Assignment 04

Norah Jones

November 21, 2024

# 1. Data Preparartion

## 1.1 Load data and review

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","true").option("escape", "\"").csv("data/lightcast\_job\_postings.csv")  
  
# 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)

[Stage 95:> (0 + 1) / 1]

+--------------------+-----------------+----------------------+----------+--------+---------+--------+--------------------+--------------------+--------------------+-----------+-------------------+--------------------+--------------------+---------------+----------------+--------+--------------------+-----------+-------------------+----------------+---------------------+-------------+-------------------+-------------+------------------+---------------+--------------------+--------------------+--------------------+-------------+------+-----------+----------------+-------------------+---------+-----------+--------------------+--------------------+-------------+------+--------------+-----+--------------------+-----+----------+---------------+--------------------+---------------+--------------------+------------+--------------------+------------+--------------------+------+--------------------+------+--------------------+------+--------------------+------+--------------------+------+--------------------+------------------+-------------------+--------------------+--------------------+--------------------+--------------------+-----------------------+--------------------+--------------------+--------------------+--------------------+--------------------+--------------------+----------+--------------------+----------+--------------------+--------------------+--------------------+--------------------+--------------------+--------------------+--------------------+----------+--------------------+----------+--------------------+----------+---------------+----------+---------------+---------------+--------------------+--------------+--------------------+--------------------------+-------------------------------+--------------------+-------------------------+-----------------------------+----------------------------------+-----------------+----------------------+-----------------------+----------------------------+------------------+-----------------------+-------+--------------------+-------+--------------------+-------+---------------+-------+---------------+-----------------+----------------------+------------+--------------------+------------+--------------------+------------+--------------------+------------+--------------------+------------+--------------------+  
| ID|LAST\_UPDATED\_DATE|LAST\_UPDATED\_TIMESTAMP|DUPLICATES| POSTED| EXPIRED|DURATION| SOURCE\_TYPES| SOURCES| URL|ACTIVE\_URLS|ACTIVE\_SOURCES\_INFO| TITLE\_RAW| BODY|MODELED\_EXPIRED|MODELED\_DURATION| COMPANY| COMPANY\_NAME|COMPANY\_RAW|COMPANY\_IS\_STAFFING|EDUCATION\_LEVELS|EDUCATION\_LEVELS\_NAME|MIN\_EDULEVELS| MIN\_EDULEVELS\_NAME|MAX\_EDULEVELS|MAX\_EDULEVELS\_NAME|EMPLOYMENT\_TYPE|EMPLOYMENT\_TYPE\_NAME|MIN\_YEARS\_EXPERIENCE|MAX\_YEARS\_EXPERIENCE|IS\_INTERNSHIP|SALARY|REMOTE\_TYPE|REMOTE\_TYPE\_NAME|ORIGINAL\_PAY\_PERIOD|SALARY\_TO|SALARY\_FROM| LOCATION| CITY| CITY\_NAME|COUNTY| COUNTY\_NAME| MSA| MSA\_NAME|STATE|STATE\_NAME|COUNTY\_OUTGOING|COUNTY\_NAME\_OUTGOING|COUNTY\_INCOMING|COUNTY\_NAME\_INCOMING|MSA\_OUTGOING| MSA\_NAME\_OUTGOING|MSA\_INCOMING| MSA\_NAME\_INCOMING|NAICS2| NAICS2\_NAME|NAICS3| NAICS3\_NAME|NAICS4| NAICS4\_NAME|NAICS5| NAICS5\_NAME|NAICS6| NAICS6\_NAME| TITLE| TITLE\_NAME| TITLE\_CLEAN| SKILLS| SKILLS\_NAME| SPECIALIZED\_SKILLS|SPECIALIZED\_SKILLS\_NAME| CERTIFICATIONS| CERTIFICATIONS\_NAME| COMMON\_SKILLS| COMMON\_SKILLS\_NAME| SOFTWARE\_SKILLS|SOFTWARE\_SKILLS\_NAME| ONET| ONET\_NAME| ONET\_2019| ONET\_2019\_NAME| CIP6| CIP6\_NAME| CIP4| CIP4\_NAME| CIP2| CIP2\_NAME|SOC\_2021\_2| SOC\_2021\_2\_NAME|SOC\_2021\_3| SOC\_2021\_3\_NAME|SOC\_2021\_4|SOC\_2021\_4\_NAME|SOC\_2021\_5|SOC\_2021\_5\_NAME|LOT\_CAREER\_AREA|LOT\_CAREER\_AREA\_NAME|LOT\_OCCUPATION| LOT\_OCCUPATION\_NAME|LOT\_SPECIALIZED\_OCCUPATION|LOT\_SPECIALIZED\_OCCUPATION\_NAME|LOT\_OCCUPATION\_GROUP|LOT\_OCCUPATION\_GROUP\_NAME|LOT\_V6\_SPECIALIZED\_OCCUPATION|LOT\_V6\_SPECIALIZED\_OCCUPATION\_NAME|LOT\_V6\_OCCUPATION|LOT\_V6\_OCCUPATION\_NAME|LOT\_V6\_OCCUPATION\_GROUP|LOT\_V6\_OCCUPATION\_GROUP\_NAME|LOT\_V6\_CAREER\_AREA|LOT\_V6\_CAREER\_AREA\_NAME| SOC\_2| SOC\_2\_NAME| SOC\_3| SOC\_3\_NAME| SOC\_4| SOC\_4\_NAME| SOC\_5| SOC\_5\_NAME|LIGHTCAST\_SECTORS|LIGHTCAST\_SECTORS\_NAME|NAICS\_2022\_2| NAICS\_2022\_2\_NAME|NAICS\_2022\_3| NAICS\_2022\_3\_NAME|NAICS\_2022\_4| NAICS\_2022\_4\_NAME|NAICS\_2022\_5| NAICS\_2022\_5\_NAME|NAICS\_2022\_6| NAICS\_2022\_6\_NAME|  
+--------------------+-----------------+----------------------+----------+--------+---------+--------+--------------------+--------------------+--------------------+-----------+-------------------+--------------------+--------------------+---------------+----------------+--------+--------------------+-----------+-------------------+----------------+---------------------+-------------+-------------------+-------------+------------------+---------------+--------------------+--------------------+--------------------+-------------+------+-----------+----------------+-------------------+---------+-----------+--------------------+--------------------+-------------+------+--------------+-----+--------------------+-----+----------+---------------+--------------------+---------------+--------------------+------------+--------------------+------------+--------------------+------+--------------------+------+--------------------+------+--------------------+------+--------------------+------+--------------------+------------------+-------------------+--------------------+--------------------+--------------------+--------------------+-----------------------+--------------------+--------------------+--------------------+--------------------+--------------------+--------------------+----------+--------------------+----------+--------------------+--------------------+--------------------+--------------------+--------------------+--------------------+--------------------+----------+--------------------+----------+--------------------+----------+---------------+----------+---------------+---------------+--------------------+--------------+--------------------+--------------------------+-------------------------------+--------------------+-------------------------+-----------------------------+----------------------------------+-----------------+----------------------+-----------------------+----------------------------+------------------+-----------------------+-------+--------------------+-------+--------------------+-------+---------------+-------+---------------+-----------------+----------------------+------------+--------------------+------------+--------------------+------------+--------------------+------------+--------------------+------------+--------------------+  
|1f57d95acf4dc67ed...| 9/6/2024| 2024-09-06 20:32:...| 0|6/2/2024| 6/8/2024| 6| [\n "Company"\n]|[\n "brassring.c...|[\n "https://sjo...| []| NULL|Enterprise Analys...|31-May-2024\n\nEn...| 6/8/2024| 6| 894731| Murphy USA| Murphy USA| false| [\n 2\n]| [\n "Bachelor's ...| 2| Bachelor's degree| NULL| NULL| 1|Full-time (> 32 h...| 2| 2| false| NULL| 0| [None]| NULL| NULL| NULL|{\n "lat": 33.20...|RWwgRG9yYWRvLCBBUg==|El Dorado, AR| 5139| Union, AR|20980| El Dorado, AR| 5| Arkansas| 5139| Union, AR| 5139| Union, AR| 20980| El Dorado, AR| 20980| El Dorado, AR| 44| Retail Trade| 441|Motor Vehicle and...| 4413|Automotive Parts,...| 44133|Automotive Parts ...|441330|Automotive Parts ...|ET29C073C03D1F86B4|Enterprise Analysts|enterprise analys...|[\n "KS126DB6T06...|[\n "Merchandisi...|[\n "KS126DB6T06...| [\n "Merchandisi...| []| []|[\n "KS126706DPF...|[\n "Mathematics...|[\n "KS440W865GC...|[\n "SQL (Progra...|15-2051.01|Business Intellig...|15-2051.01|Business Intellig...|[\n "45.0601",\n...|[\n "Economics, ...|[\n "45.06",\n ...|[\n "Economics",...|[\n "45",\n "27...|[\n "Social Scie...| 15-0000|Computer and Math...| 15-2000|Mathematical Scie...| 15-2050|Data Scientists| 15-2051|Data Scientists| 23|Information Techn...| 231010|Business Intellig...| 23101011| General ERP Analy...| 2310| Business Intellig...| 23101011| General ERP Analy...| 231010| Business Intellig...| 2310| Business Intellig...| 23| Information Techn...|15-0000|Computer and Math...|15-2000|Mathematical Scie...|15-2050|Data Scientists|15-2051|Data Scientists| [\n 7\n]| [\n "Artificial ...| 44| Retail Trade| 441|Motor Vehicle and...| 4413|Automotive Parts,...| 44133|Automotive Parts ...| 441330|Automotive Parts ...|  
|0cb072af26757b6c4...| 8/2/2024| 2024-08-02 17:08:...| 0|6/2/2024| 8/1/2024| NULL| [\n "Job Board"\n]| [\n "maine.gov"\n]|[\n "https://job...| []| NULL|Oracle Consultant...|Oracle Consultant...| 8/1/2024| NULL| 133098|Smx Corporation L...| SMX| true| [\n 99\n]| [\n "No Educatio...| 99|No Education Listed| NULL| NULL| 1|Full-time (> 32 h...| 3| 3| false| NULL| 1| Remote| NULL| NULL| NULL|{\n "lat": 44.31...| QXVndXN0YSwgTUU=| Augusta, ME| 23011| Kennebec, ME|12300|Augusta-Watervill...| 23| Maine| 23011| Kennebec, ME| 23011| Kennebec, ME| 12300|Augusta-Watervill...| 12300|Augusta-Watervill...| 56|Administrative an...| 561|Administrative an...| 5613| Employment Services| 56132|Temporary Help Se...|561320|Temporary Help Se...|ET21DDA63780A7DC09| Oracle Consultants|oracle consultant...|[\n "KS122626T55...|[\n "Procurement...|[\n "KS122626T55...| [\n "Procurement...| []| []| []| []|[\n "BGSBF3F508F...|[\n "Oracle Busi...|15-2051.01|Business Intellig...|15-2051.01|Business Intellig...| []| []| []| []| []| []| 15-0000|Computer and Math...| 15-2000|Mathematical Scie...| 15-2050|Data Scientists| 15-2051|Data Scientists| 23|Information Techn...| 231010|Business Intellig...| 23101012| Oracle Consultant...| 2310| Business Intellig...| 23101012| Oracle Consultant...| 231010| Business Intellig...| 2310| Business Intellig...| 23| Information Techn...|15-0000|Computer and Math...|15-2000|Mathematical Scie...|15-2050|Data Scientists|15-2051|Data Scientists| NULL| NULL| 56|Administrative an...| 561|Administrative an...| 5613| Employment Services| 56132|Temporary Help Se...| 561320|Temporary Help Se...|  
|85318b12b3331fa49...| 9/6/2024| 2024-09-06 20:32:...| 1|6/2/2024| 7/7/2024| 35| [\n "Job Board"\n]|[\n "dejobs.org"\n]|[\n "https://dej...| []| NULL| Data Analyst|Taking care of pe...| 6/10/2024| 8|39063746| Sedgwick| Sedgwick| false| [\n 2\n]| [\n "Bachelor's ...| 2| Bachelor's degree| NULL| NULL| 1|Full-time (> 32 h...| 5| NULL| false| NULL| 0| [None]| NULL| NULL| NULL|{\n "lat": 32.77...| RGFsbGFzLCBUWA==| Dallas, TX| 48113| Dallas, TX|19100|Dallas-Fort Worth...| 48| Texas| 48113| Dallas, TX| 48113| Dallas, TX| 19100|Dallas-Fort Worth...| 19100|Dallas-Fort Worth...| 52|Finance and Insur...| 524|Insurance Carrier...| 5242|Agencies, Brokera...| 52429|Other Insurance R...|524291| Claims Adjusting|ET3037E0C947A02404| Data Analysts| data analyst|[\n "KS1218W78FG...|[\n "Management"...|[\n "ESF3939CE1F...| [\n "Exception R...|[\n "KS683TN76T7...|[\n "Security Cl...|[\n "KS1218W78FG...|[\n "Management"...|[\n "KS126HY6YLT...|[\n "Microsoft O...|15-2051.01|Business Intellig...|15-2051.01|Business Intellig...| []| []| []| []| []| []| 15-0000|Computer and Math...| 15-2000|Mathematical Scie...| 15-2050|Data Scientists| 15-2051|Data Scientists| 23|Information Techn...| 231113|Data / Data Minin...| 23111310| Data Analyst| 2311| Data Analysis and...| 23111310| Data Analyst| 231113| Data / Data Minin...| 2311| Data Analysis and...| 23| Information Techn...|15-0000|Computer and Math...|15-2000|Mathematical Scie...|15-2050|Data Scientists|15-2051|Data Scientists| NULL| NULL| 52|Finance and Insur...| 524|Insurance Carrier...| 5242|Agencies, Brokera...| 52429|Other Insurance R...| 524291| Claims Adjusting|  
|1b5c3941e54a1889e...| 9/6/2024| 2024-09-06 20:32:...| 1|6/2/2024|7/20/2024| 48| [\n "Job Board"\n]|[\n "disabledper...|[\n "https://www...| []| NULL|Sr. Lead Data Mgm...|About this role:\...| 6/12/2024| 10|37615159| Wells Fargo|Wells Fargo| false| [\n 99\n]| [\n "No Educatio...| 99|No Education Listed| NULL| NULL| 1|Full-time (> 32 h...| 3| NULL| false| NULL| 0| [None]| NULL| NULL| NULL|{\n "lat": 33.44...| UGhvZW5peCwgQVo=| Phoenix, AZ| 4013| Maricopa, AZ|38060|Phoenix-Mesa-Chan...| 4| Arizona| 4013| Maricopa, AZ| 4013| Maricopa, AZ| 38060|Phoenix-Mesa-Chan...| 38060|Phoenix-Mesa-Chan...| 52|Finance and Insur...| 522|Credit Intermedia...| 5221|Depository Credit...| 52211| Commercial Banking|522110| Commercial Banking|ET2114E0404BA30075|Management Analysts|sr lead data mgmt...|[\n "KS123QX62QY...|[\n "Exit Strate...|[\n "KS123QX62QY...| [\n "Exit Strate...| []| []|[\n "KS7G6NP6R6L...|[\n "Reliability...|[\n "KS4409D76NW...|[\n "SAS (Softwa...|15-2051.01|Business Intellig...|15-2051.01|Business Intellig...| []| []| []| []| []| []| 15-0000|Computer and Math...| 15-2000|Mathematical Scie...| 15-2050|Data Scientists| 15-2051|Data Scientists| 23|Information Techn...| 231113|Data / Data Minin...| 23111310| Data Analyst| 2311| Data Analysis and...| 23111310| Data Analyst| 231113| Data / Data Minin...| 2311| Data Analysis and...| 23| Information Techn...|15-0000|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...| 522|Credit Intermedia...| 5221|Depository Credit...| 52211| Commercial Banking| 522110| Commercial Banking|  
|cb5ca25f02bdf25c1...| 6/19/2024| 2024-06-19 07:00:00| 0|6/2/2024|6/17/2024| 15|[\n "FreeJobBoar...|[\n "craigslist....|[\n "https://mod...| []| NULL|Comisiones de $10...|Comisiones de $10...| 6/17/2024| 15| 0| Unclassified| LH/GM| false| [\n 99\n]| [\n "No Educatio...| 99|No Education Listed| NULL| NULL| 3|Part-time / full-...| NULL| NULL| false| 92500| 0| [None]| year| 150000| 35000|{\n "lat": 37.63...| TW9kZXN0bywgQ0E=| Modesto, CA| 6099|Stanislaus, CA|33700| Modesto, CA| 6|California| 6099| Stanislaus, CA| 6099| Stanislaus, CA| 33700| Modesto, CA| 33700| Modesto, CA| 99|Unclassified Indu...| 999|Unclassified Indu...| 9999|Unclassified Indu...| 99999|Unclassified Indu...|999999|Unclassified Indu...|ET0000000000000000| Unclassified|comisiones de por...| []| []| []| []| []| []| []| []| []| []|15-2051.01|Business Intellig...|15-2051.01|Business Intellig...| []| []| []| []| []| []| 15-0000|Computer and Math...| 15-2000|Mathematical Scie...| 15-2050|Data Scientists| 15-2051|Data Scientists| 23|Information Techn...| 231010|Business Intellig...| 23101012| Oracle Consultant...| 2310| Business Intellig...| 23101012| Oracle Consultant...| 231010| Business Intellig...| 2310| Business Intellig...| 23| Information Techn...|15-0000|Computer and Math...|15-2000|Mathematical Scie...|15-2050|Data Scientists|15-2051|Data Scientists| NULL| NULL| 99|Unclassified Indu...| 999|Unclassified Indu...| 9999|Unclassified Indu...| 99999|Unclassified Indu...| 999999|Unclassified Indu...|  
+--------------------+-----------------+----------------------+----------+--------+---------+--------+--------------------+--------------------+--------------------+-----------+-------------------+--------------------+--------------------+---------------+----------------+--------+--------------------+-----------+-------------------+----------------+---------------------+-------------+-------------------+-------------+------------------+---------------+--------------------+--------------------+--------------------+-------------+------+-----------+----------------+-------------------+---------+-----------+--------------------+--------------------+-------------+------+--------------+-----+--------------------+-----+----------+---------------+--------------------+---------------+--------------------+------------+--------------------+------------+--------------------+------+--------------------+------+--------------------+------+--------------------+------+--------------------+------+--------------------+------------------+-------------------+--------------------+--------------------+--------------------+--------------------+-----------------------+--------------------+--------------------+--------------------+--------------------+--------------------+--------------------+----------+--------------------+----------+--------------------+--------------------+--------------------+--------------------+--------------------+--------------------+--------------------+----------+--------------------+----------+--------------------+----------+---------------+----------+---------------+---------------+--------------------+--------------+--------------------+--------------------------+-------------------------------+--------------------+-------------------------+-----------------------------+----------------------------------+-----------------+----------------------+-----------------------+----------------------------+------------------+-----------------------+-------+--------------------+-------+--------------------+-------+---------------+-------+---------------+-----------------+----------------------+------------+--------------------+------------+--------------------+------------+--------------------+------------+--------------------+------------+--------------------+  
only showing top 5 rows

### 1.1.1 Pick salary as target variable, state name, NAICS2\_NAME name, remote type name, employment type name, city name, education levels name, min years experience , duration will be indepent variables for the analysis.

### 1.1.2 For min years experience and duration fill na with 0.

df = df.select("SALARY","STATE\_NAME","NAICS2\_NAME", "EDUCATION\_LEVELS\_NAME", "MIN\_YEARS\_EXPERIENCE", "DURATION")  
#df.show()  
df = df.na.fill({"MIN\_YEARS\_EXPERIENCE": 0, "DURATION":0})  
#df.show()

### 1.1.3 Visualize the nas to understand the magnitude. Over 50% of salary is na, remove those values.

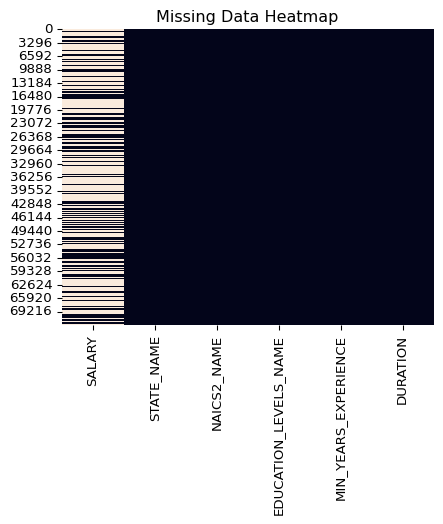
df\_pd = df.toPandas()  
#df\_pd.head(5)  
(df\_pd.isna().sum() / len(df\_pd)) \* 100

[Stage 97:> (0 + 1) / 1]

SALARY 57.505035  
STATE\_NAME 0.060691  
NAICS2\_NAME 0.060691  
EDUCATION\_LEVELS\_NAME 0.060691  
MIN\_YEARS\_EXPERIENCE 0.000000  
DURATION 0.000000  
dtype: float64

### 1.1.4 Using seaborn review a heat map of NA values. The independant variables are whole but salary is missing data.

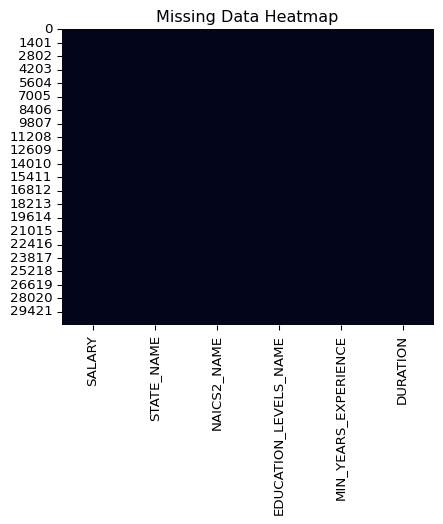
sns.heatmap(df\_pd.isna(), cbar=False)  
plt.title("Missing Data Heatmap")  
plt.show()



### 1.1.5 Drop all records where salary is NA.

df = df.na.drop(subset=["SALARY"])  
df\_pd = df.toPandas()  
  
sns.heatmap(df\_pd.isna(), cbar=False)  
plt.title("Missing Data Heatmap")  
plt.show()

[Stage 98:> (0 + 1) / 1]



### 1.1.6 The data is cleaned and is ready for modeling.

(df\_pd.isna().sum() / len(df\_pd)) \* 100

SALARY 0.0  
STATE\_NAME 0.0  
NAICS2\_NAME 0.0  
EDUCATION\_LEVELS\_NAME 0.0  
MIN\_YEARS\_EXPERIENCE 0.0  
DURATION 0.0  
dtype: float64

# 2. Feature Engineering

## 2.1 First take the input variables and split into numeric and non numeric goups. State name, education levels, NAICS\_NAME are all categoric variables. Min years experience and duration are numeric.

# Suppose you have these columns  
categorical\_cols = ["STATE\_NAME", "EDUCATION\_LEVELS\_NAME","NAICS2\_NAME"]  
numeric\_cols = ["MIN\_YEARS\_EXPERIENCE", "DURATION"]

### 2.1.1 For the categorical columns, assign index values to each column and then one hot encode the columns as a vector.

indexers = [StringIndexer(inputCol=col, outputCol=f"{col}\_indexed") for col in categorical\_cols]  
encoders = [OneHotEncoder(inputCols=[f"{col}\_indexed"], outputCols=[f"{col}\_encoded"]) for col in categorical\_cols]

### 2.1.2 Next compile the one hot encoded columns with the numeric columns in a vector to be used in feature modeling.

assembler\_inputs = [f"{col}\_encoded" for col in categorical\_cols] + numeric\_cols  
assembler = VectorAssembler(inputCols=assembler\_inputs, outputCol="features")

### 2.1.3 Store these data preparation steps as a pipeline for further use

pipeline = Pipeline(stages=indexers + encoders + [assembler])

### 2.1.4 For polynomial square min years experience

from pyspark.sql.functions import col, pow  
  
df\_poly = df.withColumn("MIN\_YEARS\_EXPERIENCE\_SQ", pow(col("MIN\_YEARS\_EXPERIENCE"), 2))

### 2.1.5 Assemble vector using min years and min years experience for polynomial features.

from pyspark.ml.feature import VectorAssembler  
  
assembler\_poly = VectorAssembler(  
 inputCols=["MIN\_YEARS\_EXPERIENCE", "MIN\_YEARS\_EXPERIENCE\_SQ"],  
 outputCol="features\_poly"  
)  
  
df\_poly = assembler\_poly.transform(df\_poly)

df\_poly.printSchema()

root  
 |-- SALARY: integer (nullable = true)  
 |-- STATE\_NAME: string (nullable = true)  
 |-- NAICS2\_NAME: string (nullable = true)  
 |-- EDUCATION\_LEVELS\_NAME: string (nullable = true)  
 |-- MIN\_YEARS\_EXPERIENCE: integer (nullable = false)  
 |-- DURATION: integer (nullable = false)  
 |-- MIN\_YEARS\_EXPERIENCE\_SQ: double (nullable = false)  
 |-- features\_poly: vector (nullable = true)

#df\_poly.show()

### 2.1.6 Now split the data for training and testing in a 70/30% split.

train\_df, test\_df = df.randomSplit([0.7, 0.3], seed=42)

### 2.1.7 Use the previusly created pipeline the prepare the training and test data for use.

pipeline\_model = pipeline.fit(train\_df)  
train\_ready = pipeline\_model.transform(train\_df)  
test\_ready = pipeline\_model.transform(test\_df)

[Stage 99:> (0 + 1) / 1] [Stage 105:> (0 + 1) / 1] [Stage 111:> (0 + 1) / 1]

### 2.1.8 Parse target variable and features vector for modeling.

train\_ready = train\_ready["SALARY","features"]  
#train\_ready.show()  
  
test\_ready = test\_ready["SALARY","features"]

### 2.1.9 Confirm schema for linear regression model.

train\_ready.printSchema()

root  
 |-- SALARY: integer (nullable = true)  
 |-- features: vector (nullable = true)

# 3. Modeling

## 3.1 Build and fit model

### 3.1.1 Penalalize large coefficients and keep all features. Allow for intercept and standardize the data accross variables.

lrm = LinearRegression(  
 featuresCol="features",  
 labelCol="SALARY",  
 predictionCol="prediction",  
 maxIter=100,  
 regParam=0.1,  
 elasticNetParam=0.0,  
 fitIntercept=True,  
 standardization=True,  
)  
  
modelLR = lrm.fit(train\_ready)

[Stage 117:> (0 + 1) / 1] [Stage 118:> (0 + 1) / 1]

## 3.2 Linear Regression Model Summary on Train and test data

summary = modelLR.summary

### 3.2.1 Issue existed with spillover so control with se, tvals, pvals. For features, loop through for only the number of coefficients so to avoid spillover.

### 3.2.2 Linear regression model performed well. All features had statistically significant p values, moderately large t values, reasonably equivalent coefficient magnitudes with respect to salary, which is reflected in the stndard error being significantly smaller than the respective coefficient.

se = summary.coefficientStandardErrors[1:]  
tvals = summary.tValues[1:]  
pvals = summary.pValues[1:]  
  
coef\_df = pd.DataFrame({  
 "Feature": [f"feature\_{i+1}" for i in range(len(modelLR.coefficients))],  
 "Coefficient": modelLR.coefficients.toArray(),  
 "StdError": se,  
 "tValue": tvals,  
 "pValue": pvals  
})  
  
coef\_df.head(100)

|  | Feature | Coefficient | StdError | tValue | pValue |
| --- | --- | --- | --- | --- | --- |
| 0 | feature\_1 | 27160.771987 | 5586.524673 | 3.197485 | 1.388322e-03 |
| 1 | feature\_2 | 17862.826140 | 5604.708248 | 3.970201 | 7.204534e-05 |
| 2 | feature\_3 | 22251.815915 | 5628.321573 | 2.900919 | 3.724464e-03 |
| 3 | feature\_4 | 16327.306545 | 5639.799141 | 3.565083 | 3.645294e-04 |
| 4 | feature\_5 | 20106.349447 | 5650.125035 | 3.825928 | 1.306483e-04 |
| ... | ... | ... | ... | ... | ... |
| 93 | feature\_94 | -13950.827368 | 9974.182772 | -0.376460 | 7.065783e-01 |
| 94 | feature\_95 | -3754.885150 | 10060.013381 | 0.278668 | 7.805023e-01 |
| 95 | feature\_96 | 2803.403880 | 73.807420 | 74.588198 | 0.000000e+00 |
| 96 | feature\_97 | 5505.162463 | 15.692949 | -6.188578 | 6.180532e-10 |
| 97 | feature\_98 | -97.117036 | 36792.595281 | 0.164635 | 8.692328e-01 |

## 3.3 Now validate and run the model using the test data

testmodelLR = lrm.fit(test\_ready)

[Stage 119:> (0 + 1) / 1] [Stage 120:> (0 + 1) / 1]

testsummary = testmodelLR.summary  
  
t\_values = summary.tValues  
p\_values = summary.pValues  
  
stats\_df = pd.DataFrame({  
 "t\_value": t\_values,  
 "p\_value": p\_values  
})  
  
desc\_stats = stats\_df.describe()  
print(desc\_stats)

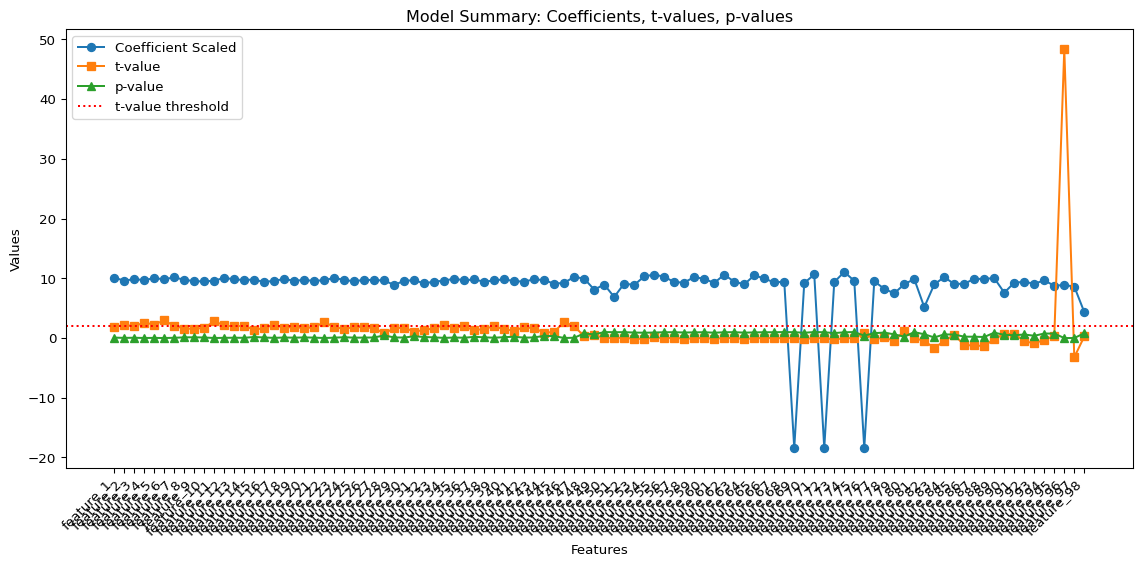
t\_value p\_value  
count 99.000000 99.000000  
mean 2.496936 0.176394  
std 7.503150 0.265126  
min -6.188578 0.000000  
25% 0.898590 0.003787  
50% 2.039629 0.036848  
75% 2.887619 0.279912  
max 74.588198 0.981592

### 3.3.1 Issue existed with spillover so control with se, tvals, pvals. For features, loop through for only the number of coefficients so to avoid spillover.

### 3.3.2 The linear regression model performed okay on the test data. Accross the 99 features, the average p value was not significant, however the median value was significant at appx. .037 - the 75th percentile is not significant but the 25th is. THere were a few features that produced extreme results in all of the t value (~-6/74), p value (0/.98), and coefficient (a few spikes). R squared states ~35% of variation in salary can be explained by the features in the data. On average predictions are off by ~$35,834.

se = testsummary.coefficientStandardErrors[1:]  
tvals = testsummary.tValues[1:]  
pvals = testsummary.pValues[1:]  
  
testcoef\_df = pd.DataFrame({  
 "Feature": [f"feature\_{i+1}" for i in range(len(testmodelLR.coefficients))],  
 "Coefficient": testmodelLR.coefficients.toArray(),  
 "StdError": se,  
 "tValue": tvals,  
 "pValue": pvals  
})  
  
print("R2:", testsummary.r2)  
print("RMSE:", testsummary.rootMeanSquaredError)  
  
#coef\_df.head(100)  
# Scale coefficients to t-values range  
testcoef\_df["Coefficient\_log"] = np.log(np.abs(testcoef\_df["Coefficient"]) + 1e-8)   
  
import matplotlib.pyplot as plt  
  
plt.figure(figsize=(12,6))  
  
# x-axis = features  
x = testcoef\_df["Feature"]  
  
# plot each statistic as a line  
plt.plot(x, testcoef\_df["Coefficient\_log"], marker='o', label="Coefficient Scaled")  
plt.plot(x, testcoef\_df["tValue"], marker='s', label="t-value")  
plt.plot(x, testcoef\_df["pValue"], marker='^', label="p-value")  
  
plt.axhline(y=2, color='red', linestyle=':', label='t-value threshold')  
  
plt.xticks(rotation=45, ha='right') # rotate feature names for readability  
plt.xlabel("Features")  
plt.ylabel("Values")  
plt.title("Model Summary: Coefficients, t-values, p-values")  
plt.legend()  
plt.tight\_layout()  
plt.show()

R2: 0.35636023407578243  
RMSE: 35834.08039301742



# 4. Polynomial Regression

from pyspark.ml.feature import PolynomialExpansion, VectorAssembler

## 4.1 Combine numeric and non numeric columns.

assembler\_poly = VectorAssembler(  
 inputCols=["MIN\_YEARS\_EXPERIENCE", "DURATION"],  
 outputCol="numeric\_features"  
)  
  
#expand to polynomial  
poly\_expansion = PolynomialExpansion(  
 degree=2,   
 inputCol="numeric\_features",   
 outputCol="poly\_features"  
)

### 4.1.1 Use the pipeline created earlier to prepare the polynomial features.

assembler\_final = VectorAssembler(  
 inputCols=["poly\_features"] + [f"{col}\_encoded" for col in categorical\_cols],  
 outputCol="features"  
)  
  
poly\_pipeline = Pipeline(stages=indexers + encoders + [assembler\_poly, poly\_expansion, assembler\_final])

### 4.1.2 Fit model and apply same parameters as the linear regression. ~36.8% of variation in salary can be explained by the feature variables. And on average the prediction was off by $36,025.

poly\_model = poly\_pipeline.fit(train\_df)  
train\_poly = poly\_model.transform(train\_df)  
test\_poly = poly\_model.transform(test\_df)  
  
lrm\_poly = LinearRegression(  
 featuresCol="features",  
 labelCol="SALARY",  
 predictionCol="prediction",  
 maxIter=100,  
 regParam=0.1,  
 elasticNetParam=0.0  
)  
  
model\_poly = lrm\_poly.fit(train\_poly)  
summary\_poly = model\_poly.summary  
  
print("R2:", summary\_poly.r2)  
print("RMSE:", summary\_poly.rootMeanSquaredError)

[Stage 121:> (0 + 1) / 1] [Stage 127:> (0 + 1) / 1] [Stage 133:> (0 + 1) / 1] [Stage 139:> (0 + 1) / 1] [Stage 140:> (0 + 1) / 1] [Stage 141:> (0 + 1) / 1] [Stage 142:> (0 + 1) / 1]

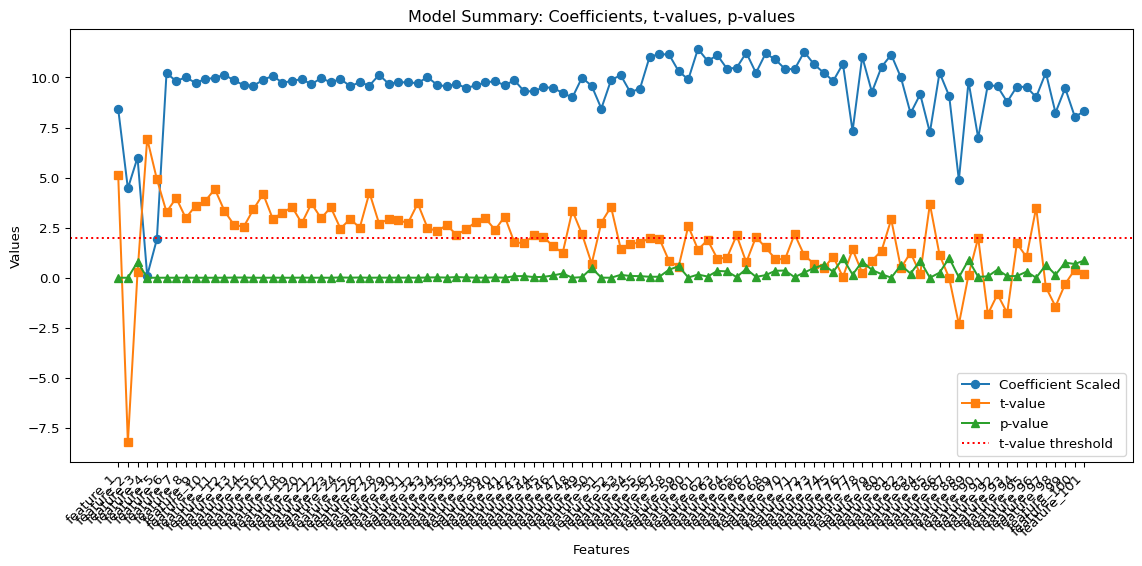
R2: 0.36831696100365496  
RMSE: 36025.51327867095

se = summary\_poly.coefficientStandardErrors[1:]  
tvals = summary\_poly.tValues[1:]  
pvals = summary\_poly.pValues[1:]  
  
coef\_df = pd.DataFrame({  
 "Feature": [f"feature\_{i+1}" for i in range(len(model\_poly.coefficients))],  
 "Coefficient": model\_poly.coefficients.toArray(),  
 "StdError": se,  
 "tValue": tvals,  
 "pValue": pvals  
})  
  
coef\_df.head()

|  | Feature | Coefficient | StdError | tValue | pValue |
| --- | --- | --- | --- | --- | --- |
| 0 | feature\_1 | 4534.036373 | 16.938884 | 5.113944 | 3.182049e-07 |
| 1 | feature\_2 | 86.624507 | 49.042361 | -8.224292 | 2.220446e-16 |
| 2 | feature\_3 | -403.338710 | 4.098090 | 0.271263 | 7.861914e-01 |
| 3 | feature\_4 | 1.111660 | 0.982271 | 6.941764 | 3.982370e-12 |
| 4 | feature\_5 | 6.818695 | 5553.886969 | 4.923227 | 8.575835e-07 |

### 4.1.3 Overall the polynomial model performed worse than the linear regression model. All of the p value, t value, and coefficients experience high volatilitity and varying results.

# Scale coefficients with log  
coef\_df["Coefficient\_log"] = np.log(np.abs(coef\_df["Coefficient"]) + 1e-8)   
  
plt.figure(figsize=(12,6))  
  
# x-axis = features  
x = coef\_df["Feature"]  
  
# plot each statistic as a line  
plt.plot(x, coef\_df["Coefficient\_log"], marker='o', label="Coefficient Scaled")  
plt.plot(x, coef\_df["tValue"], marker='s', label="t-value")  
plt.plot(x, coef\_df["pValue"], marker='^', label="p-value")  
  
plt.axhline(y=2, color='red', linestyle=':', label='t-value threshold')  
  
plt.xticks(rotation=45, ha='right') # rotate feature names for readability  
plt.xlabel("Features")  
plt.ylabel("Values")  
plt.title("Model Summary: Coefficients, t-values, p-values")  
plt.legend()  
plt.tight\_layout()  
plt.show()



# 5. Random Forest

## 5.1 Setup model

### 5.1.1 100 trees, 10 levels of depth to each branch, max of 18 bins for feature categorization.

from pyspark.ml.regression import RandomForestRegressor  
  
rf = RandomForestRegressor(  
 labelCol="SALARY",  
 featuresCol="features",  
 numTrees=100, # number of trees (more trees = more stability)  
 maxDepth=10, # depth of each tree  
 maxBins=18, # controls how continuous features are binned  
 seed=42  
)

### 5.1.2 Like the other models, assemble the data and prepare it using the pipeline setup earlier in the project.

rf\_pipeline = Pipeline(stages=indexers + encoders + [assembler, rf])

### 5.1.3 Train and then make predictions on test data. [stopping point]

rf\_model = rf\_pipeline.fit(train\_df)  
predictions = rf\_model.transform(test\_df)

[Stage 143:> (0 + 1) / 1] [Stage 149:> (0 + 1) / 1] [Stage 155:> (0 + 1) / 1] [Stage 161:> (0 + 1) / 1] [Stage 162:> (0 + 1) / 1] [Stage 163:> (0 + 1) / 1] [Stage 165:> (0 + 1) / 1] [Stage 167:> (0 + 1) / 1] [Stage 169:> (0 + 1) / 1] [Stage 171:> (0 + 1) / 1] [Stage 173:> (0 + 1) / 1] 25/10/17 03:39:51 WARN DAGScheduler: Broadcasting large task binary with size 1319.0 KiB  
[Stage 175:> (0 + 1) / 1] 25/10/17 03:39:53 WARN DAGScheduler: Broadcasting large task binary with size 2.1 MiB  
[Stage 177:> (0 + 1) / 1] 25/10/17 03:39:56 WARN DAGScheduler: Broadcasting large task binary with size 3.4 MiB  
[Stage 179:> (0 + 1) / 1] 25/10/17 03:39:59 WARN DAGScheduler: Broadcasting large task binary with size 5.1 MiB  
[Stage 181:> (0 + 1) / 1][Stage 182:> (0 + 1) / 1] 25/10/17 03:40:05 WARN DAGScheduler: Broadcasting large task binary with size 7.4 MiB  
[Stage 183:> (0 + 1) / 1][Stage 184:> (0 + 1) / 1]

evaluator = RegressionEvaluator(  
 labelCol="SALARY",  
 predictionCol="prediction",  
 metricName="r2"  
)  
  
r2 = evaluator.evaluate(predictions)  
rmse = RegressionEvaluator(  
 labelCol="SALARY", predictionCol="prediction", metricName="rmse"  
).evaluate(predictions)  
  
mae = RegressionEvaluator(  
 labelCol="SALARY", predictionCol="prediction", metricName="mae"  
).evaluate(predictions)  
  
print(f"R²: {r2:.3f}")  
print(f"RMSE: {rmse:.3f}")  
print(f"MAE: {mae:.3f}")

[Stage 185:> (0 + 1) / 1] [Stage 186:> (0 + 1) / 1] [Stage 187:> (0 + 1) / 1]

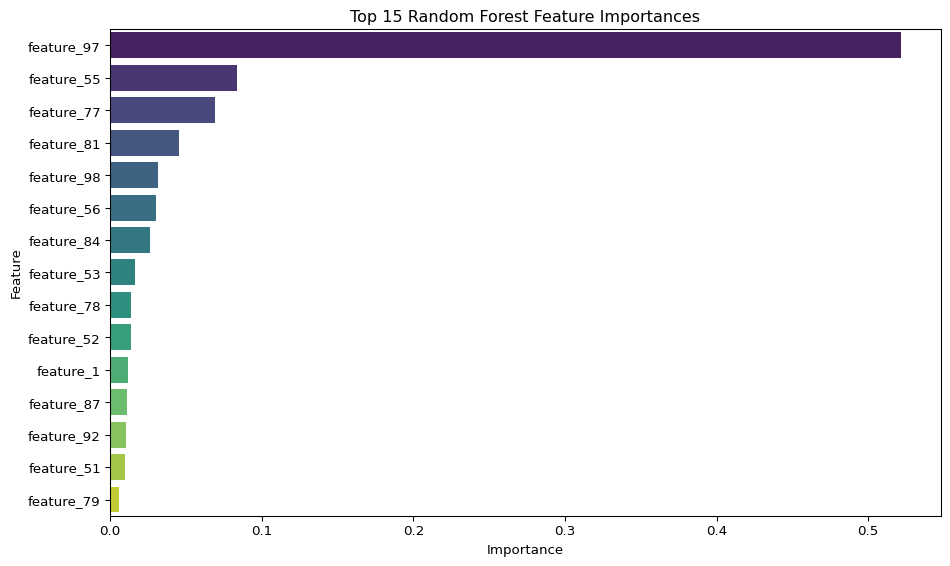
R²: 0.429  
RMSE: 33744.372  
MAE: 25175.654

rf\_stage = rf\_model.stages[-1]   
  
importances = rf\_stage.featureImportances.toArray()  
  
feat\_imp = pd.DataFrame({  
 "Feature": [f"feature\_{i+1}" for i in range(len(importances))],  
 "Importance": importances  
}).sort\_values(by="Importance", ascending=False)  
  
feat\_imp.head(10)

|  | Feature | Importance |
| --- | --- | --- |
| 96 | feature\_97 | 0.521858 |
| 54 | feature\_55 | 0.083882 |
| 76 | feature\_77 | 0.069038 |
| 80 | feature\_81 | 0.045592 |
| 97 | feature\_98 | 0.031267 |
| 55 | feature\_56 | 0.029943 |
| 83 | feature\_84 | 0.025972 |
| 52 | feature\_53 | 0.016110 |
| 77 | feature\_78 | 0.013780 |
| 51 | feature\_52 | 0.013470 |

import matplotlib.pyplot as plt  
import seaborn as sns  
  
plt.figure(figsize=(10,6))  
sns.barplot(data=feat\_imp.head(15), y="Feature", x="Importance", palette="viridis")  
plt.title("Top 15 Random Forest Feature Importances")  
plt.tight\_layout()  
plt.show()

/tmp/ipykernel\_8037/3261347940.py:5: FutureWarning:  
  
  
  
Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `y` variable to `hue` and set `legend=False` for the same effect.



# 6. Model Comparisons