the testing data. This means we're overfit to our synthetic data and the trained model isn't very useful

In [14]:
```
# import and use the "RegressionEvaluator" function of the pyspark library
from pyspark.ml.evaluation import RegressionEvaluator

# calculate the r2 metric for the test data (test_df)
predictions = trained_model_pipeline.transform(test_df)
evaluator = RegressionEvaluator(
    predictionCol="prediction",
    labelCol="total_paid_amt",
    metricName="r2"
)
print(f"Testing Data r2={evaluator.evaluate(predictions)}")
print("Execution-Date & -Time of Final Result:",time.strftime("%d.%m.%Y %H:%
M:%S"))
print("**** End of Program Executuion *****")
```

```
Testing Data r2=-75.01896166725219
Execution-Date & -Time of Final Result: 19.05.2021 16:49:35
**** End of Program Executuion *****
```

Summary

Congratulations! You have learned to extract, transform and load a set of beneficiary claims data.

Split the data and feed it to a machine learning algorithm to train it for predicting beneficiary claims income.