Assignment 04

Anu Sharma

October 3, 2025

## 1 Github URL

https://github.com/met-ad-688/assignment-04-anush-09.git

## 2 Generative AI Use Declaration

1. Used AI for code cleaning and sorting errors.
2. Used it for figuring out some plotting issues in the last cell.

## 3 Note to Instructor:

I used Generalized Linear Regression considering Module05 Lab02 Section4.

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","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)

## 4 Feature Engineering

from pyspark.sql.functions import col, pow  
from pyspark.ml.feature import StringIndexer, OneHotEncoder, VectorAssembler  
from pyspark.ml import Pipeline  
from pyspark.ml.regression import GeneralizedLinearRegression  
  
# Drop missing values  
key\_features = [  
 "DURATION", "SALARY\_FROM", "SALARY\_TO", "MIN\_YEARS\_EXPERIENCE",  
 "EMPLOYMENT\_TYPE\_NAME", "STATE\_NAME", "SALARY"  
]  
df\_clean = df.dropna(subset=key\_features)  
  
# Categorical transformations  
cat\_cols = ["EMPLOYMENT\_TYPE\_NAME", "STATE\_NAME"]  
indexers = [StringIndexer(inputCol=c, outputCol=c+"\_IDX", handleInvalid="keep") for c in cat\_cols]  
encoders = [OneHotEncoder(inputCol=c+"\_IDX", outputCol=c+"\_OHE", dropLast=True) for c in cat\_cols]  
  
# Assemble features  
cont\_cols = ["DURATION", "SALARY\_FROM", "SALARY\_TO", "MIN\_YEARS\_EXPERIENCE"]  
assembler\_inputs = cont\_cols + [c+"\_OHE" for c in cat\_cols]  
  
assembler = VectorAssembler(  
 inputCols=assembler\_inputs,  
 outputCol="features"  
)  
  
pipeline = Pipeline(stages=indexers + encoders + [assembler])  
df\_transformed = pipeline.fit(df\_clean).transform(df\_clean)  
  
# Train-test split (80/20)  
train\_df, test\_df = df\_transformed.randomSplit([0.8, 0.2], seed=42)  
# 80% for training and 20% for testing to ensure enough data for model training and validation.  
  
# Create polynomial feature ( by squaring MIN\_YEARS\_EXPERIENCE)  
df\_poly = df\_transformed.withColumn(  
 "MIN\_YEARS\_EXPERIENCE\_SQ", pow(col("MIN\_YEARS\_EXPERIENCE"), 2)  
)  
  
# Assemble polynomial features into new vector  
poly\_assembler = VectorAssembler(  
 inputCols=["features", "MIN\_YEARS\_EXPERIENCE\_SQ"],  
 outputCol="features\_poly"  
)  
df\_final = poly\_assembler.transform(df\_poly)  
df\_final.show(5)

+--------------------+-----------------+----------------------+----------+--------+---------+--------+-------------------+--------------------+--------------------+-----------+-------------------+--------------------+--------------------+---------------+----------------+--------+------------+-----------+-------------------+----------------+---------------------+-------------+-------------------+-------------+------------------+---------------+--------------------+--------------------+--------------------+-------------+------+-----------+----------------+-------------------+---------+-----------+--------------------+--------------------+------------------+------+--------------------+-----+--------------------+-----+----------+---------------+--------------------+---------------+--------------------+------------+--------------------+------------+--------------------+------+--------------------+------+--------------------+------+--------------------+------+--------------------+------+--------------------+------------------+--------------------+--------------------+--------------------+--------------------+--------------------+-----------------------+--------------+-------------------+--------------------+--------------------+--------------------+--------------------+----------+--------------------+----------+--------------------+-----------------+--------------------+---------------+--------------------+------------+--------------------+----------+--------------------+----------+--------------------+----------+---------------+----------+---------------+---------------+--------------------+--------------+--------------------+--------------------------+-------------------------------+--------------------+-------------------------+-----------------------------+----------------------------------+-----------------+----------------------+-----------------------+----------------------------+------------------+-----------------------+-------+--------------------+-------+--------------------+-------+---------------+-------+---------------+-----------------+----------------------+------------+--------------------+------------+--------------------+------------+--------------------+------------+--------------------+------------+--------------------+------------------------+--------------+------------------------+---------------+--------------------+-----------------------+--------------------+  
| 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|EMPLOYMENT\_TYPE\_NAME\_IDX|STATE\_NAME\_IDX|EMPLOYMENT\_TYPE\_NAME\_OHE| STATE\_NAME\_OHE| features|MIN\_YEARS\_EXPERIENCE\_SQ| features\_poly|  
+--------------------+-----------------+----------------------+----------+--------+---------+--------+-------------------+--------------------+--------------------+-----------+-------------------+--------------------+--------------------+---------------+----------------+--------+------------+-----------+-------------------+----------------+---------------------+-------------+-------------------+-------------+------------------+---------------+--------------------+--------------------+--------------------+-------------+------+-----------+----------------+-------------------+---------+-----------+--------------------+--------------------+------------------+------+--------------------+-----+--------------------+-----+----------+---------------+--------------------+---------------+--------------------+------------+--------------------+------------+--------------------+------+--------------------+------+--------------------+------+--------------------+------+--------------------+------+--------------------+------------------+--------------------+--------------------+--------------------+--------------------+--------------------+-----------------------+--------------+-------------------+--------------------+--------------------+--------------------+--------------------+----------+--------------------+----------+--------------------+-----------------+--------------------+---------------+--------------------+------------+--------------------+----------+--------------------+----------+--------------------+----------+---------------+----------+---------------+---------------+--------------------+--------------+--------------------+--------------------------+-------------------------------+--------------------+-------------------------+-----------------------------+----------------------------------+-----------------+----------------------+-----------------------+----------------------------+------------------+-----------------------+-------+--------------------+-------+--------------------+-------+---------------+-------+---------------+-----------------+----------------------+------------+--------------------+------------+--------------------+------------+--------------------+------------+--------------------+------------+--------------------+------------------------+--------------+------------------------+---------------+--------------------+-----------------------+--------------------+  
|57b527ea0f91db5bb...| 9/6/2024| 2024-09-06 20:32:...| 0|6/2/2024|7/27/2024| 55|[\n "Job Board"\n]|[\n "simplyhired...|[\n "https://www...| []| NULL|Power, Utilities ...|Power, Utilities ...| 7/27/2024| 55| 5732448| Deloitte| Deloitte| false| [\n 2,\n 3\n]| [\n "Bachelor's ...| 2| Bachelor's degree| 3| Master's degree| 1|Full-time (> 32 h...| 6| NULL| false|192800| 0| [None]| year| 241000| 144600|{\n "lat": 42.33...| RGV0cm9pdCwgTUk=| Detroit, MI| 26163| Wayne, MI|19820|Detroit-Warren-De...| 26| Michigan| 26163| Wayne, MI| 26163| Wayne, MI| 19820|Detroit-Warren-De...| 19820|Detroit-Warren-De...| 54|Professional, Sci...| 541|Professional, Sci...| 5416|Management, Scien...| 54161|Management Consul...|541611|Administrative Ma...|ET8AEDEB1F4C3091D3|Management Consul...|power utilities r...|[\n "KS122VL71WF...|[\n "Design Spec...|[\n "KS122VL71WF...| [\n "Design Spec...| []| []|[\n "KS1218W78FG...|[\n "Management"...|[\n "KS1219W70LY...|[\n "C++ (Progra...|15-2051.01|Business Intellig...|15-2051.01|Business Intellig...|[\n "45.0702"\n]|[\n "Geographic ...|[\n "45.07"\n]|[\n "Geography a...|[\n "45"\n]|[\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 3\n]| [\n "Green Jobs:...| 54|Professional, Sci...| 541|Professional, Sci...| 5416|Management, Scien...| 54161|Management Consul...| 541611|Administrative Ma...| 0.0| 8.0| (3,[0],[1.0])| (51,[8],[1.0])|(58,[0,1,2,3,4,15...| 36.0|(59,[0,1,2,3,4,15...|  
|dd191e2ce3062c371...| 9/6/2024| 2024-09-06 20:32:...| 0|6/2/2024|6/20/2024| 18|[\n "Job Board"\n]|[\n "phoenixrecr...|[\n "https://www...| []| NULL| SAP FSCM Consultant|Job Description: ...| 6/20/2024| 18| 8592955| Accenture| Accenture| false| [\n 1,\n 2\n]| [\n "Associate d...| 1| Associate degree| 2| Bachelor's degree| 1|Full-time (> 32 h...| 12| NULL| false|125900| 0| [None]| year| 188600| 63200|{\n "lat": 0,\n ...|W1Vua25vd24gQ2l0e...|[Unknown City], AZ| 4999|[Unknown county], AZ| NULL| NULL| 4| Arizona| 4999|[Unknown county], AZ| 4999|[Unknown county], AZ| NULL| NULL| NULL| NULL| 54|Professional, Sci...| 541|Professional, Sci...| 5415|Computer Systems ...| 54151|Computer Systems ...|541512|Computer Systems ...|ETF594A2C05D212506|Peoplesoft FSCM C...| sap fscm consultant|[\n "KS7G7VL78R2...|[\n "Profit Cent...|[\n "KS7G7VL78R2...| [\n "Profit Cent...| []| []|[\n "KS122ZF75YV...|[\n "Digitizatio...|[\n "KS7G7VL78R2...|[\n "Profit Cent...|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...| 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| NULL| NULL| 54|Professional, Sci...| 541|Professional, Sci...| 5415|Computer Systems ...| 54151|Computer Systems ...| 541512|Computer Systems ...| 0.0| 11.0| (3,[0],[1.0])|(51,[11],[1.0])|(58,[0,1,2,3,4,18...| 144.0|(59,[0,1,2,3,4,18...|  
|146621e071735303b...| 7/12/2024| 2024-07-13 03:22:...| 0|6/2/2024|6/22/2024| 20| [\n "Company"\n]| [\n "jobs.net"\n]|[\n "https://bcf...| []| NULL| DATA ANALYTICS|Data Modeler Mode...| 6/22/2024| 20|99484525| BCforward| BCforward| true| [\n 99\n]| [\n "No Educatio...| 99|No Education Listed| NULL| NULL| 1|Full-time (> 32 h...| 5| NULL| false|118560| 1| Remote| hour| 120328| 116792|{\n "lat": 39.76...|SW5kaWFuYXBvbGlzL...| Indianapolis, IN| 18097| Marion, IN|26900|Indianapolis-Carm...| 18| Indiana| 18097| Marion, IN| 18097| Marion, IN| 26900|Indianapolis-Carm...| 26900|Indianapolis-Carm...| 54|Professional, Sci...| 541|Professional, Sci...| 5416|Management, Scien...| 54161|Management Consul...|541611|Administrative Ma...|ET0000000000000000| Unclassified| data analytics|[\n "KSOH5A22ONU...|[\n "Databricks"...|[\n "KSOH5A22ONU...| [\n "Databricks"...| []| []|[\n "KS1203C6N9B...|[\n "Research",\...|[\n "KSOH5A22ONU...|[\n "Databricks"...|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 7\n]| [\n "Artificial ...| 54|Professional, Sci...| 541|Professional, Sci...| 5416|Management, Scien...| 54161|Management Consul...| 541611|Administrative Ma...| 0.0| 21.0| (3,[0],[1.0])|(51,[21],[1.0])|(58,[0,1,2,3,4,28...| 25.0|(59,[0,1,2,3,4,28...|  
|799924ec2f2bf4f2e...| 9/6/2024| 2024-09-06 20:32:...| 0|6/2/2024|7/27/2024| 55|[\n "Job Board"\n]|[\n "simplyhired...|[\n "https://www...| []| NULL|Power, Utilities ...|Power, Utilities ...| 7/27/2024| 55| 5732448| Deloitte| Deloitte| false| [\n 2,\n 3\n]| [\n "Bachelor's ...| 2| Bachelor's degree| 3| Master's degree| 1|Full-time (> 32 h...| 6| NULL| false|192800| 0| [None]| year| 241000| 144600|{\n "lat": 39.10...|Q2luY2lubmF0aSwgT0g=| Cincinnati, OH| 39061| Hamilton, OH|17140|Cincinnati, OH-KY-IN| 39| Ohio| 39061| Hamilton, OH| 39061| Hamilton, OH| 17140|Cincinnati, OH-KY-IN| 17140|Cincinnati, OH-KY-IN| 54|Professional, Sci...| 541|Professional, Sci...| 5416|Management, Scien...| 54161|Management Consul...|541611|Administrative Ma...|ET8AEDEB1F4C3091D3|Management Consul...|power utilities r...|[\n "KS122VL71WF...|[\n "Design Spec...|[\n "KS122VL71WF...| [\n "Design Spec...| []| []|[\n "KS1218W78FG...|[\n "Management"...|[\n "KS1219W70LY...|[\n "C++ (Progra...|15-2051.01|Business Intellig...|15-2051.01|Business Intellig...|[\n "45.0702"\n]|[\n "Geographic ...|[\n "45.07"\n]|[\n "Geography a...|[\n "45"\n]|[\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 3\n]| [\n "Green Jobs:...| 54|Professional, Sci...| 541|Professional, Sci...| 5416|Management, Scien...| 54161|Management Consul...| 541611|Administrative Ma...| 0.0| 7.0| (3,[0],[1.0])| (51,[7],[1.0])|(58,[0,1,2,3,4,14...| 36.0|(59,[0,1,2,3,4,14...|  
|6396d0f5e8c9feaa0...| 9/6/2024| 2024-09-06 20:32:...| 0|6/2/2024|6/18/2024| 16|[\n "Job Board"\n]|[\n "phoenixrecr...|[\n "https://www...| []| NULL| SAP BTP Consultant|Job Description: ...| 6/18/2024| 16| 8592955| Accenture| Accenture| false| [\n 1,\n 2\n]| [\n "Associate d...| 1| Associate degree| 2| Bachelor's degree| 1|Full-time (> 32 h...| 12| NULL| false|116500| 0| [None]| year| 169800| 63200|{\n "lat": 0,\n ...|W1Vua25vd24gQ2l0e...|[Unknown City], AZ| 4999|[Unknown county], AZ| NULL| NULL| 4| Arizona| 4999|[Unknown county], AZ| 4999|[Unknown county], AZ| NULL| NULL| NULL| NULL| 54|Professional, Sci...| 541|Professional, Sci...| 5415|Computer Systems ...| 54151|Computer Systems ...|541512|Computer Systems ...|ETCBD54F6B90888F99| SAP Consultants| sap btp consultant|[\n "KS122ZF75YV...|[\n "Digitizatio...|[\n "ES43DB1E2DE...| [\n "Cloud Servi...| []| []|[\n "KS122ZF75YV...|[\n "Digitizatio...|[\n "KS1267G65F6...|[\n "Apache Mave...|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...| 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| NULL| NULL| 54|Professional, Sci...| 541|Professional, Sci...| 5415|Computer Systems ...| 54151|Computer Systems ...| 541512|Computer Systems ...| 0.0| 11.0| (3,[0],[1.0])|(51,[11],[1.0])|(58,[0,1,2,3,4,18...| 144.0|(59,[0,1,2,3,4,18...|  
+--------------------+-----------------+----------------------+----------+--------+---------+--------+-------------------+--------------------+--------------------+-----------+-------------------+--------------------+--------------------+---------------+----------------+--------+------------+-----------+-------------------+----------------+---------------------+-------------+-------------------+-------------+------------------+---------------+--------------------+--------------------+--------------------+-------------+------+-----------+----------------+-------------------+---------+-----------+--------------------+--------------------+------------------+------+--------------------+-----+--------------------+-----+----------+---------------+--------------------+---------------+--------------------+------------+--------------------+------------+--------------------+------+--------------------+------+--------------------+------+--------------------+------+--------------------+------+--------------------+------------------+--------------------+--------------------+--------------------+--------------------+--------------------+-----------------------+--------------+-------------------+--------------------+--------------------+--------------------+--------------------+----------+--------------------+----------+--------------------+-----------------+--------------------+---------------+--------------------+------------+--------------------+----------+--------------------+----------+--------------------+----------+---------------+----------+---------------+---------------+--------------------+--------------+--------------------+--------------------------+-------------------------------+--------------------+-------------------------+-----------------------------+----------------------------------+-----------------+----------------------+-----------------------+----------------------------+------------------+-----------------------+-------+--------------------+-------+--------------------+-------+---------------+-------+---------------+-----------------+----------------------+------------+--------------------+------------+--------------------+------------+--------------------+------------+--------------------+------------+--------------------+------------------------+--------------+------------------------+---------------+--------------------+-----------------------+--------------------+  
only showing top 5 rows

## 5 Generalized Linear Regression model

lr = GeneralizedLinearRegression(  
 family="gaussian",  
 link="identity",  
 featuresCol="features",  
 labelCol="SALARY",  
 maxIter=10,  
 regParam=0.3  
)  
  
lr\_model = lr.fit(train\_df)  
  
from pyspark.ml.evaluation import RegressionEvaluator  
lr\_predictions = lr\_model.transform(test\_df)  
evaluator = RegressionEvaluator(labelCol="SALARY", predictionCol="prediction")  
  
print("Coefficients:", lr\_model.coefficients)  
print("Intercept:", lr\_model.intercept)  
print("R²:", evaluator.setMetricName("r2").evaluate(lr\_predictions))  
print("RMSE:", evaluator.setMetricName("rmse").evaluate(lr\_predictions))  
print("MAE:", evaluator.setMetricName("mae").evaluate(lr\_predictions))  
  
# Coefficient statistics  
training\_summary = lr\_model.summary  
  
try:  
 coefs = lr\_model.coefficients.toArray().tolist()  
 se = training\_summary.coefficientStandardErrors  
 tvals = training\_summary.tValues  
 pvals = training\_summary.pValues  
  
 coef\_df = spark.createDataFrame(  
 [  
 (float(coefs[i]), float(se[i]), float(tvals[i]), float(pvals[i]),  
 float(coefs[i] - 1.96\*se[i]), float(coefs[i] + 1.96\*se[i]))  
 for i in range(len(coefs))  
 ],  
 ["Coefficient", "StdError", "tValue", "pValue", "CI\_lower", "CI\_upper"]  
 )  
  
 coef\_df.show(truncate=False)  
  
except Exception as e:  
 # Actualy not needed for GLR but I added it when I was previously using LR.  
 print("Coefficient statistics not available (L-BGFS fallback):", str(e))  
# df\_final.printSchema()

Coefficients: [0.10577257338987292,0.4897760646675632,0.5031355811149286,5.821447063989891,-63.13234211764908,-37.24918495447914,240.2496005871974,48.31465068112287,152.96207700756838,-176.77153212602502,85.33910314560474,434.28702577440487,-161.42079058360767,-15.678039153034717,-118.88023960359375,-161.45114829498186,-102.29690138402427,202.46320733110622,-141.30430068749794,-10.336921084699812,46.18232664748482,-139.7522513504501,-91.77765784576877,756.3227801189014,88.81214010053442,-194.06691355630176,92.42948084492659,-238.66548174672735,-232.9013492626481,49.64728458137663,-169.13719812248235,-145.48484292668024,-121.95157094464349,-99.6819901995056,-159.90690621771986,-201.88892491962216,-187.93226914248663,-97.41512683302918,-200.87692671645,-193.2008460392148,-224.35294989361188,50.530082652301964,42.19504489586543,-80.67373549847771,-132.2894187558734,-134.85646865840482,-191.28731365448144,-161.90766503071904,147.48514217917366,344.69364885521287,172.6083494391896,-142.18206008275982,-192.08893673890222,-86.81171897251811,-139.9320627067798,-118.84967713486162,374.329274291932,-226.00705073681388]  
Intercept: 771.471741322882

R²: 0.9991007618414153

RMSE: 1269.8322518177556

MAE: 439.38359836620407

+-------------------+---------------------+----------------------+-------------------+-------------------+------------------+  
|Coefficient |StdError |tValue |pValue |CI\_lower |CI\_upper |  
+-------------------+---------------------+----------------------+-------------------+-------------------+------------------+  
|0.10577257338987292|0.8210812224588767 |0.12882108431747807 |0.8975014656141185 |-1.5035466226295255|1.7150917694092713|  
|0.4897760646675632 |4.7249581342235624E-4|1036.5722843553756 |0.0 |0.4888499728732554 |0.490702156461871 |  
|0.5031355811149286 |3.462378911127178E-4 |1453.149970091323 |0.0 |0.5024569548483477 |0.5038142073815095|  
|5.821447063989891 |4.284466788662476 |1.358733151904587 |0.17425770783310224|-2.576107841788562 |14.219001969768343|  
|-63.13234211764908 |15760.586188378582 |-0.004005710280256017 |0.9968039833938374 |-30953.88127133967 |30827.61658710437 |  
|-37.24918495447914 |15760.644925545 |-0.0023634302486001263|0.9981142982265387 |-30928.113239022678|30853.61486911372 |  
|240.2496005871974 |15760.786548337737 |0.015243503225575763 |0.9878381785111885 |-30650.892034154767|31131.391235329163|  
|48.31465068112287 |4500.489171196371 |0.010735422049305658 |0.9914347225454412 |-8772.644124863764 |8869.27342622601 |  
|152.96207700756838 |4500.527913588752 |0.03398758544430299 |0.9728876379345217 |-8668.072633626385 |8973.996787641523 |  
|-176.77153212602502|4500.662388715528 |-0.03927678125096491 |0.9686703973498092 |-8998.06981400846 |8644.52674975641 |  
|85.33910314560474 |4500.661123424321 |0.01896145939569754 |0.9848721785122763 |-8735.956698766064 |8906.634905057273 |  
|434.28702577440487 |4500.712114327836 |0.09649295816807957 |0.923130741260461 |-8387.108718308154 |9255.682769856965 |  
|-161.42079058360767|4500.800288897419 |-0.035864908510115576 |0.9713906968532522 |-8982.989356822549 |8660.147775655332 |  
|-15.678039153034717|4500.846381301049 |-0.003483353535052823 |0.9972207517613536 |-8837.336946503092 |8805.98086819702 |  
|-118.88023960359375|4500.835491139143 |-0.026412927074903073 |0.9789284399252751 |-8940.517802236314 |8702.757323029127 |  
|-161.45114829498186|4500.876955298087 |-0.03587104244316965 |0.9713858059543845 |-8983.169980679231 |8660.267684089267 |  
|-102.29690138402427|4500.890163796606 |-0.022728148801954864 |0.9818675149292697 |-8924.041622425373 |8719.447819657324 |  
|202.46320733110622 |4500.914400805749 |0.04498268336204338 |0.964121889624981 |-8619.329018248163 |9024.255432910375 |  
|-141.30430068749794|4500.940965632624 |-0.03139439103210657 |0.974955556934114 |-8963.148593327442 |8680.539991952446 |  
|-10.336921084699812|4500.925771743812 |-0.002296621097284823 |0.9981676026572368 |-8832.151433702571 |8811.477591533172 |  
+-------------------+---------------------+----------------------+-------------------+-------------------+------------------+  
only showing top 20 rows

### 5.1 Inference

The Generalized Linear Regression model gave very high accuracy (R² ≈ 0.9991), explaining almost all variation in salaries. Prediction errors were very small (RMSE ≈ 1269, MAE ≈ 439). As expected, more years of experience are linked with higher pay, and job type and location also matter. Looking at coefficients, some predictors (like the second and third features) are highly significant with very low p‑values, while many others have large standard errors and are not statistically meaningful. This shows the presence of multicollinearity: several features are strongly correlated, so the model predicts well overall but individual coefficient estimates are unstable.

## 6 Polynomial Regression model

from pyspark.ml.regression import GeneralizedLinearRegression  
  
# train/test  
train\_poly, test\_poly = df\_final.randomSplit([0.8, 0.2], seed=42)  
  
# Linear Regression model using polynomial features  
lr\_poly = GeneralizedLinearRegression(  
 family="gaussian",  
 link="identity",  
 featuresCol="features\_poly",  
 labelCol="SALARY",  
 maxIter=10,  
 regParam=0.3  
)  
  
# Fit model on training data  
lr\_poly\_model = lr\_poly.fit(train\_poly)  
  
from pyspark.ml.evaluation import RegressionEvaluator  
poly\_predictions = lr\_poly\_model.transform(test\_poly)  
evaluator = RegressionEvaluator(labelCol="SALARY", predictionCol="prediction")  
  
print("Polynomial Regression Results")  
print("Coefficients:", lr\_poly\_model.coefficients)  
print("Intercept:", lr\_poly\_model.intercept)  
print("R²:", evaluator.setMetricName("r2").evaluate(poly\_predictions))  
print("RMSE:", evaluator.setMetricName("rmse").evaluate(poly\_predictions))  
print("MAE:", evaluator.setMetricName("mae").evaluate(poly\_predictions))  
  
# Coefficient statistics  
poly\_summary = lr\_poly\_model.summary  
  
try:  
 coefs = lr\_poly\_model.coefficients.toArray().tolist()  
 se = poly\_summary.coefficientStandardErrors  
 tvals = poly\_summary.tValues  
 pvals = poly\_summary.pValues  
  
 coef\_poly\_df = spark.createDataFrame(  
 [  
 (float(coefs[i]), float(se[i]), float(tvals[i]), float(pvals[i]),  
 float(coefs[i] - 1.96\*se[i]), float(coefs[i] + 1.96\*se[i]))  
 for i in range(len(coefs))  
 ],  
 ["Coefficient", "StdError", "tValue", "pValue", "CI\_lower", "CI\_upper"]  
 )  
  
 coef\_poly\_df.show(truncate=False)  
  
except Exception as e:  
 # Actualy not needed for GLR but I added it when I was previously using LR.  
 print("Coefficient statistics not available (L-BGFS fallback):", str(e))

Polynomial Regression Results  
Coefficients: [0.10864189686946173,0.4898446498795814,0.5031330094998685,-7.294074629446415,-61.606183030030984,-36.42477835447172,234.5785726337609,47.66495235497268,153.30872148381684,-176.2483283327536,83.9420652739141,434.07175893403144,-161.97789924707783,-15.370297973052363,-119.3787727802767,-161.94452829454534,-102.0712038173867,201.45561915371567,-140.2180176255666,-9.181294490038646,45.46520302014749,-139.59475217986548,-90.96205773513772,755.7997862985653,91.25321470444824,-192.37388895594734,92.06768904691285,-238.82762343754223,-232.79724289167783,46.988988491411725,-168.5208658938645,-145.86104897935874,-122.65183729247258,-97.95064001860672,-155.79027411246554,-202.70247726846625,-187.7309421134543,-98.09748718742412,-200.76815604456365,-192.61329916789478,-224.5383725545915,52.618094862437424,43.978613627274086,-81.22525014807809,-131.6257250593234,-134.83735400335175,-191.49970597043983,-162.87607913749576,146.14802544190084,341.3370623433174,171.74420670609214,-142.83066925953963,-195.21965382722956,-85.13674407585455,-139.27991520093798,-118.77551011346809,376.6718850326892,-225.64627278434565,0.9616178547957062]  
Intercept: 795.9342167087742

R²: 0.9991016319916092

RMSE: 1269.2177248430587

MAE: 438.815563380054

+-------------------+---------------------+----------------------+------------------+-------------------+-------------------+  
|Coefficient |StdError |tValue |pValue |CI\_lower |CI\_upper |  
+-------------------+---------------------+----------------------+------------------+-------------------+-------------------+  
|0.10864189686946173|0.8210834577114463 |0.13231529127656813 |0.8947372190517244|-1.500681680244973 |1.7179654739838965 |  
|0.4898446498795814 |4.7714484248743565E-4|1026.6162520502987 |0.0 |0.488909445988306 |0.49077985377085676|  
|0.5031330094998685 |3.4624581479118603E-4|1453.1092882763016 |0.0 |0.5024543677028778 |0.5038116512968592 |  
|-7.294074629446415 |13.410941333060745 |-0.543889832063094 |0.5865277899146577|-33.57951964224547 |18.991370383352645 |  
|-61.606183030030984|15760.538813755664 |-0.0039088881261002085|0.996881234114777 |-30952.262257991133|30829.04989193107 |  
|-36.42477835447172 |15760.597501615233 |-0.0023111292798854047|0.9981560271332635|-30927.19588152033 |30854.346324811388 |  
|234.5785726337609 |15760.740061617484 |0.0148837282841201 |0.9881251986163893|-30656.471948136506|31125.629093404026 |  
|47.66495235497268 |4500.475667629807 |0.010591092114508779 |0.9915498721391205|-8773.267356199447 |8868.597260909393 |  
|153.30872148381684 |4500.51437841101 |0.03406471096264897 |0.9728261376737084|-8667.699460201762 |8974.316903169394 |  
|-176.2483283327536 |4500.648869151094 |-0.03916064848800287 |0.9687629846908643|-8997.520111868896 |8645.02345520339 |  
|83.9420652739141 |4500.647778877783 |0.018651107440103808 |0.9851197544098045|-8737.32758132654 |8905.211711874368 |  
|434.07175893403144 |4500.6985708954635 |0.09644541888253318 |0.9231684953558019|-8387.297440021077 |9255.44095788914 |  
|-161.97789924707783|4500.786772737273 |-0.03598879649847679 |0.9712919145273127|-8983.519973812134 |8659.564175317977 |  
|-15.370297973052363|4500.832842508601 |-0.0034149897387625522|0.9972752967985401|-8837.00266928991 |8806.262073343807 |  
|-119.3787727802767 |4500.821968423608 |-0.026523771350611444 |0.978840031946246 |-8940.989830890549 |8702.232285329994 |  
|-161.94452829454534|4500.863431924387 |-0.03598076918883637 |0.9712983151072367|-8983.636854866345 |8659.747798277254 |  
|-102.0712038173867 |4500.876620307754 |-0.0226780719464395 |0.9819074593485553|-8923.789379620584 |8719.646971985812 |  
|201.45561915371567 |4500.900957815064 |0.04475895404983786 |0.9643002163572982|-8620.31025816381 |9023.221496471242 |  
|-140.2180176255666 |4500.927539746499 |-0.031153138189259532 |0.9751479506306358|-8962.035995528704 |8681.599960277572 |  
|-9.181294490038646 |4500.9123621176905 |-0.0020398740858217517|0.9983724523724109|-8830.969524240712 |8812.606935260634 |  
+-------------------+---------------------+----------------------+------------------+-------------------+-------------------+  
only showing top 20 rows

### 6.1 Inference

The Polynomial Regression model also achieved very high accuracy (R² ≈ 0.9991), explaining nearly all variation in salaries. Errors were very small (RMSE ≈ 1269, MAE ≈ 439), almost the same as the linear model. The results again confirm that experience, job type, and location strongly influence pay. Looking at coefficients, a few predictors (like the second and third features) are highly significant with very low p‑values, but many others show very large standard errors and are not statistically reliable. This again points to multicollinearity: several features overlap in information, so while the model predicts salaries very well, the individual coefficient estimates remain unstable

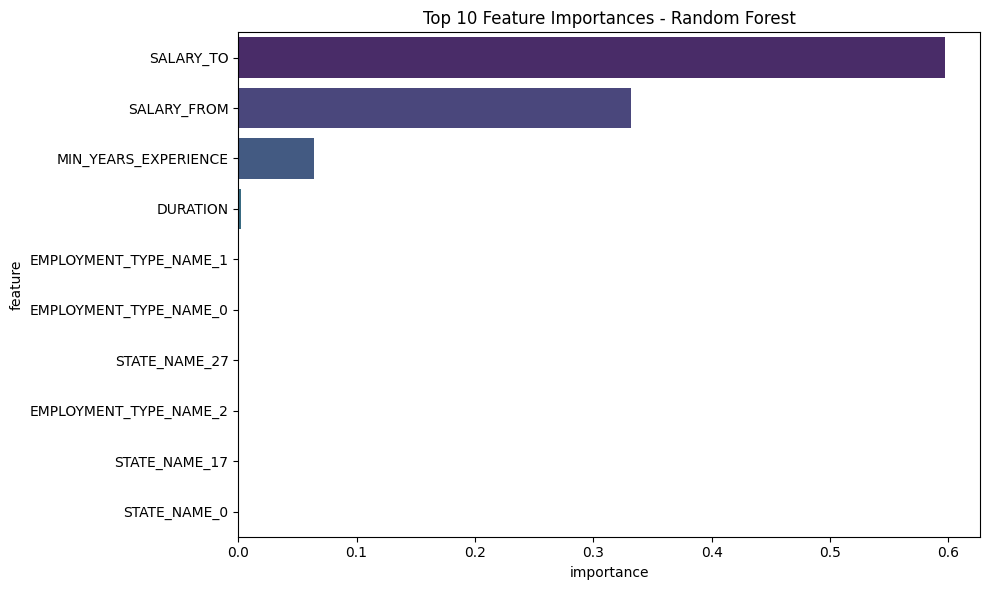
## 7 Random Forest Regression

from pyspark.ml.regression import RandomForestRegressor  
from pyspark.ml.evaluation import RegressionEvaluator  
  
# Train-test split  
train\_rf, test\_rf = df\_transformed.randomSplit([0.8, 0.2], seed=42)  
  
# Random Forest Regression  
rf = RandomForestRegressor(  
 featuresCol="features",  
 labelCol="SALARY",  
 numTrees=200,  
 maxDepth=6,  
 seed=42  
)  
  
# Model  
rf\_model = rf.fit(train\_rf)  
  
# Evaluation  
rf\_predictions = rf\_model.transform(test\_rf)  
  
evaluator = RegressionEvaluator(labelCol="SALARY", predictionCol="prediction")  
  
r2 = evaluator.setMetricName("r2").evaluate(rf\_predictions)  
rmse = evaluator.setMetricName("rmse").evaluate(rf\_predictions)  
mae = evaluator.setMetricName("mae").evaluate(rf\_predictions)  
  
print("Random Forest Results")  
print("R²:", r2)  
print("RMSE:", rmse)  
print("MAE:", mae)

Random Forest Results  
R²: 0.9724973538145091  
RMSE: 7022.572482052369  
MAE: 4380.591032239985

### 7.1 Feature Importance Plot

import pandas as pd  
import seaborn as sns  
import matplotlib.pyplot as plt  
  
# OHE vector size  
ohe\_sizes = {c+"\_OHE": df\_transformed.select(c+"\_OHE").head()[0].size for c in cat\_cols}  
  
# Expanded feature names  
expanded\_feature\_names = []  
for col in cont\_cols:  
 expanded\_feature\_names.append(col)  
  
for c in cat\_cols:  
 for i in range(ohe\_sizes[c+"\_OHE"]):  
 expanded\_feature\_names.append(f"{c}\_{i}")  
  
importances = rf\_model.featureImportances.toArray()  
  
feat\_imp = pd.DataFrame({  
 "feature": expanded\_feature\_names,  
 "importance": importances  
})  
  
# Top 10 plot  
top10 = feat\_imp.sort\_values("importance", ascending=False).head(10)  
  
plt.figure(figsize=(10,6))  
sns.barplot(x="importance", y="feature", data=top10, palette="viridis")  
plt.title("Top 10 Feature Importances - Random Forest")  
plt.tight\_layout()  
plt.savefig("\_output/rf\_feature\_importance.png")  
plt.show()  
plt.close()



### 7.2 Inference

The Random Forest model explained about 97% of salary variation (R² ≈ 0.97). But its errors were higher (RMSE ≈ 7023, MAE ≈ 4381) compared to linear models. From feature importance, we see that SALARY\_TO and SALARY\_FROM are the main drivers, with years of experience adding some effect. Other features like job type and state have very small impact. This means the model predicts well, but it mostly depends on salary range fields and gives less extra insight.

## 8 Model Comparison

import pandas as pd  
import numpy as np  
from pyspark.ml.evaluation import RegressionEvaluator  
  
# Predictions for all three models  
lr\_predictions = lr\_model.transform(test\_df)  
poly\_predictions = lr\_poly\_model.transform(test\_poly)  
rf\_predictions = rf\_model.transform(test\_rf)  
  
evaluator = RegressionEvaluator(labelCol="SALARY", predictionCol="prediction")  
  
# Linear Regression metrics  
lr\_r2 = evaluator.setMetricName("r2").evaluate(lr\_predictions)  
lr\_rmse = evaluator.setMetricName("rmse").evaluate(lr\_predictions)  
lr\_mae = evaluator.setMetricName("mae").evaluate(lr\_predictions)  
  
# AIC for Linear Regression  
lr\_aic = lr\_model.summary.aic  
print(f"Linear Regression AIC: {lr\_aic}")  
  
# Polynomial Regression metrics  
poly\_r2 = evaluator.setMetricName("r2").evaluate(poly\_predictions)  
poly\_rmse = evaluator.setMetricName("rmse").evaluate(poly\_predictions)  
poly\_mae = evaluator.setMetricName("mae").evaluate(poly\_predictions)  
  
# AIC for Polynomial Regression  
poly\_aic = lr\_poly\_model.summary.aic  
print(f"Polynomial Regression AIC: {poly\_aic}")  
  
# AIC is not directly available for Random Forest  
rf\_aic = None  
  
# BIC calculation for Linear and Polynomial models  
def calculate\_bic(model\_summary, n\_obs):  
 """Calculate BIC for PySpark GeneralizedLinearRegression models"""  
 try:  
 k = len(model\_summary.coefficientStandardErrors) + 1  
 deviance = model\_summary.deviance  
 dispersion = model\_summary.dispersion  
  
 # Log Likelihood calculation  
 log\_likelihood = -0.5 \* (n\_obs \* np.log(2 \* np.pi) + n\_obs \* np.log(dispersion) + deviance/dispersion)  
  
 # BIC calculation  
 bic = k \* np.log(n\_obs) - 2 \* log\_likelihood  
 return bic  
 except Exception as e:  
 print(f"BIC calculation failed: {str(e)}")  
 return None  
  
n\_obs = test\_df.count()  
  
lr\_bic = calculate\_bic(lr\_model.summary, n\_obs)  
poly\_bic = calculate\_bic(lr\_poly\_model.summary, n\_obs)  
  
print(f"Linear Regression BIC: {lr\_bic}")  
print(f"Polynomial Regression BIC: {poly\_bic}")  
  
comparison\_data = {  
 'Model': ['Generalized Linear Regression', 'Polynomial Regression', 'Random Forest'],  
 'R²': [lr\_r2, poly\_r2, r2],  
 'RMSE': [lr\_rmse, poly\_rmse, rmse],  
 'MAE': [lr\_mae, poly\_mae, mae],  
 'AIC': [lr\_aic, poly\_aic, 'N/A'],  
 'BIC': [lr\_bic, poly\_bic, 'N/A']  
}  
  
comparison\_df = pd.DataFrame(comparison\_data)  
print("Model Comparison:")  
print(comparison\_df.to\_string(index=False))

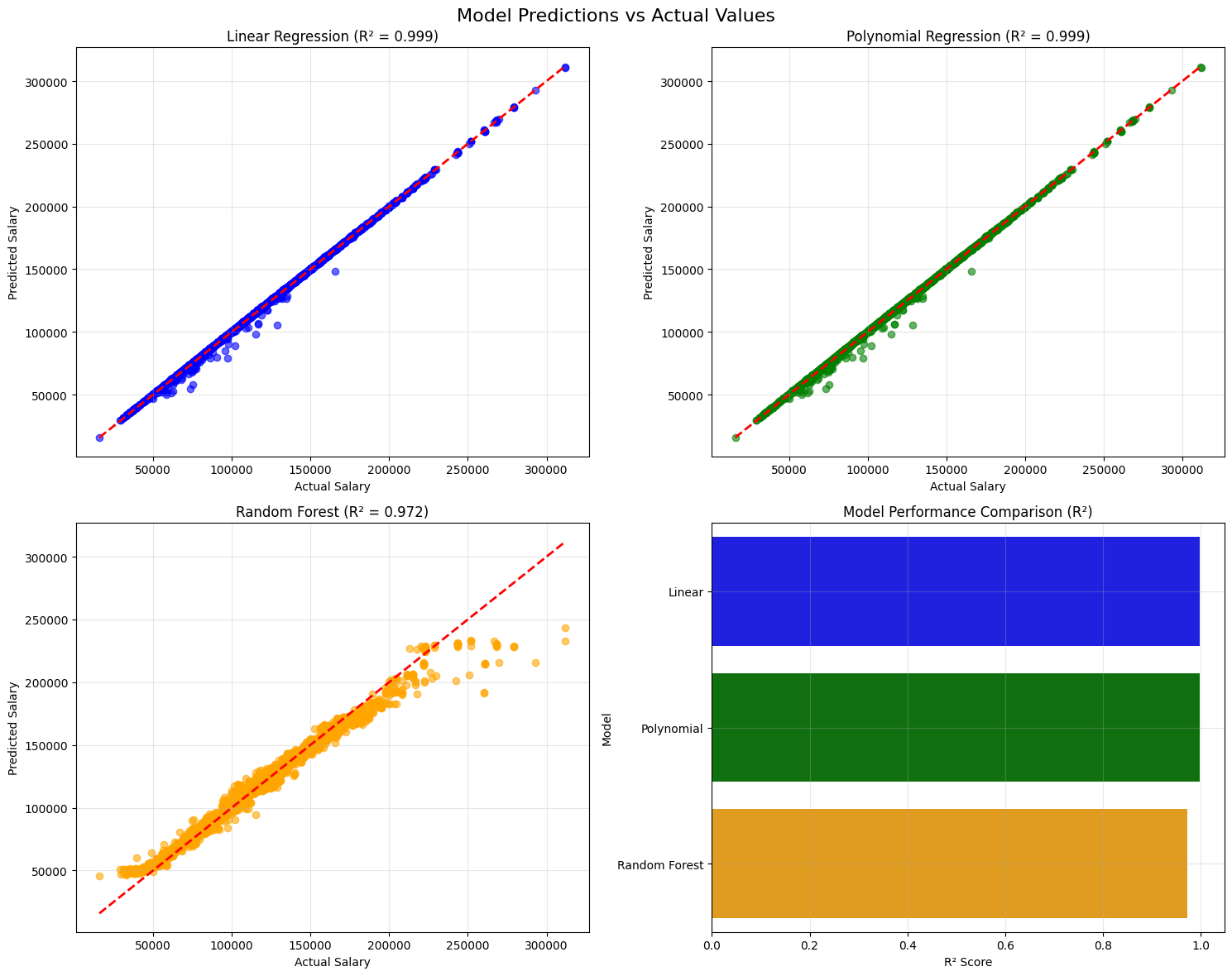
Linear Regression AIC: 198674.9323303638

Polynomial Regression AIC: 198675.85731451394

Linear Regression BIC: 57339.69563876513  
Polynomial Regression BIC: 57346.620360372595  
Model Comparison:  
 Model R² RMSE MAE AIC BIC  
Generalized Linear Regression 0.999101 1269.832252 439.383598 198674.93233 57339.695639  
 Polynomial Regression 0.999102 1269.217725 438.815563 198675.857315 57346.62036  
 Random Forest 0.972497 7022.572482 4380.591032 N/A N/A

### 8.1 Visualization Comparison

import matplotlib.pyplot as plt  
import seaborn as sns  
  
lr\_pd = lr\_predictions.select("SALARY", "prediction").toPandas()  
poly\_pd = poly\_predictions.select("SALARY", "prediction").toPandas()  
rf\_pd = rf\_predictions.select("SALARY", "prediction").toPandas()  
  
fig, axes = plt.subplots(2, 2, figsize=(15, 12))  
fig.suptitle('Model Predictions vs Actual Values', fontsize=16)  
  
# Generalized Linear Regression  
axes[0, 0].scatter(lr\_pd['SALARY'], lr\_pd['prediction'], alpha=0.6, color='blue')  
axes[0, 0].plot([lr\_pd['SALARY'].min(), lr\_pd['SALARY'].max()],  
 [lr\_pd['SALARY'].min(), lr\_pd['SALARY'].max()], 'r--', lw=2)  
axes[0, 0].set\_xlabel('Actual Salary')  
axes[0, 0].set\_ylabel('Predicted Salary')  
axes[0, 0].set\_title(f'Linear Regression (R² = {lr\_r2:.3f})')  
axes[0, 0].grid(True, alpha=0.3)  
  
# Polynomial Regression  
axes[0, 1].scatter(poly\_pd['SALARY'], poly\_pd['prediction'], alpha=0.6, color='green')  
axes[0, 1].plot([poly\_pd['SALARY'].min(), poly\_pd['SALARY'].max()],  
 [poly\_pd['SALARY'].min(), poly\_pd['SALARY'].max()], 'r--', lw=2)  
axes[0, 1].set\_xlabel('Actual Salary')  
axes[0, 1].set\_ylabel('Predicted Salary')  
axes[0, 1].set\_title(f'Polynomial Regression (R² = {poly\_r2:.3f})')  
axes[0, 1].grid(True, alpha=0.3)  
  
# Random Forest  
axes[1, 0].scatter(rf\_pd['SALARY'], rf\_pd['prediction'], alpha=0.6, color='orange')  
axes[1, 0].plot([rf\_pd['SALARY'].min(), rf\_pd['SALARY'].max()],  
 [rf\_pd['SALARY'].min(), rf\_pd['SALARY'].max()], 'r--', lw=2)  
axes[1, 0].set\_xlabel('Actual Salary')  
axes[1, 0].set\_ylabel('Predicted Salary')  
axes[1, 0].set\_title(f'Random Forest (R² = {r2:.3f})')  
axes[1, 0].grid(True, alpha=0.3)  
  
# Model comparison bar chart using Seaborn  
model\_names = ['Linear', 'Polynomial', 'Random Forest']  
r2\_scores = [lr\_r2, poly\_r2, r2]  
colors = ['blue', 'green', 'orange']  
comparison\_plot\_df = pd.DataFrame({  
 'Model': model\_names,  
 'R²': r2\_scores  
})  
  
sns.barplot(data=comparison\_plot\_df, x='R²', y='Model', palette=colors, ax=axes[1, 1])  
axes[1, 1].set\_xlabel('R² Score')  
axes[1, 1].set\_title('Model Performance Comparison (R²)')  
axes[1, 1].grid(True, alpha=0.3)  
  
plt.tight\_layout()  
plt.savefig("\_output/model\_comparison.png", dpi=300, bbox\_inches='tight')  
plt.show()  
plt.close()



### 8.2 Inference

Both the Generalized Linear Regression and Polynomial Regression models performed almost identically, with R² ≈ 0.9991 and very low errors (RMSE ≈ 1269, MAE ≈ 439). Their AIC and BIC values are also very close, showing no real advantage of adding polynomial terms. The Random Forest model explained less variation (R² ≈ 0.97) and had much higher errors (RMSE ≈ 7023, MAE ≈ 4381). Since AIC and BIC are not available for Random Forest, comparison is based only on accuracy metrics. Overall, the linear models clearly outperform Random Forest for this dataset. Between them, the simpler Generalized Linear Regression is preferable, as it achieves the same accuracy with lower complexity and easier interpretation.