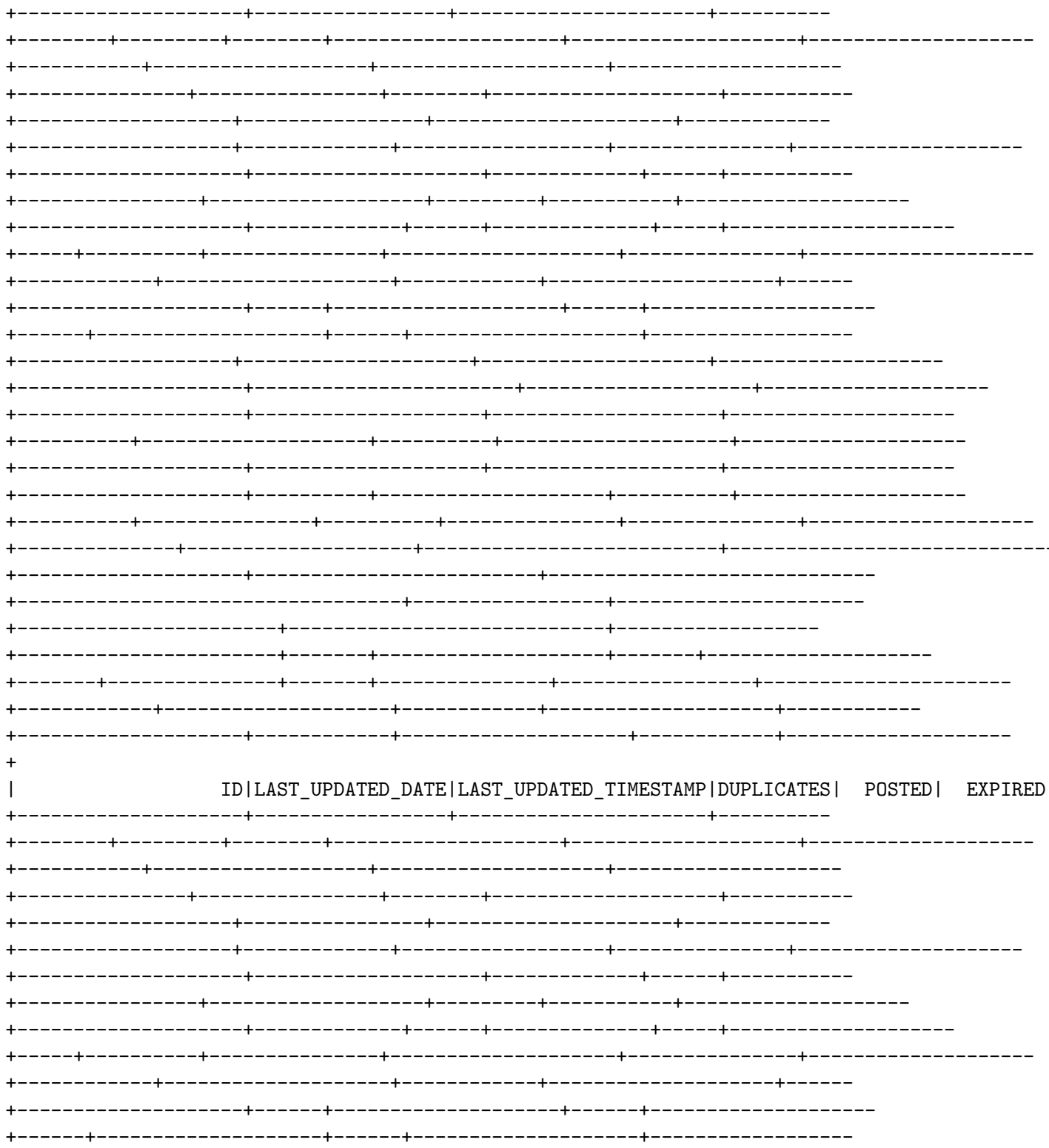
# Assignment 04

JulioVargas

October 7, 2025

```
from pyspark.sql import SparkSession
import pandas as pd
import plotly.express as px
import plotly.io as pio
import numpy as np

np.random.seed(42)

pio.renderers.default = "notebook+notebook_connected+vscode"

# Initialize Spark Session
spark = SparkSession.builder.appName("LightcastData").getOrCreate()

# Load Data
df = spark.read.option("header", "true").option("inferSchema", "true").option("multiLine","t

# Show Schema and Sample Data
print("---This is Diagnostic check, No need to print it in the final doc---")

# df.printSchema() # comment this line when rendering the submission
df.show(5)
print(df.count())
#pd.set_option("display.max_rows", None)
#pd.DataFrame(df.columns, columns=["Column Names"])
```

```
[Stage 283:>                                                        (0 + 1) / 1]

---This is Diagnostic check, No need to print it in the final doc---
```



```
+------------------+------------------+------------------+------------------
+------------------+--------------------+------------------+------------------
+------------------+------------------+------------------+------------------
+---------+------------------+---------+------------------+------------------
+------------------+------------------+------------------+------------------
+------------------+---------+------------------+---------+------------------
+---------+--------------+---------+--------------+--------------+------------------
+-------------+------------------+------------------------+--------------------------
+------------------+------------------------+---------------------------
+--------------------------------+----------------+--------------------
+----------------------+---------------------------+-----------------
+----------------------+-------+--------------------+-------+--------------------
+-------+--------------+-------+--------------+----------------+----------------------
+-----------+------------------+-----------+------------------+-----------
+------------------+-----------+------------------+-----------+------------------
+
|1f57d95acf4dc67ed...|         9/6/2024|  2024-09-06 20:32:...|         0|6/2/2024| 6/8/2024
May-2024\n\nEn...|       6/8/2024|               6|  894731|          Murphy USA| Murphy USA
time (> 32 h...|                   2|                   2|        false|  NULL|          0|
2051.01|Business Intellig...|15-2051.01|Business Intellig...|[\n  "45.0601",\n...|[\n  "Econ
0000|Computer and Math...|   15-2000|Mathematical Scie...|   15-2050|Data Scientists|   15-
2051|Data Scientists|             23|Information Techn...|        231010|Business Intellig..
0000|Computer and Math...|15-2000|Mathematical Scie...|15-2050|Data Scientists|15-
2051|Data Scientists|        [\n  7\n]|  [\n  "Artificial ...|          44|        Retail Tra
|0cb072af26757b6c4...|         8/2/2024|  2024-08-02 17:08:...|         0|6/2/2024| 8/1/2024
time (> 32 h...|                   3|                   3|        false|  NULL|          1|
Watervill...|   23|     Maine|          23011|        Kennebec, ME|          23011|        K
Watervill...|       12300|Augusta-Watervill...|    56|Administrative an...|   561|Administrat
2051.01|Business Intellig...|15-2051.01|Business Intellig...|                  []|
0000|Computer and Math...|   15-2000|Mathematical Scie...|   15-2050|Data Scientists|   15-
2051|Data Scientists|             23|Information Techn...|        231010|Business Intellig..
0000|Computer and Math...|15-2000|Mathematical Scie...|15-2050|Data Scientists|15-
2051|Data Scientists|             NULL|                  NULL|          56|Administrative an
|85318b12b3331fa49...|         9/6/2024|  2024-09-06 20:32:...|         1|6/2/2024| 7/7/2024
time (> 32 h...|                   5|                NULL|        false|  NULL|          0|
Fort Worth...|   48|     Texas|          48113|          Dallas, TX|          48113|
Fort Worth...|       19100|Dallas-Fort Worth...|    52|Finance and Insur...|   524|Insurance
2051.01|Business Intellig...|15-2051.01|Business Intellig...|                  []|
0000|Computer and Math...|   15-2000|Mathematical Scie...|   15-2050|Data Scientists|   15-
2051|Data Scientists|             23|Information Techn...|        231113|Data / Data Minin..
0000|Computer and Math...|15-2000|Mathematical Scie...|15-2050|Data Scientists|15-
2051|Data Scientists|             NULL|                  NULL|          52|Finance and Insur
|1b5c3941e54a1889e...|         9/6/2024|  2024-09-06 20:32:...|         1|6/2/2024|7/20/2024
```

```
time (> 32 h...|                   3|                NULL|        false|  NULL|          0|
Mesa-Chan...|    4|   Arizona|           4013|        Maricopa, AZ|           4013|        Ma
Mesa-Chan...|       38060|Phoenix-Mesa-Chan...|    52|Finance and Insur...|   522|Credit Inte
2051.01|Business Intellig...|15-2051.01|Business Intellig...|                  []|
0000|Computer and Math...|   15-2000|Mathematical Scie...|   15-2050|Data Scientists|   15-
2051|Data Scientists|             23|Information Techn...|        231113|Data / Data Minin..
0000|Computer and Math...|15-2000|Mathematical Scie...|15-2050|Data Scientists|15-
2051|Data Scientists|        [\n  6\n]|  [\n  "Data Privac...|          52|Finance and Insur
|cb5ca25f02bdf25c1...|        6/19/2024|   2024-06-19 07:00:00|         0|6/2/2024|6/17/2024
time / full-...|                NULL|                NULL|        false| 92500|          0|
2051.01|Business Intellig...|15-2051.01|Business Intellig...|                  []|
0000|Computer and Math...|   15-2000|Mathematical Scie...|   15-2050|Data Scientists|   15-
2051|Data Scientists|             23|Information Techn...|        231010|Business Intellig..
0000|Computer and Math...|15-2000|Mathematical Scie...|15-2050|Data Scientists|15-
2051|Data Scientists|             NULL|                  NULL|          99|Unclassified Indu
+--------------------+-----------------+----------------------+----------
+--------+---------+--------+--------------------+--------------------+--------------------
+----------+------------------+--------------------+--------------------
+--------------+---------------+--------+--------------------+----------
+------------------+---------------+--------------------+-------------
+------------------+-------------+------------------+--------------+--------------------
+--------------------+--------------------+-------------+------+----------
+----------------+------------------+---------+-----------+--------------------
+--------------------+-------------+------+-------------+-----+--------------------
+-----+----------+--------------+--------------------+--------------+--------------------
+------------+--------------------+------------+--------------------+------
+--------------------+------+--------------------+------+--------------------
+------+--------------------+------+--------------------+------------------
+------------------+--------------------+--------------------+--------------------
+--------------------+----------------------+--------------------+--------------------
+--------------------+--------------------+--------------------+--------------------
+---------+------------------+---------+--------------------+--------------------
+--------------------+--------------------+--------------------+--------------------
+--------------------+---------+--------------------+---------+--------------------
+---------+---------------+---------+---------------+---------------+--------------------
+--------------+-------------------+--------------------------+----------------------------
+--------------------+--------------------------+------------------------------
+----------------------------------+-----------------+----------------------
+----------------------+----------------------------+------------------
+----------------------+-------+--------------------+-------+--------------------
+-------+--------------+-------+--------------+----------------+----------------------
+------------+--------------------+------------+--------------------+------------
+--------------------+------------+--------------------+------------+--------------------
```

```
+
only showing top 5 rows

[Stage 285:>                                                        (0 + 1) / 1]

72498
```

Feature Engineering is a crucial step in preparing your data for machine learning. In this lab, we will focus on the following tasks:

1. Drop rows with missing values in the target variable and key features.
2. By now you are already familiar with the code and the data. Based on your understanding please choose any 3 (my code output has 10) variables as:
    1. three continuous variables and, `MIN_YEARS_EXPERIENCE` (total 4, use your best judgment!)
    2. two categorical.
    3. Your dependent variable (y) is `SALARY`.
3. Convert categorical variables into numerical representations using StringIndexer and OneHotEncoder.
4. Assemble features into a single vector using VectorAssembler.
5. Split the data into training and testing sets.
6. You can use pipeline to do the above steps in one go.
7. Create a new column `MIN_YEARS_EXPERIENCE_SQ` by squaring the `MIN_YEARS_EXPERIENCE` column.
8. Assemble the polynomial features into a new vector column `features_poly` using VectorAssembler.
9. Show the final structure of the DataFrame with the new features.

```
+------+--------------------+--------------------+--------+------------------
+-------------+----------+---------------+----------------------+------------------
+------------------+
|SALARY|MIN_YEARS_EXPERIENCE|MAX_YEARS_EXPERIENCE|DURATION|COMPANY_IS_STAFFING|IS_INTERNSHIP
+------+--------------------+--------------------+--------+------------------
+-------------+----------+---------------+----------------------+------------------
+------------------+
|NULL  |2                   |2                   |6       |false              |false
time (> 32 hours)|Bachelor's degree  |NULL              |
|NULL  |3                   |3                   |NULL    |true               |false
time (> 32 hours)|No Education Listed|NULL              |
```

```
|NULL  |5                    |NULL                |35      |false              |false
time (> 32 hours)|Bachelor's degree  |NULL              |
|NULL  |3                    |NULL                |48      |false              |false
time (> 32 hours)|No Education Listed|NULL              |
|92500 |NULL                 |NULL                |15      |false              |false
time / full-time |No Education Listed|NULL              |
+------+---------------------+--------------------+--------+-------------------
+------------+---------+---------------+---------------------+------------------
+------------------+
only showing top 5 rows
```

```python
from pyspark.sql.functions import col, sum as spark_sum, when, trim, length
import hvplot.pandas  # enables hvplot on pandas

missing_df = df_eda.select([
    spark_sum(
        when(col(c).isNull() | (length(trim(col(c))) == 0), 1)
        .otherwise(0)
    ).alias(c)
    for c in df_eda.columns
])

#print(missing_df.show())

#to table with T Transpose
missing_pd = missing_df.toPandas().T.reset_index()
#put names to columns
missing_pd.columns = ["column", "missing_count"]

total_rows = df_eda.count()
missing_pd["missing_pct"] = 100 * missing_pd["missing_count"] / total_rows

# hvplot.bar ; line; scatter; (hist); (box); area; (heatmap); (hexbin); points
missing_pd.sort_values("missing_pct", ascending=False).hvplot.bar(
    x="column", y="missing_pct", rot=90,
    title="Percentage of Missing Values by Column",
    height=600, width=900,
    ylabel="Missing Percentage (%)", xlabel="Features"
).opts(xrotation=45)
```

```
Unable to display output for mime type(s): text/html
```

```
Unable to display output for mime type(s): application/javascript, application/vnd.holoviews_

Unable to display output for mime type(s): application/javascript, application/vnd.holoviews_

Unable to display output for mime type(s): application/vnd.holoviews_exec.v0+json, text/html

[Stage 289:>                                                        (0 + 1) / 1]

:Bars   [column]   (missing_pct)
```

```python
# For REMOTE_TYPE_NAME replace Remote with Remote, [None] with Undefined,
# Not Remote with On Premise, Hybrid Remote with Hybrid, and Null with On Premise
## data frame (eda) exploratory data analysis

df_eda = df_eda.withColumn(
    "REMOTE_TYPE_NAME",
    when(col("REMOTE_TYPE_NAME") == "Remote", "Remote")
    .when(col("REMOTE_TYPE_NAME") == "[None]", "Undefined")
    .when(col("REMOTE_TYPE_NAME") == "Not Remote", "On-Premise")
    .when(col("REMOTE_TYPE_NAME") == "Hybrid Remote", "Hybrid")
    .when(col("REMOTE_TYPE_NAME").isNull(), "On-Premise")
    .otherwise(col("REMOTE_TYPE_NAME"))
)

# df_eda.createOrReplaceTempView("df_eda")
categorical_cols = [
    "REMOTE_TYPE_NAME"
]

for colname in categorical_cols:
    print(f"\n---- {colname} ----")
    df_eda.select(colname).distinct().show(10, truncate=False)
```

```
---- REMOTE_TYPE_NAME ----

[Stage 295:>                                                        (0 + 1) / 1]
```

```
+----------------+
|REMOTE_TYPE_NAME|
+----------------+
|Remote          |
|On-Premise      |
|Hybrid          |
|Undefined       |
+----------------+
```

```python
# ---- EMPLOYMENT_TYPE_NAME ----

# +------------------------+
# |EMPLOYMENT_TYPE_NAME    |
# +------------------------+
# |Part-time / full-time   |
# |Part-time (â‰¤ 32 hours)|
# |Full-time (> 32 hours)  |
# |NULL                    |
# +------------------------+

df_eda = df_eda.withColumn(
    "EMPLOYMENT_TYPE_NAME",
    when(col("EMPLOYMENT_TYPE_NAME") == "Part-time / full-time", "Flexible")
    .when(col("EMPLOYMENT_TYPE_NAME") == "Part-time (â‰¤ 32 hours)", "Parttime")
    .when(col("EMPLOYMENT_TYPE_NAME") == "Full-time (> 32 hours)", "Fulltime")
    .when(col("EMPLOYMENT_TYPE_NAME").isNull(), "Fulltime")
    .otherwise(col("EMPLOYMENT_TYPE_NAME"))
)

# df_eda.createOrReplaceTempView("df_eda")
categorical_cols = [
    "EMPLOYMENT_TYPE_NAME"
]

for colname in categorical_cols:
    print(f"\n---- {colname} ----")
    df_eda.select(colname).distinct().show(10, truncate=False)
```

```
---- EMPLOYMENT_TYPE_NAME ----
```

```
[Stage 298:>                                                        (0 + 1) / 1]

+--------------------+
|EMPLOYMENT_TYPE_NAME|
+--------------------+
|Flexible            |
|Fulltime            |
|Parttime            |
+--------------------+
```

```python
# replace COMPANY_IS_STAFFING NULL with false, and IS_INTERNSHIP NULL with false
df_eda = df_eda.withColumn(
    "COMPANY_IS_STAFFING",
    when(col("COMPANY_IS_STAFFING").isNull(), False)
    .otherwise(col("COMPANY_IS_STAFFING"))
)

df_eda = df_eda.withColumn(
    "IS_INTERNSHIP",
    when(col("IS_INTERNSHIP").isNull(), False)
    .otherwise(col("IS_INTERNSHIP"))
)

# df_eda.createOrReplaceTempView("df_eda")
categorical_cols = [
    "COMPANY_IS_STAFFING", "IS_INTERNSHIP"
]

for colname in categorical_cols:
    print(f"\n---- {colname} ----")
    df_eda.select(colname).distinct().show(10, truncate=False)
```

```
---- COMPANY_IS_STAFFING ----

[Stage 301:>                                                        (0 + 1) / 1]

+-------------------+
|COMPANY_IS_STAFFING|
```

```
+-------------------+
|true               |
|false              |
+-------------------+


---- IS_INTERNSHIP ----

[Stage 304:>                                                        (0 + 1) / 1]

+-------------+
|IS_INTERNSHIP|
+-------------+
|true         |
|false        |
+-------------+
```

```python
import pandas as pd

# sample subset of data only 1% of the data
df_sample = df_eda.sample(fraction=0.01, seed=42).toPandas()

#print(df_eda.count())  #72498
#print(len(df_sample))  #790

# create new DataFrame where each cell missing (True) or not (False)
missing_mask = df_sample.isnull()

# Melt into long-form  | 4 columns: index, column, is_missing
missing_long = (
    missing_mask.reset_index()
    .melt(id_vars="index", var_name="column", value_name="is_missing")
)

# Convert boolean to int
missing_long["is_missing"] = missing_long["is_missing"].astype(int)

print(missing_long)
```

```
# Plot heatmap
missing_long.hvplot.heatmap(
    x="column", y="index", C="is_missing",
    cmap="Reds", colorbar=False,
    width=900, height=700,
    title="Heatmap of Missing Values (Sample)"
).opts(xrotation=45)
```

```
[Stage 307:>                                                        (0 + 1) / 1]
```

| | index | column | is_missing |
|---|---|---|---|
| 0 | 0 | SALARY | 1 |
| 1 | 1 | SALARY | 0 |
| 2 | 2 | SALARY | 1 |
| 3 | 3 | SALARY | 1 |
| 4 | 4 | SALARY | 1 |
| ... | ... | ... | ... |
| 8685 | 785 | MAX_EDULEVELS_NAME | 0 |
| 8686 | 786 | MAX_EDULEVELS_NAME | 0 |
| 8687 | 787 | MAX_EDULEVELS_NAME | 1 |
| 8688 | 788 | MAX_EDULEVELS_NAME | 1 |
| 8689 | 789 | MAX_EDULEVELS_NAME | 1 |

```
[8690 rows x 3 columns]
```

```
:HeatMap   [column,index]   (is_missing)
```

```
from pyspark.sql.functions import countDistinct

#show number of unique values per column
df_eda.select([
    countDistinct(c).alias(c + "_nunique")
    for c in df_eda.columns
]).show(truncate=False)
```

```
[Stage 308:>                                                        (0 + 1) / 1]
```

```
+--------------+---------------------------+---------------------------
+----------------+---------------------------+---------------------+------------------
+------------------------+---------------------------+-------------------------
+--------------------------+
```

```
|SALARY_nunique|MIN_YEARS_EXPERIENCE_nunique|MAX_YEARS_EXPERIENCE_nunique|DURATION_nunique|C
+--------------+----------------------------+----------------------------
+---------------+----------------------------+--------------------+-----------------
+-----------------------+----------------------------+-------------------------
+-------------------------+
|6052          |16                          |15                          |60             |2
+--------------+----------------------------+----------------------------
+---------------+----------------------------+--------------------+-----------------
+-----------------------+----------------------------+-------------------------
+-------------------------+
```

```
categorical_cols = [
    "STATE_NAME", "REMOTE_TYPE_NAME", "EMPLOYMENT_TYPE_NAME",
    "MIN_EDULEVELS_NAME", "COMPANY_IS_STAFFING", "IS_INTERNSHIP"
]

for colname in categorical_cols:
    print(f"\n---- {colname} ----")
    df_eda.select(colname).distinct().show(10, truncate=False)
```

```
---- STATE_NAME ----

[Stage 314:>                                                        (0 + 1) / 1]

+------------+
|STATE_NAME  |
+------------+
|Utah        |
|Hawaii      |
|Minnesota   |
|Ohio        |
|Arkansas    |
|Oregon      |
|Texas       |
|North Dakota|
|Pennsylvania|
|Connecticut |
+------------+
```

```
only showing top 10 rows

---- REMOTE_TYPE_NAME ----

[Stage 317:>                                                        (0 + 1) / 1]

+----------------+
|REMOTE_TYPE_NAME|
+----------------+
|Remote          |
|On-Premise      |
|Hybrid          |
|Undefined       |
+----------------+


---- EMPLOYMENT_TYPE_NAME ----

[Stage 320:>                                                        (0 + 1) / 1]

+--------------------+
|EMPLOYMENT_TYPE_NAME|
+--------------------+
|Flexible            |
|Fulltime            |
|Parttime            |
+--------------------+


---- MIN_EDULEVELS_NAME ----

[Stage 323:>                                                        (0 + 1) / 1]

+---------------------------+
|MIN_EDULEVELS_NAME         |
+---------------------------+
|Bachelor's degree          |
|Ph.D. or professional degree|
|High school or GED         |
|Master's degree            |
|No Education Listed        |
```

```
|Associate degree            |
|NULL                        |
+----------------------------+


---- COMPANY_IS_STAFFING ----

[Stage 326:>                                                        (0 + 1) / 1]

+-------------------+
|COMPANY_IS_STAFFING|
+-------------------+
|true               |
|false              |
+-------------------+


---- IS_INTERNSHIP ----

[Stage 329:>                                                        (0 + 1) / 1]

+-------------+
|IS_INTERNSHIP|
+-------------+
|true         |
|false        |
+-------------+
```

```
# Calculate median of the Duration Column

median_duration = df_eda.approxQuantile("DURATION", [0.5], 0.01)[0]

# Check for missing values in Duration column and replace null with median

df_eda = df_eda.withColumn(
    "DURATION",
    when(col("DURATION").isNull(), median_duration)
    .otherwise(col("DURATION"))
) # Assuming median duration is 30 days
```

```
[Stage 332:>                                                        (0 + 1) / 1]
```

```
import pandas as pd

# sample subset of data
df_sample = df_eda.sample(fraction=0.10, seed=42).toPandas()

# Boolean mask (True if missing)
missing_mask = df_sample.isnull()

# Melt into long-form
missing_long = (
    missing_mask.reset_index()
    .melt(id_vars="index", var_name="column", value_name="is_missing")
)

# Convert boolean to int
missing_long["is_missing"] = missing_long["is_missing"].astype(int)

# Plot heatmap
missing_long.hvplot.heatmap(
    x="column", y="index", C="is_missing",
    cmap="Reds", colorbar=False,
    width=900, height=700,
    title="Heatmap of Missing Values (Sample)"
).opts(xrotation=45)
```

```
[Stage 333:>                                                        (0 + 1) / 1]
```

```
:HeatMap   [column,index]   (is_missing)
```

```
df_eda.show(5, truncate=False)
```

```
+------+--------------------+--------------------+--------+------------------
+-------------+----------+---------------+--------------------+------------------
+------------------+
|SALARY|MIN_YEARS_EXPERIENCE|MAX_YEARS_EXPERIENCE|DURATION|COMPANY_IS_STAFFING|IS_INTERNSHIP
+------+--------------------+--------------------+--------+------------------
+-------------+----------+---------------+--------------------+------------------
+------------------+
|NULL  |2                   |2                   |6.0     |false              |false
```

```
|NULL  |3                  |3                  |18.0    |true               |false
|NULL  |5                  |NULL               |35.0    |false              |false
|NULL  |3                  |NULL               |48.0    |false              |false
|92500 |NULL               |NULL               |15.0    |false              |false
+------+-------------------+-------------------+--------+-------------------
+-------------+----------+---------------+--------------------+------------------
+------------------+
only showing top 5 rows
```

```
# Drop rows with NA values in relevant columns
df_feature_engg = df_eda.dropna(subset=[
    "SALARY", "MIN_YEARS_EXPERIENCE", "MAX_YEARS_EXPERIENCE","STATE_NAME",
     "EMPLOYMENT_TYPE_NAME", "REMOTE_TYPE_NAME","MIN_EDULEVELS_NAME",
    "DURATION", "IS_INTERNSHIP", "COMPANY_IS_STAFFING"
])

# Categorical columns
categorical_cols = ["STATE_NAME","MIN_EDULEVELS_NAME","EMPLOYMENT_TYPE_NAME", "REMOTE_TYPE_NA

# Index and One-Hot Encode
indexers = [StringIndexer(inputCol=col, outputCol=f"{col}_idx", handleInvalid='skip') for col
encoders = [OneHotEncoder(inputCol=f"{col}_idx", outputCol=f"{col}_vec") for col in categoric

pipeline = Pipeline(stages=indexers)
indexed_df = pipeline.fit(df_feature_engg).transform(df_feature_engg)
indexed_df.select("EMPLOYMENT_TYPE_NAME","EMPLOYMENT_TYPE_NAME_idx","REMOTE_TYPE_NAME","REMOT

pipeline = Pipeline(stages=indexers + encoders)
encoded_df = pipeline.fit(df_feature_engg).transform(df_feature_engg)
encoded_df.show()
```

```
[Stage 335:>                                                        (0 + 1) / 1]

+--------------------+-----------------------+----------------+--------------------
+
|EMPLOYMENT_TYPE_NAME|EMPLOYMENT_TYPE_NAME_idx|REMOTE_TYPE_NAME|REMOTE_TYPE_NAME_idx|
+--------------------+-----------------------+----------------+--------------------
+
|           Fulltime|                    0.0|       Undefined|                0.0|
|           Fulltime|                    0.0|       Undefined|                0.0|
|           Fulltime|                    0.0|          Remote|                1.0|
|           Fulltime|                    0.0|       Undefined|                0.0|
```

```
|            Fulltime|                    0.0|          Remote|                 1.0|
|            Fulltime|                    0.0|       Undefined|                 0.0|
|            Fulltime|                    0.0|          Remote|                 1.0|
|            Fulltime|                    0.0|          Remote|                 1.0|
|            Fulltime|                    0.0|       Undefined|                 0.0|
|            Fulltime|                    0.0|       Undefined|                 0.0|
|            Fulltime|                    0.0|       Undefined|                 0.0|
|            Fulltime|                    0.0|       Undefined|                 0.0|
|            Fulltime|                    0.0|      On-Premise|                 3.0|
|            Fulltime|                    0.0|       Undefined|                 0.0|
|            Fulltime|                    0.0|       Undefined|                 0.0|
|            Fulltime|                    0.0|       Undefined|                 0.0|
|            Fulltime|                    0.0|          Remote|                 1.0|
|            Flexible|                    2.0|       Undefined|                 0.0|
|            Fulltime|                    0.0|       Undefined|                 0.0|
|            Fulltime|                    0.0|       Undefined|                 0.0|
+--------------------+------------------------+----------------+--------------------
+
only showing top 20 rows

[Stage 360:>                                                        (0 + 1) / 1]

+------+--------------------+--------------------+--------+--------------------
+-------------+--------------+---------------+--------------------+--------------------
+--------------------+-------------+----------------------+------------------------
+--------------------+--------------+----------------------+------------------------
+--------------------+
|SALARY|MIN_YEARS_EXPERIENCE|MAX_YEARS_EXPERIENCE|DURATION|COMPANY_IS_STAFFING|IS_INTERNSHIP
+------+--------------------+--------------------+--------+--------------------
+-------------+--------------+---------------+--------------------+--------------------
+--------------------+-------------+----------------------+------------------------
+--------------------+--------------+----------------------+------------------------
+--------------------+
| 92962|                   2|                   2|    18.0|              false|        false
| 75026|                   2|                   2|    18.0|               true|        false
| 60923|                   1|                   1|    18.0|              false|        false
|131100|                   2|                   2|    11.0|              false|        false
|136950|                   3|                   3|    18.0|              false|        false
|122500|                   5|                   5|    18.0|              false|        false
|136950|                   3|                   3|    28.0|              false|        false
|136950|                   3|                   3|    28.0|              false|        false
| 55120|                   2|                   2|    18.0|              false|        false
```

```
|104000|                   3|                   3|     8.0|              false|        false
|145319|                   4|                   4|    18.0|              false|        false
| 80000|                   3|                   3|    37.0|              false|        false
|102500|                   3|                   3|    28.0|              false|        false
Premise|            Fulltime| Bachelor's degree|                NULL|          27.0|
| 86117|                   2|                   2|    14.0|              false|        false
| 72800|                   3|                   3|    15.0|               true|        false
|162300|                   6|                   6|    18.0|              false|        false
|121500|                   3|                   3|    18.0|              false|        false
|117500|                   3|                   3|    14.0|              false|        false
|102000|                   3|                   3|    25.0|              false|        false
|142300|                   6|                   6|    18.0|              false|        false
+------+--------------------+--------------------+--------+-------------------
+-------------+--------------+----------------+--------------------+------------------
+--------------------+-------------+----------------------+-----------------------
+--------------------+--------------+----------------------+-----------------------
+--------------------+
only showing top 20 rows
```

```python
# Assemble base features (for GLR and Random Forest)
assembler = VectorAssembler(
    inputCols=[
        "MIN_YEARS_EXPERIENCE", "DURATION",
        "IS_INTERNSHIP", "COMPANY_IS_STAFFING"
    ] + [f"{col}_vec" for col in categorical_cols],
    outputCol="features"
)

# Build pipeline and transform
pipeline = Pipeline(stages=indexers + encoders + [assembler])
data = pipeline.fit(df_feature_engg).transform(df_feature_engg)

data.show(5, truncate=False)

# Create squared term for Polynomial Regression
data = data.withColumn("MIN_YEARS_EXPERIENCE_SQ", pow(col("MIN_YEARS_EXPERIENCE"), 2))

# Assemble polynomial features
assembler_poly = VectorAssembler(
    inputCols=[
        "MIN_YEARS_EXPERIENCE", "MIN_YEARS_EXPERIENCE_SQ",
        "DURATION", "IS_INTERNSHIP", "COMPANY_IS_STAFFING"
```

```
    ] + [f"{col}_vec" for col in categorical_cols],
    outputCol="features_poly"

)

data=assembler_poly.transform(data)

#show final structure
data.select("SALARY", "features", "features_poly").show(5, truncate=False)
```

```
[Stage 385:>                                                        (0 + 1) / 1]

+------+--------------------+--------------------+--------+--------------------
+------------+-----------+---------------+--------------------+-----------------
+---------------------------+-------------+----------------------+-------------------------
+--------------------+--------------+----------------------+------------------------
+--------------------+-----------------------------------------------------------
+
|SALARY|MIN_YEARS_EXPERIENCE|MAX_YEARS_EXPERIENCE|DURATION|COMPANY_IS_STAFFING|IS_INTERNSHIP
+------+--------------------+--------------------+--------+--------------------
+------------+-----------+---------------+--------------------+-----------------
+---------------------------+-------------+----------------------+-------------------------
+--------------------+--------------+----------------------+------------------------
+--------------------+-----------------------------------------------------------
+
|92962 |2                   |2                   |18.0    |false               |false
|75026 |2                   |2                   |18.0    |true                |false
|60923 |1                   |1                   |18.0    |false               |false
|131100|2                   |2                   |11.0    |false               |false
|136950|3                   |3                   |18.0    |false               |false
+------+--------------------+--------------------+--------+--------------------
+------------+-----------+---------------+--------------------+-----------------
+---------------------------+-------------+----------------------+-------------------------
+--------------------+--------------+----------------------+------------------------
+--------------------+-----------------------------------------------------------
+
only showing top 5 rows
+------+-----------------------------------------------------------+---------------------------
+
|SALARY|features                                                   |features_poly
+------+-----------------------------------------------------------+---------------------------
+
```

```
|92962 |(64,[0,1,5,54,59,61],[2.0,18.0,1.0,1.0,1.0,1.0])       |(65,[0,1,2,6,55,60,62],[2.0,4
|75026 |(64,[0,1,3,25,54,59,61],[2.0,18.0,1.0,1.0,1.0,1.0,1.0])|(65,[0,1,2,4,26,55,60,62],[2
|60923 |(64,[0,1,5,54,59,62],[1.0,18.0,1.0,1.0,1.0,1.0])       |(65,[0,1,2,6,55,60,63],[1.0,1
|131100|(64,[0,1,19,54,59,61],[2.0,11.0,1.0,1.0,1.0,1.0])      |(65,[0,1,2,20,55,60,62],[2.0
|136950|(64,[0,1,44,54,59,62],[3.0,18.0,1.0,1.0,1.0,1.0])      |(65,[0,1,2,45,55,60,63],[3.0
+------+--------------------------------------------------------+----------------------------
+
only showing top 5 rows

[Stage 411:>                                                        (0 + 1) / 1]

(3756, 22)

[Stage 414:>                                                        (0 + 1) / 1]

(3060, 22)

[Stage 417:>                                                        (0 + 1) / 1]

(696, 22)
```

```python
from pyspark.ml.regression import LinearRegression

# Initialize  Regression model
# Basic Linear Regression model
mr = LinearRegression(featuresCol="features", labelCol="SALARY")

# Polynomial Regression using squared term features
#mr = LinearRegression(featuresCol="features_poly", labelCol="SALARY")

# Generalized Linear Regression (supports distributions like Gaussian, Poisson)
    # mr = GeneralizedLinearRegression(
    #     featuresCol="features", labelCol="SALARY",
    #     family="gaussian", link="identity"
    # )


# Train the model
```

```
mr_model = mr.fit(regression_train)

# Evaluate on test data
test_results = mr_model.evaluate(regression_test)

# Print metrics
print("RMSE:", test_results.rootMeanSquaredError)
print("R2:", test_results.r2)
```

```
25/10/07 07:38:44 WARN Instrumentation: [3e82e2e4] regParam is zero, which might cause numeri
[Stage 420:>                                                        (0 + 1) / 1]

RMSE: 29114.307979774672
R2: 0.34792446541830324
```

```
coeffs = mr_model.coefficients
intercept = mr_model.intercept

print("Intercept:", intercept)
print("Coefficients:", coeffs)
```

```
Intercept: 131328.3004796009
Coefficients: [8523.062910159928,-98.93546907813675,2782.413073078266,-
1078.1337256635686,12347.650329505026,12099.991428639481,5508.201281626465,4103.394476179377
755.6703653785958,9973.18649877864,1314.7076955620487,-1299.0660250961937,2093.9361992914087
5124.810223960465,4945.904023459982,-293.75547075283794,434.7029433494427,4697.278843397262,
11920.218306321509,-2098.935017913266,28.958169601522584,10422.439688057706,-
1534.9827239260164,3298.1875476982027,2561.1538943029723,8352.834248147985,-
3844.4745256701326,-1105.821356233869,1840.7175686222201,487.1782311291216,5803.378236307139
3041.9102995939397,10591.372903022557,-577.2862695106961,-3619.269866311597,5432.888820586008
520.5179852949578,3209.602981992066,-3999.8839871360237,-3502.2852404853415,8178.297251076545
54961.00350028223,-49607.85277992999,-84497.42795385023,-79343.56469541852,-
40206.932513201114,-17034.987287641532,-20945.521186969952,13599.615064274507,19866.84231589
```