Universidad Nacional de Ingeniería

Facultad de Ciencias

Escuela Profesional de Ciencias de la Computación

Practica Calificada 4

Estudiantes:

Penadillo Lazares Wenses Johan

Villarroel Lajo Gerald Takeshi

Curso:

CC531 Análisis en Macrodatos

Resumen

En el presente trabajo se realizará el análisis y exploración de datos a una base de datos de ofertas laborales de la plataforma de Linkedin para la carrera de ciencias de la computación, detallando cada paso con la finalidad de realizar consultas con Spark y Spark SQL con UDF usando el lenguaje de programación Scala, esto con la finalidad de poder analizar distintos puntos sobre la demanda laboral. Así mismo aplicar distintos modelos de Machine Learning sobre ciertos campos.

Palabras clave: Spark, Spark SQL, UDF, Scala, ML, MLlib.

Índice General

Índice de Figuras

Capítulo 1

Introducción

El presente trabajo tiene como propósito resolver la aplicación de consultas y análisis de datos mediante el uso del lenguaje de programación Scala, Spark y Spark SQL con UDF y MLlib. Para ello se usara una base de datos de ofertas de trabajo para la carrera de ciencias de la computación en la plataforma de Linkedin obtenido con técnicas de Webscraping.

Se abordarán consultas complejas aplicando distintas funciones de filtrado, agregación, selección, agrupamiento y ordenación con el fin de poder realizar un análisis sobre las tendencias de las ofertas laborales.

Capítulo 2

Marco Teórico

2.1. Apache Spark

Apache Spark es un motor unificado de analíticas para procesar datos a gran escala que integra módulos para SQL, streaming, aprendizaje automático y procesamiento de grafos. Spark se puede ejecutar de forma independiente o en Apache Hadoop, Apache Mesos, Kubernetes, la nube y distintas fuentes de datos [4].

2.1.1. Características:

- Velocidad: Spark puede ejecutar aplicaciones hasta 100 veces más rápido en memoria y 10 veces más rápido en disco que Hadoop.
- Facilidad de Uso: Cuenta con más de 80 operadores capaces de mejorar el desarrollo de aplicaciones en paralelo. Se puede utilizar de forma interactiva desde el shell de Scala, Python, R y SQL para escribir aplicaciones rápidamente.
- Integración con Herramientas: Se puede usar una pila de bibliotecas que incluye SQL, DataFrame, MLlib para aprendizaje automático, GraphX y Spark Streaming [7].

2.1.2. Componentes Principales:

- Spark Core: La base del proyecto Spark, que proporciona gestión de memoria, planificación de tareas, recuperación ante fallos, etc.
- Spark SQL: módulo de Spark que permite utilizar datos estructurados, se puede consultar datos estructurados de programas de Spark con SQL o con la API de DataFrame que resulte más cómoda.
- Spark Streaming: Dacilita la creación de soluciones de streaming escalables y tolerantes a fallos.

- MLlib: Biblioteca escalable de aprendizaje automático de Spark. Contiene numerosos algoritmos de aprendizaje de uso habitual, como clasificación, regresión, recomendación y agrupación en clústeres.
- GraphX: API de Spark para grafos y computación en paralelo de grafos. Es flexible y funciona a la perfección tanto con grafos como con colecciones [7].

2.2. Spark SQL

2.2.1. Características:

- Consultas SQL: Permite ejecutar consultas SQL en Spark.
- DataFrames: Proporciona una abstracción de datos estructurados similar a una tabla en una base de datos relacional.
- Compatibilidad con Hive: Soporta consultas HiveQL y se puede integrar con el metastore de Hive [3].

NOTA: ***(Uso en el Proyecto)***

- Consulta de Datos: Para ejecutar consultas SQL en los datos cargados desde un archivo CSV o una base de datos PostgreSQL.
- Transformaciones: Para realizar transformaciones en los datos utilizando DataFrames y Datasets.

2.3. Funciones Definidas por el Usuario

También conocido por sus siglas en inglés UDF (User-Defined Functions), son funciones definidas por el usuario que permite reutilizar la lógica personalizada en el entorno del usuario.
Un beneficio importante de las UDF es que permiten a los usuarios expresar la lógica en lenguajes familiares, lo que reduce el costo humano asociado con la refactorización del código. Para consultas ad hoc, limpieza manual de datos, análisis de datos exploratorios y la mayoría de las operaciones en conjuntos de datos pequeños o medianos, es poco probable que los costos generales de latencia asociados con las UDF superen los costos asociados con la refactorización del código [6].
NOTA: ***(Uso en el Proyecto)***

- Personalización: Permite generar piezas de código personalizadas que no está disponible en las funciones integradas de Spark. Por ejemplo, cuando se filtran los trabajos que requieren una determinada habilidad y utilizan un conjunto específico de lenguajes de programación.

2.4. Machine Learning con Spark MLlib

MLlib es la biblioteca de aprendizaje automático (ML) de Spark. Su objetivo es hacer que el aprendizaje automático práctico sea escalable y sencillo. Entre las herramientas que proporciona tenemos: [1]

- Algoritmos de aprendizaje automático: algoritmos de aprendizaje comunes como clasificación, regresión, agrupación y filtrado colaborativo.
- Caracterización: extracción de características, transformación, reducción de dimensionalidad y selección.
- Pipelines: herramientas para construir, evaluar y ajustar ML Pipelines
- Persistencia: guardar y cargar algoritmos, modelos y Pipelines
- Utilidades: álgebra lineal, estadística, manejo de datos, etc.

2.5. Conjunto de datos distribuidos resilientes

También conocido por sus siglas en inglés RDD (Resilient Distributed Dataset), es una colección de elementos particionados a través de los nodos del clúster que pueden operarse en paralelo. Los RDD se crean a partir de un archivo en el sistema de archivos de Hadoop (o cualquier otro sistema de archivos compatible con Hadoop) o una colección existente de Scala en el programa driver, y se transforman [2].
NOTA: ***(Uso en el Proyecto)***

- Transformaciones y Acciones: Los RDDs soportan dos tipos de operaciones: transformaciones (e.g., map, filter, reduceByKey) y acciones (e.g., collect, count, saveAsTextFile).
- Persistencia y Caché: Los RDDs pueden ser persistidos en memoria o disco para optimizar la reutilización de datos en múltiples operaciones.

Poseen un mecanismo de tolerancia a fallos, donde los RDDs pueden ser reconstruidos automáticamente a partir de su linaje (histórico de transformaciones) en caso de que una partición del RDD se pierda.

Capítulo 3

Metodología

3.1. Marco Teórico

Establecimos el marco teórico anterior basándonos en los conceptos necesarios y requeridos para el desarrollo de las consultas y sus respectivos gráficos, proporcionándonos un enfoque conceptual para abordar el problema de investigación.

3.2. Herramientas y Tecnologías

Se utilizó el lenguaje de programación Python para la limpieza de los datos. Se uso el lenguaje de programación Scala junto a Spark, Spark SQL con UDF y MLlib para el desarrollo de las consultas. La base de datos usada esta relacionada con datos relevates de ofertas de trabajo de la plataforma de Linkedin que fue extraída con técnicas de WebScraping.

3.3. Métodos de Investigación

En este estudio, aplicamos un enfoque experimental y aplicado para analizar el dataset de descripciones de propuestas de trabajo para postulantes de la carrera de Ciencia de la Computación obtenida mediante WebScraping. Se realizó una exploración de los datos haciendo uso de Python y sus respectivas librerías, como Pandas para la manipulación y análisis de datos en estructuras tabulares, Matplotlib y Seaborn para la creación de visualizaciones gráficas, NLTK (Natural Language Toolkit) para tareas de procesamiento de lenguaje natural, Counter para contar elementos en listas y otras colecciones, WordCloud para generar nubes de palabras visuales, y el módulo ***re*** para trabajar con expresiones regulares y filtrar datos textuales. Esta metodología nos ayudó a aplicar conocimientos de procesamiento de lenguaje natural (PLN) y visualización de datos para resolver las distintas consultas generadas a partir de la base de datos obtenida.

3.4. Validación de Resultados

Los resultados obtenidos que en el Capítulo 5 se muestran, son validados mediante la comparación con los datos obtenidos de las consultas con Spark Scala y las obtenidas mediante Spark SQL usando las fórmulas creadas a partir del usuario (UDF).

Capítulo 4

Bases de datos

En esta sección se explican los detalles de los campos de la base de datos, el proceso de limpieza y exploración.

4.1. Obtención y limpieza de base de datos

Para el presente trabajo usaremos la base de datos [5], la cual fue creada extrayendo datos de ofertas de trabajo de Linkedin usando técnicas de Webscraping.
Descripción de campos relevantes de las bases de datos:

- **job_title:** Titulo de la oferta publicada.
- **job_location:** Ubicación del puesto de trabajo.
- **search_city:** Ciudad usada en la consulta de la búsqueda del trabajo.
- **search_country:** País usada en la consulta de la búsqueda del trabajo.
- **job_level:** Nivel de trabajo pedido para la posición de trabajo.
- **job_type:** Tipo de empleo ofrecido por la oferta de trabajo.
- **job_summary:** Descripción de la oferta de trabajo.
- **job_skills:** Skills solicitadas para la oferta del puesto de trabajo.

El proceso de limpieza consintió en una serie de pasos donde se aplico expresiones regulares, técnicas de NLP y filtrado de palabras clave:

- Reconocimiento de columnas útiles para el desarrollo del presente trabajo (job_title, job_summary, job_skills).
- Eliminación de columnas no útiles para este trabajo.

- Eliminación de caracteres no alfanuméricos del alfabeto ingles.
- Normalización del texto (convertir mayúsculas a minúsculas).
- Eliminar *stopwords* que no agregan información relevante.
- Lematización para reducir el numero de palabras del texto.
- Filtrado de palabras clave para extraer información relevante y crear nuevos campos.
- Guardar todos los registros filtrados en un archivo csv que luego fue almacenado en una base de datos.

4.2. Exploración de datos

El proceso de exploración consistió en una exploración visual de campos relevantes.

Inspeccionamos las palabras mas frecuentes del campo *job_title* como podemos ver en la Figura 4.1.

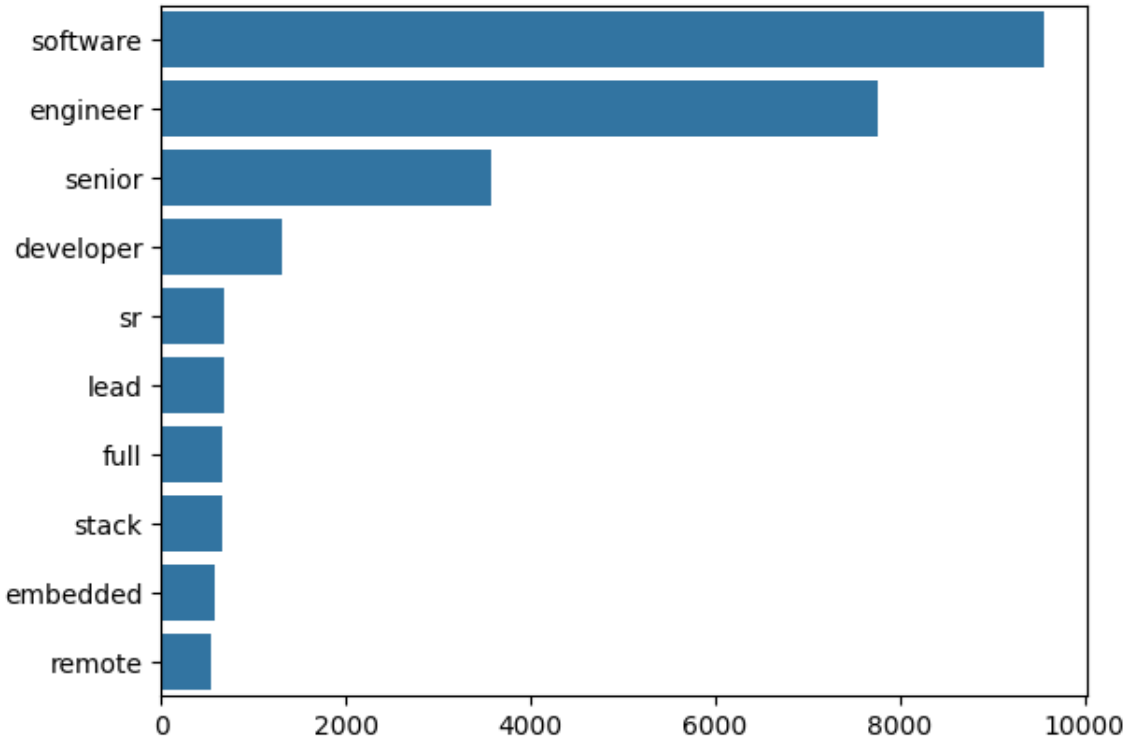


Figura 4.1: Palabras mas frecuentes en el campo job_title.

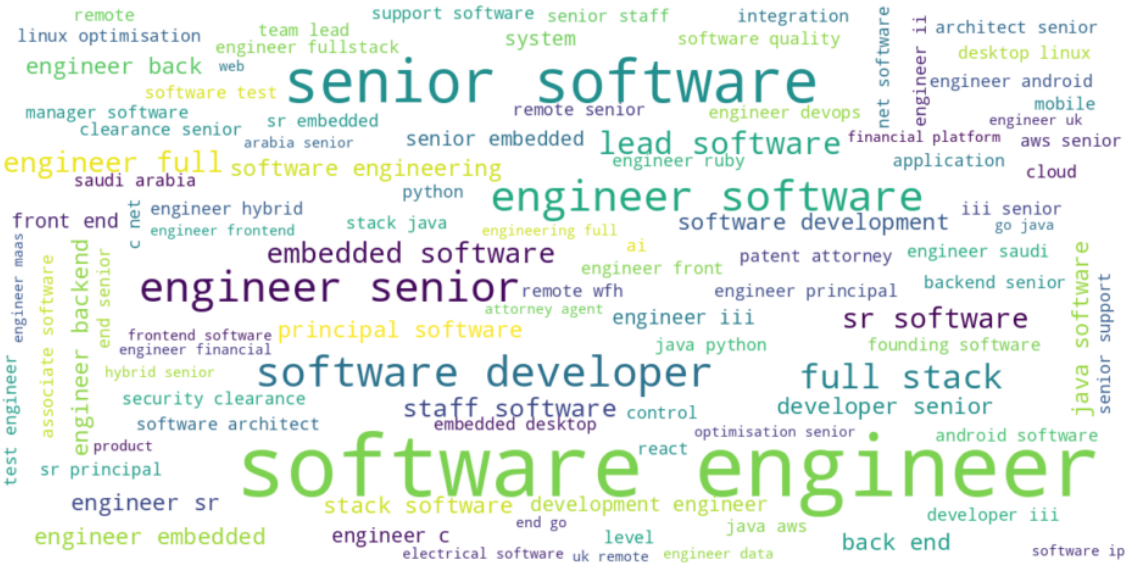


Figura 4.2: Nube de palabras del campo job_title.

Inspeccionamos las palabras mas frecuentes del campo *job_summary* como podemos ver en la Figura 4.3.

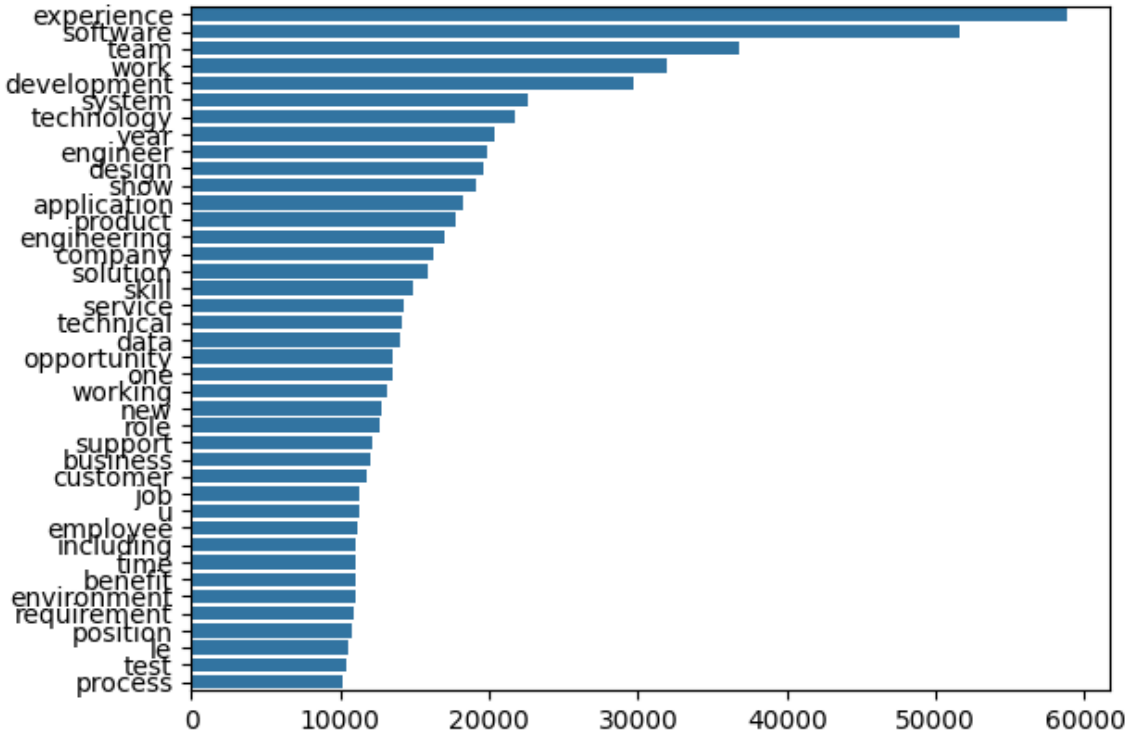


Figura 4.3: Palabras mas frecuentes en el campo job_summary.

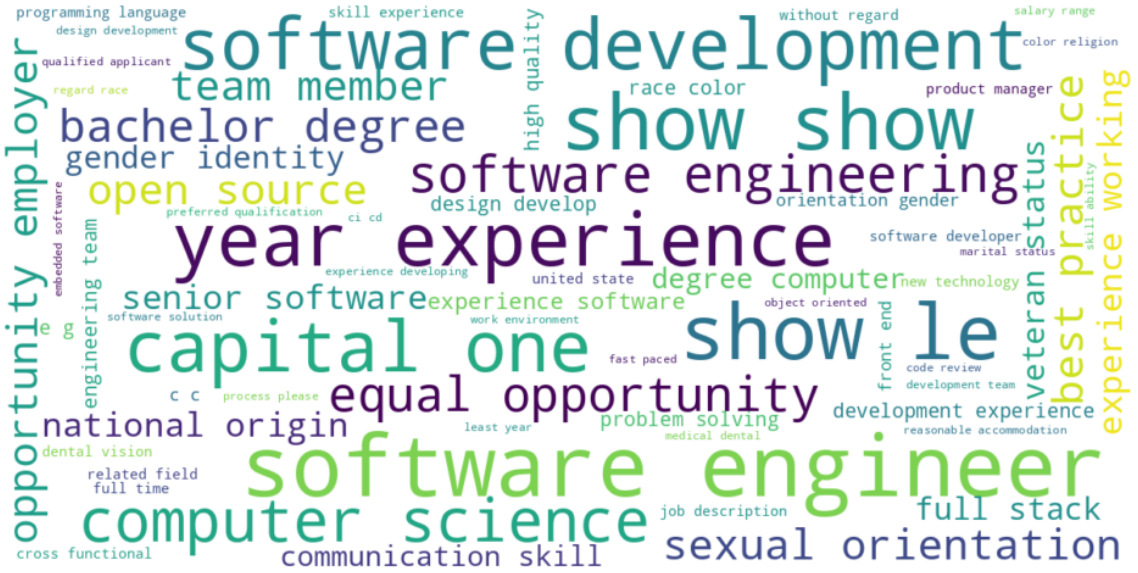


Figura 4.4: Nube de palabras del campo job_summary.

Inspeccionamos las palabras mas frecuentes del campo *job_skills* como podemos ver en la Figura 4.5.

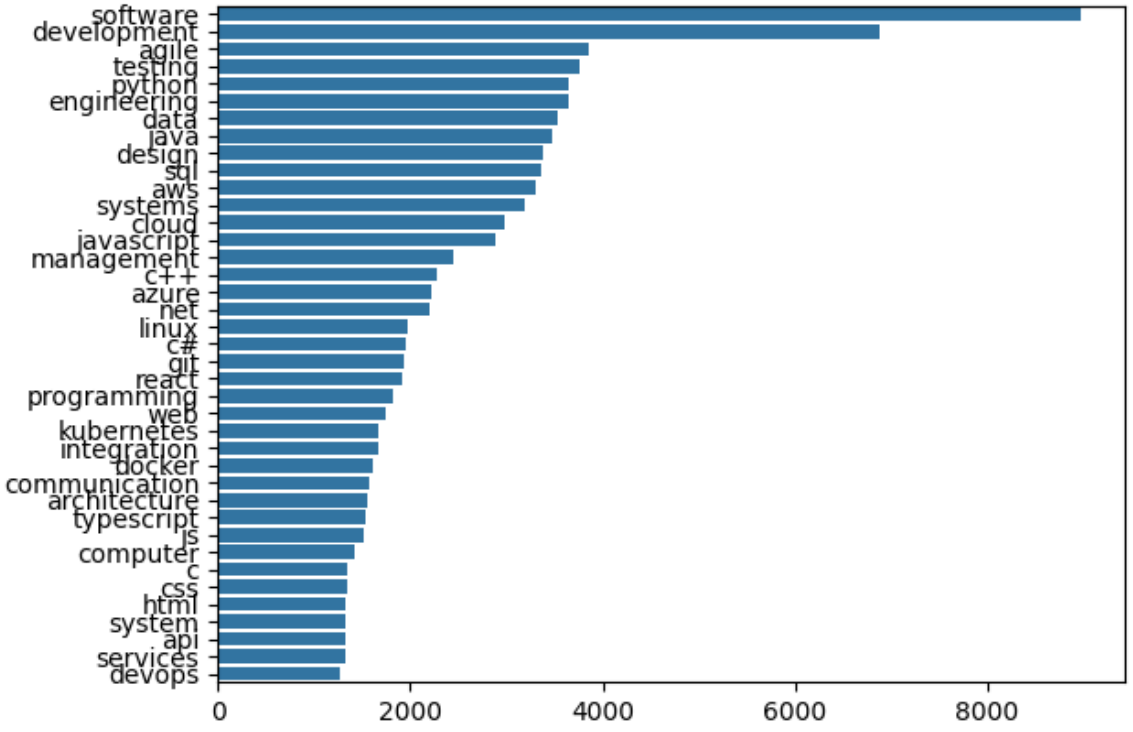


Figura 4.5: Palabras mas frecuentes en el campo job_skills.

Figura 4.6: Nube de palabras del campo job_skills.

Luego de la inspección visual podemos determinar las palabras claves a usar para el filtrado y extracción de datos.

Capítulo 5

Desarrollo, Resultados y Discusión

En esta sección se detallan los pasos realizados durante el desarrollo de este trabajo de investigación.

5.1. Consultas Spark y Spark SQL con UDF

Para realizar las consultas se uso Spark, Spark SQL con UDF y el lenguaje de programación Scala. Para ejecutar

5.1.1. Consulta 1

Mostrar trabajos que requieren Teamwork como habilidad blanda y utilizan los lenguajes de programación Python y Java.

```
+--------------------+--------------------+--------------------+
|            job_name|programming_languages|         soft_skills|
+--------------------+--------------------+--------------------+
|BackEnd Software  ...| Java-Go-SQL-Scala ...|            Teamwork|
|Full Stack Softwa ...| TypeScript-Java-G ...|            Teamwork|
|Full Stack Softwa ...| Java-JavaScript-P ...|            Teamwork|
|Full Stack Softwa ...| TypeScript-Java-C ...|            Teamwork|
|   Software Engineer| Java-TypeScript-P ...|Problem Solving-O ... |
|Quality Software  ...|      Java-SQL-Python|            Teamwork|
|BackEnd Software  ...| Java-Go-SQL-HTML- ...|            Teamwork|
|BackEnd Software  ...| Java-Go-SQL-Scala ...|Organization-Team ... |
|BackEnd Software  ...| Java-Go-SQL-Scala ...| Teamwork-Creativity|
|Full Stack Softwa ...| TypeScript-Java-C ...|Problem Solving-T ... |
|Full Stack Softwa ...| TypeScript-Java-G ...| Teamwork-Creativity|
|Full Stack Softwa ...| TypeScript-Java-C ...|            Teamwork|
|Full Stack Softwa ...| TypeScript-Java-C ...|Organization-Team ... |
|BackEnd Software  ...| Java-Go-SQL-Scala ...|            Teamwork|
|BackEnd Software  ...| Java-Go-SQL-Scala ...| Teamwork-Creativity|
|Full Stack Softwa ...| TypeScript-Java-C ...|            Teamwork|
|   Software Engineer| Java-Go-SQL-Scala ...|            Teamwork|
|  Software Developer| Java-CSS-HTML-Jav ...|Problem Solving-L ... |
|Full Stack Softwa ...|    JavaScript-Python|Communication-Tea ... |
|   Software Engineer|          Java-Python|Communication-Tea ... |
+--------------------+--------------------+--------------------+
only showing top 20 rows

Execution time: 1681 milliseconds
```

Figura 5.1: Respuesta de la query 1 con Spark y Scala.

```
+--------------------+--------------------+--------------------+
|            job_name|programming_languages|         soft_skills|
+--------------------+--------------------+--------------------+
|BackEnd Software  ...| Java-Go-SQL-Scala ...|            Teamwork|
|Full Stack Softwa ...| TypeScript-Java-G ...|            Teamwork|
|Full Stack Softwa ...| Java-JavaScript-P ...|            Teamwork|
|Full Stack Softwa ...| TypeScript-Java-C ...|            Teamwork|
|   Software Engineer| Java-TypeScript-P ...|Problem Solving-O ... |
|Quality Software  ...|      Java-SQL-Python|            Teamwork|
|BackEnd Software  ...| Java-Go-SQL-HTML- ...|            Teamwork|
|BackEnd Software  ...| Java-Go-SQL-Scala ...|Organization-Team ... |
|BackEnd Software  ...| Java-Go-SQL-Scala ...| Teamwork-Creativity|
|Full Stack Softwa ...| TypeScript-Java-C ...|Problem Solving-T ... |
|Full Stack Softwa ...| TypeScript-Java-G ...| Teamwork-Creativity|
|Full Stack Softwa ...| TypeScript-Java-C ...|            Teamwork|
|Full Stack Softwa ...| TypeScript-Java-C ...|Organization-Team ... |
|BackEnd Software  ...| Java-Go-SQL-Scala ...|            Teamwork|
|BackEnd Software  ...| Java-Go-SQL-Scala ...| Teamwork-Creativity|
|Full Stack Softwa ...| TypeScript-Java-C ...|            Teamwork|
|   Software Engineer| Java-Go-SQL-Scala ...|            Teamwork|
|  Software Developer| Java-CSS-HTML-Jav ...|Problem Solving-L ... |
|Full Stack Softwa ...|    JavaScript-Python|Communication-Tea ... |
|   Software Engineer|          Java-Python|Communication-Tea ... |
+--------------------+--------------------+--------------------+
only showing top 20 rows

Execution time: 1661 milliseconds
```

Figura 5.2: Respuesta de la query 1 con Spark SQL con UDF.

En la Figura 5.1 y Figura 5.2 podemos ver la respuesta de la consulta. En cuanto a tiempos de ejecución en este caso con UDF es ligeramente menor.

5.1.2. Consulta 2

Mostrar los trabajos que solicitan en Cincinnati con algún tipo de Nivel Académico.

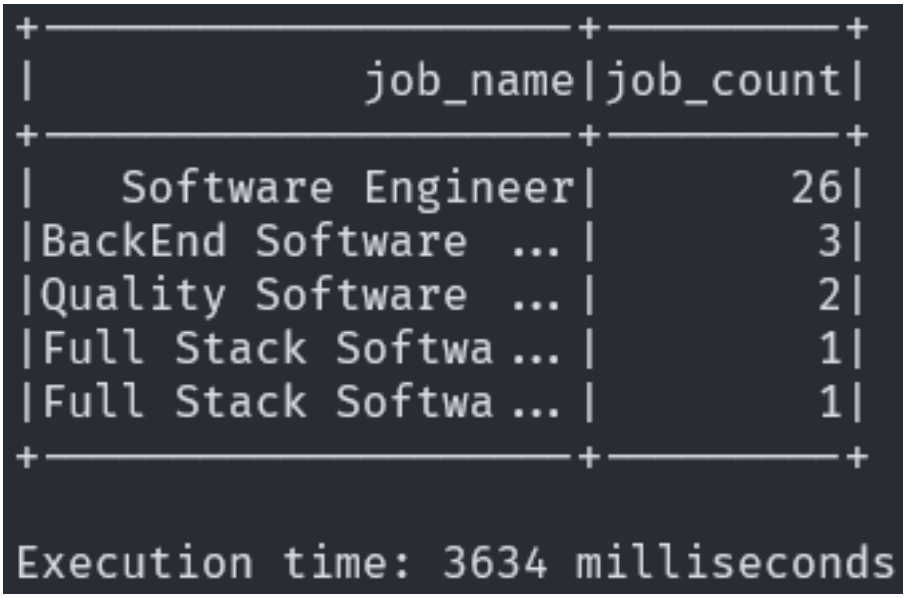


Figura 5.3: Respuesta de la query 2 con Spark y Scala.

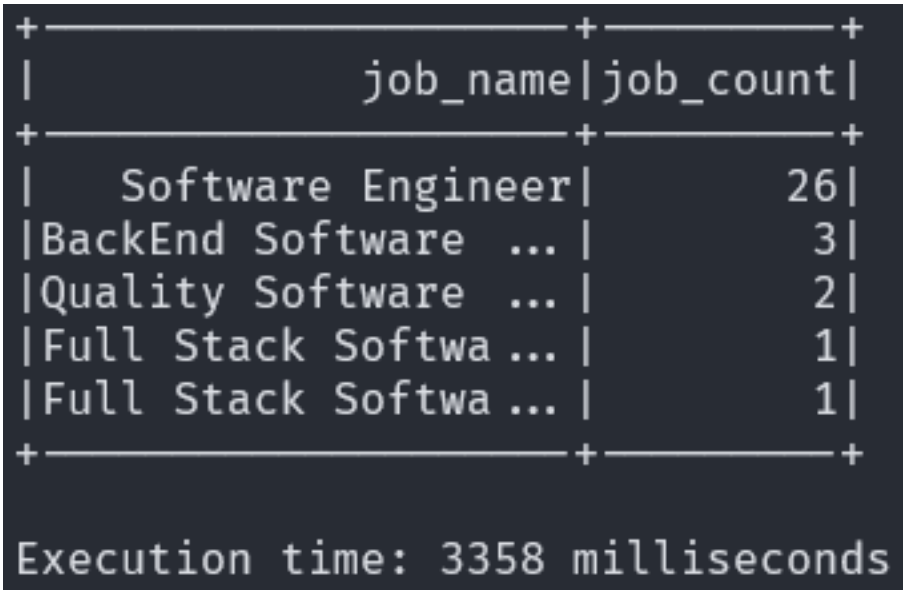


Figura 5.4: Respuesta de la query 2 con Spark SQL con UDF.

En la Figura 5.3 y Figura 5.4 podemos ver la respuesta de la consulta. En cuanto a tiempos de ejecución en este caso con UDF es ligeramente menor.

5.1.3. Consulta 3

Mostrar los trabajos que requieren Web Development como conocimiento y una certificación con la palabra clave “Certified”.

```
+--------------------+--------------------+--------------------+
|            job_name|           knowledge|      certifications|
+--------------------+--------------------+--------------------+
|   Software Engineer|Security-Version  ...|Certified Kuberne...|
|  Software Developer|Testing-Databases ...|Salesforce Certif...|
|  Software Developer|Testing-Security- ...|Certified Informa...|
|   Software Engineer|Testing-DevOps-Ve ...|Certified ScrumMa...|
|   Software Engineer|Testing-DevOps-Ve ...|Certified Informa...|
|  Software Architect|DevOps-Security-D ...|Certified Informa...|
|Full Stack Softwa ...|Testing-DevOps-Ve ...|Certified Informa...|
|   Software Engineer|Testing-Security- ...|Certified Ethical ...|
|  Software Developer|Testing-Security- ...|Certified Informa...|
|   Software Engineer|DevOps-Databases- ...|Salesforce Certif...|
|  Software Developer|Agile-Web Develop ...|Certified ScrumMa...|
|  Software Developer|Agile-Web Develop ...|Certified ScrumMa...|
|  Software Developer|DevOps-Security-P ...|Certified Informa...|
|  Software Developer|Testing-DevOps-Da ...|Certified ScrumMa...|
|  Software Developer|Agile-DevOps-Web  ...|Certified Informa...|
|   Software Engineer|Security-Mobile D ...|Certified ScrumMa...|
|  Software Developer|Testing-DevOps-Se ...|Cisco Certified N...|
|   Software Engineer|Testing-Web Devel ...|Certified ScrumMa...|
+--------------------+--------------------+--------------------+

Execution time: 1833 milliseconds
```

Figura 5.5: Respuesta de la query 3 con Spark y Scala.

```
+--------------------+--------------------+--------------------+
|            job_name|           knowledge|      certifications|
+--------------------+--------------------+--------------------+
|   Software Engineer|Security-Version  ...|Certified Kuberne...|
|  Software Developer|Testing-Databases ...|Salesforce Certif...|
|  Software Developer|Testing-Security- ...|Certified Informa...|
|   Software Engineer|Testing-DevOps-Ve ...|Certified ScrumMa...|
|   Software Engineer|Testing-DevOps-Ve ...|Certified Informa...|
|  Software Architect|DevOps-Security-D ...|Certified Informa...|
|Full Stack Softwa ...|Testing-DevOps-Ve ...|Certified Informa...|
|   Software Engineer|Testing-Security- ...|Certified Ethical ...|
|  Software Developer|Testing-Security- ...|Certified Informa...|
|   Software Engineer|DevOps-Databases- ...|Salesforce Certif...|
|  Software Developer|Agile-Web Develop ...|Certified ScrumMa...|
|  Software Developer|Agile-Web Develop ...|Certified ScrumMa...|
|  Software Developer|DevOps-Security-P ...|Certified Informa...|
|  Software Developer|Testing-DevOps-Da ...|Certified ScrumMa...|
|  Software Developer|Agile-DevOps-Web  ...|Certified Informa...|
|   Software Engineer|Security-Mobile D ...|Certified ScrumMa...|
|  Software Developer|Testing-DevOps-Se ...|Cisco Certified N...|
|   Software Engineer|Testing-Web Devel ...|Certified ScrumMa...|
+--------------------+--------------------+--------------------+

Execution time: 1833 milliseconds
```

Figura 5.6: Respuesta de la query 3 con Spark SQL con UDF.

En la Figura 5.5 y Figura 5.6 podemos ver la respuesta de la consulta. En cuanto a tiempos de ejecución en este caso vemos que son similares.

5.1.4. Consulta 4

Mostrar el promedio de habilidades blandas requerido por nivel de trabajo.

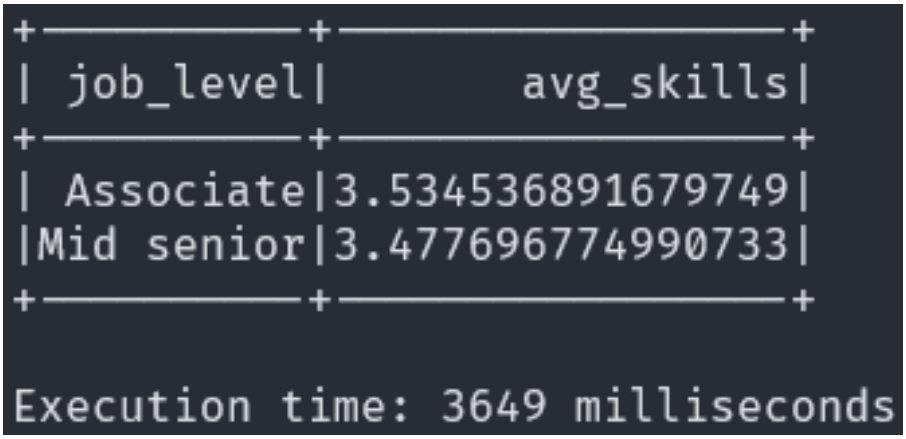


Figura 5.7: Respuesta de la query 4 con Spark.

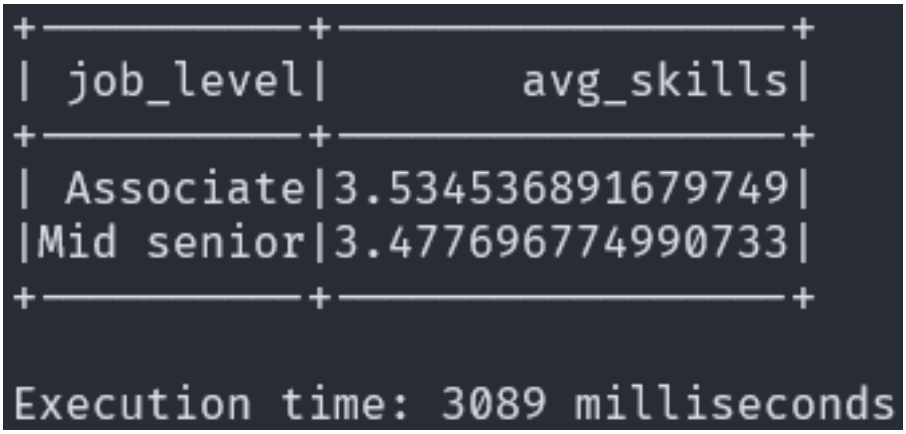
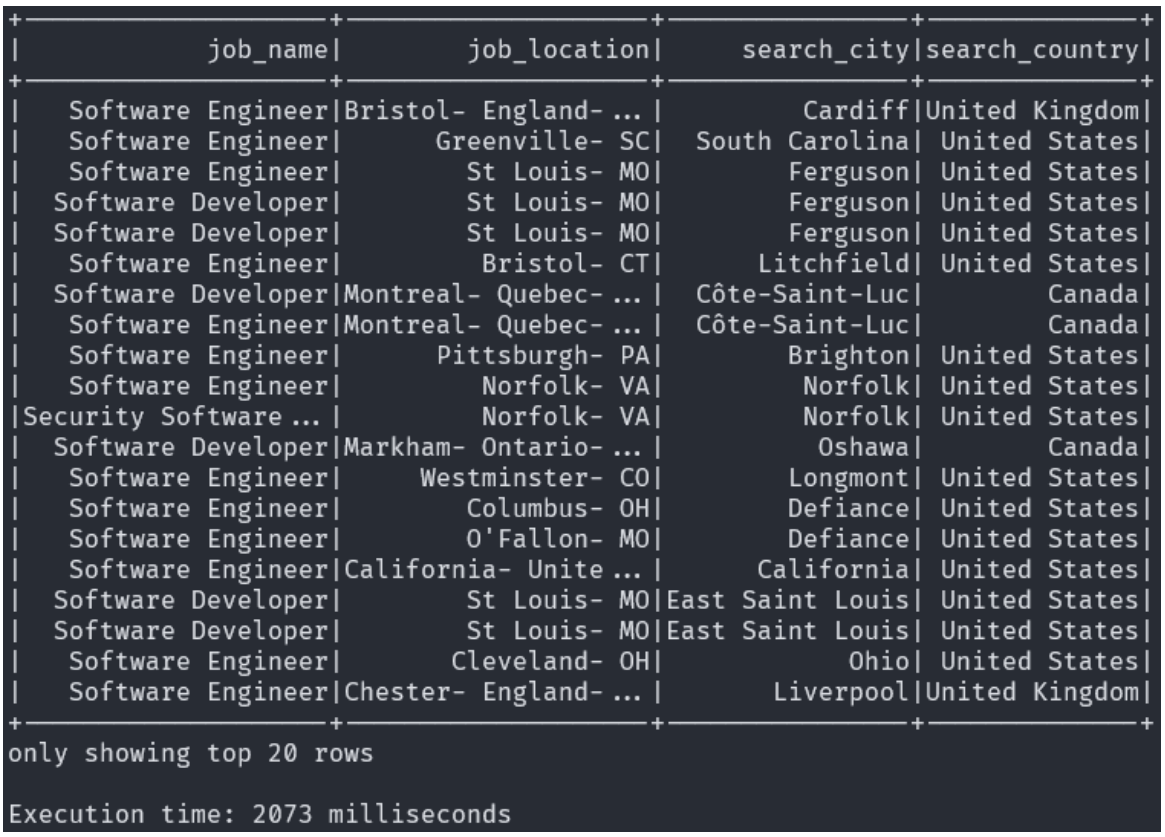


Figura 5.8: Respuesta de la query 4 con Spark SQL con UDF.

En la Figura 5.7 y Figura 5.8 podemos ver la respuesta de la consulta. En cuanto a tiempos de ejecución en este caso con UDF es ligeramente menor.

5.1.5. Consulta 5

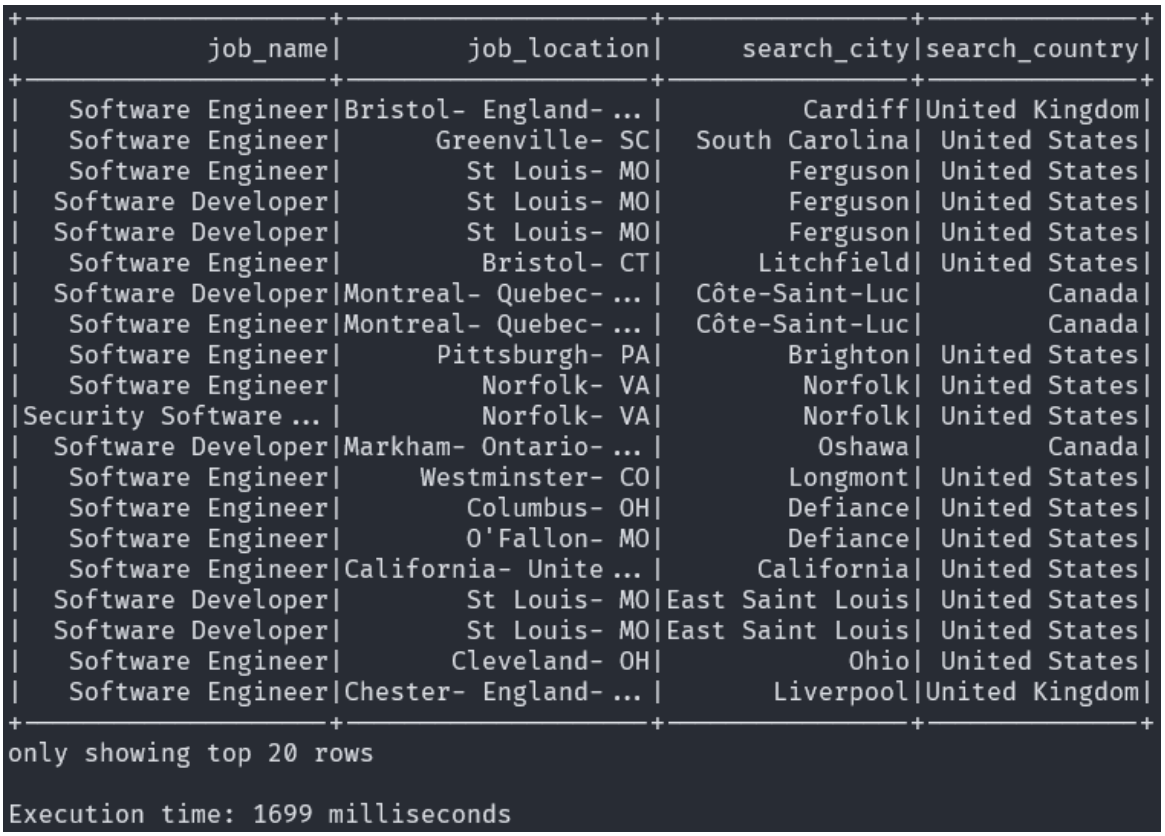
Mostrar los trabajos que requieren conocimientos en Agile y un grado académico de Bachelor's.



```
+--------------------+--------------------+-----------------+---------------+
|            job_name|        job_location|     search_city|search_country|
+--------------------+--------------------+-----------------+---------------+
|   Software Engineer|Bristol- England-...|         Cardiff|United Kingdom|
|   Software Engineer|      Greenville- SC|  South Carolina| United States|
|   Software Engineer|        St Louis- MO|         Ferguson| United States|
|  Software Developer|        St Louis- MO|         Ferguson| United States|
|  Software Developer|        St Louis- MO|         Ferguson| United States|
|   Software Engineer|         Bristol- CT|       Litchfield| United States|
|  Software Developer|Montreal- Quebec-...|  Côte-Saint-Luc|         Canada|
|   Software Engineer|Montreal- Quebec-...|  Côte-Saint-Luc|         Canada|
|   Software Engineer|      Pittsburgh- PA|         Brighton| United States|
|   Software Engineer|         Norfolk- VA|          Norfolk| United States|
|Security Software...|         Norfolk- VA|          Norfolk| United States|
|  Software Developer|Markham- Ontario-...|          Oshawa|         Canada|
|   Software Engineer|     Westminster- CO|         Longmont| United States|
|   Software Engineer|        Columbus- OH|         Defiance| United States|
|   Software Engineer|        O'Fallon- MO|         Defiance| United States|
|   Software Engineer|California- Unite...|       California| United States|
|  Software Developer|        St Louis- MO|East Saint Louis| United States|
|  Software Developer|        St Louis- MO|East Saint Louis| United States|
|   Software Engineer|       Cleveland- OH|             Ohio| United States|
|   Software Engineer|Chester- England-...|        Liverpool|United Kingdom|
+--------------------+--------------------+-----------------+---------------+
only showing top 20 rows

Execution time: 2073 milliseconds
```

Figura 5.9: Respuesta de la query 5 con Spark.



```
+--------------------+--------------------+-----------------+---------------+
|            job_name|        job_location|     search_city|search_country|
+--------------------+--------------------+-----------------+---------------+
|   Software Engineer|Bristol- England-...|         Cardiff|United Kingdom|
|   Software Engineer|      Greenville- SC|  South Carolina| United States|
|   Software Engineer|        St Louis- MO|         Ferguson| United States|
|  Software Developer|        St Louis- MO|         Ferguson| United States|
|  Software Developer|        St Louis- MO|         Ferguson| United States|
|   Software Engineer|         Bristol- CT|       Litchfield| United States|
|  Software Developer|Montreal- Quebec-...|  Côte-Saint-Luc|         Canada|
|   Software Engineer|Montreal- Quebec-...|  Côte-Saint-Luc|         Canada|
|   Software Engineer|      Pittsburgh- PA|         Brighton| United States|
|   Software Engineer|         Norfolk- VA|          Norfolk| United States|
|Security Software...|         Norfolk- VA|          Norfolk| United States|
|  Software Developer|Markham- Ontario-...|          Oshawa|         Canada|
|   Software Engineer|     Westminster- CO|         Longmont| United States|
|   Software Engineer|        Columbus- OH|         Defiance| United States|
|   Software Engineer|        O'Fallon- MO|         Defiance| United States|
|   Software Engineer|California- Unite...|       California| United States|
|  Software Developer|        St Louis- MO|East Saint Louis| United States|
|  Software Developer|        St Louis- MO|East Saint Louis| United States|
|   Software Engineer|       Cleveland- OH|             Ohio| United States|
|   Software Engineer|Chester- England-...|        Liverpool|United Kingdom|
+--------------------+--------------------+-----------------+---------------+
only showing top 20 rows

Execution time: 1699 milliseconds
```

Figura 5.10: Respuesta de la query 5 con Spark SQL con UDF.

En la Figura 5.9 y Figura 5.10 podemos ver la respuesta de la consulta. En cuanto a tiempos de ejecución en este caso con UDF es menor.

5.1.6. Consulta 6

Mostrar el lenguaje de programación más solicitado para trabajos presenciales.

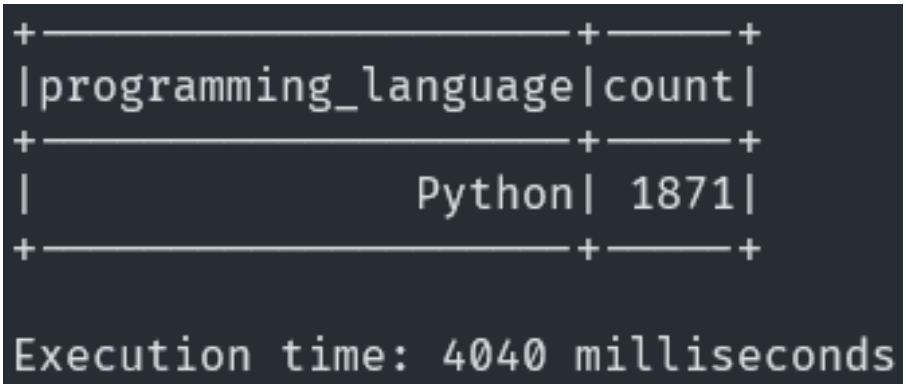


Figura 5.11: Respuesta de la query 6 con Spark.

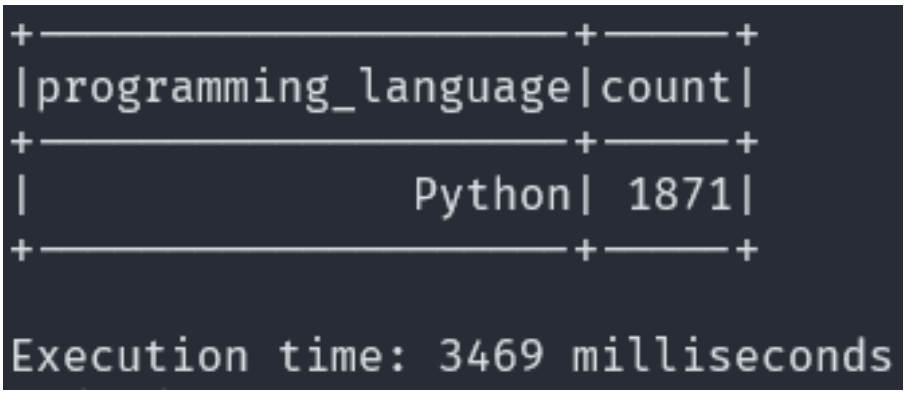
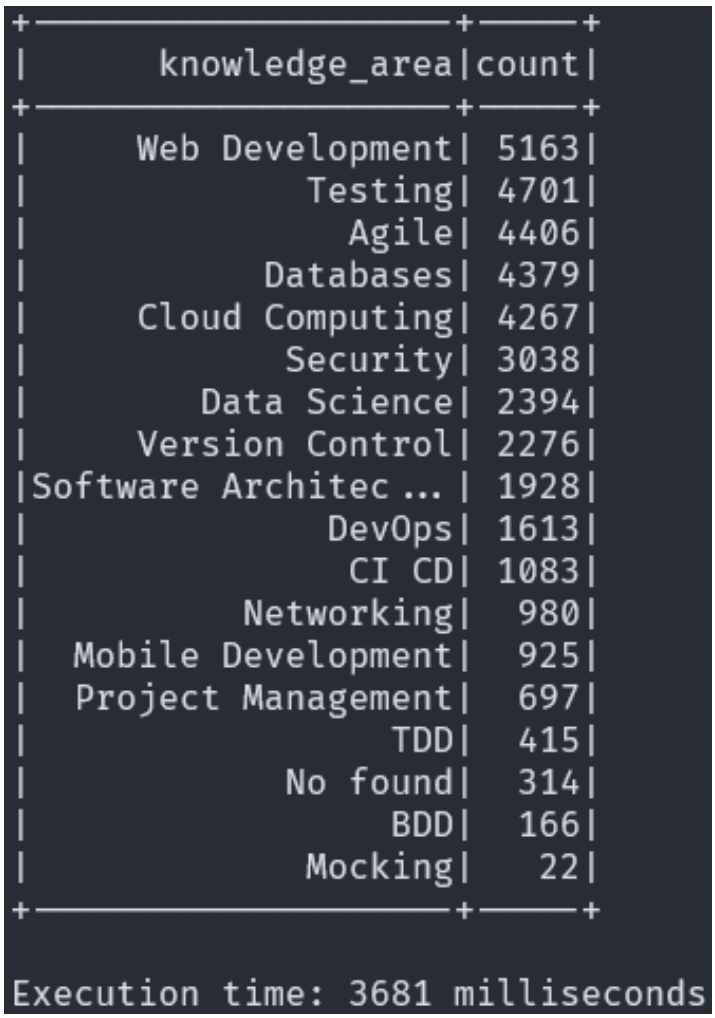


Figura 5.12: Respuesta de la query 6 con Spark SQL con UDF.

En la Figura 5.11 y Figura 5.12 podemos ver la respuesta de la consulta. En cuanto a tiempos de ejecución en este caso con UDF es ligeramente menor.

5.1.7. Consulta 7

Mostrar el numero de trabajos donde solicitan cada área de conocimiento.



knowledge_area	count
Web Development	5163
Testing	4701
Agile	4406
Databases	4379
Cloud Computing	4267
Security	3038
Data Science	2394
Version Control	2276
Software Architec ...	1928
DevOps	1613
CI CD	1083
Networking	980
Mobile Development	925
Project Management	697
TDD	415
No found	314
BDD	166
Mocking	22

Execution time: 3681 milliseconds

Figura 5.13: Respuesta de la query 7 con Spark.

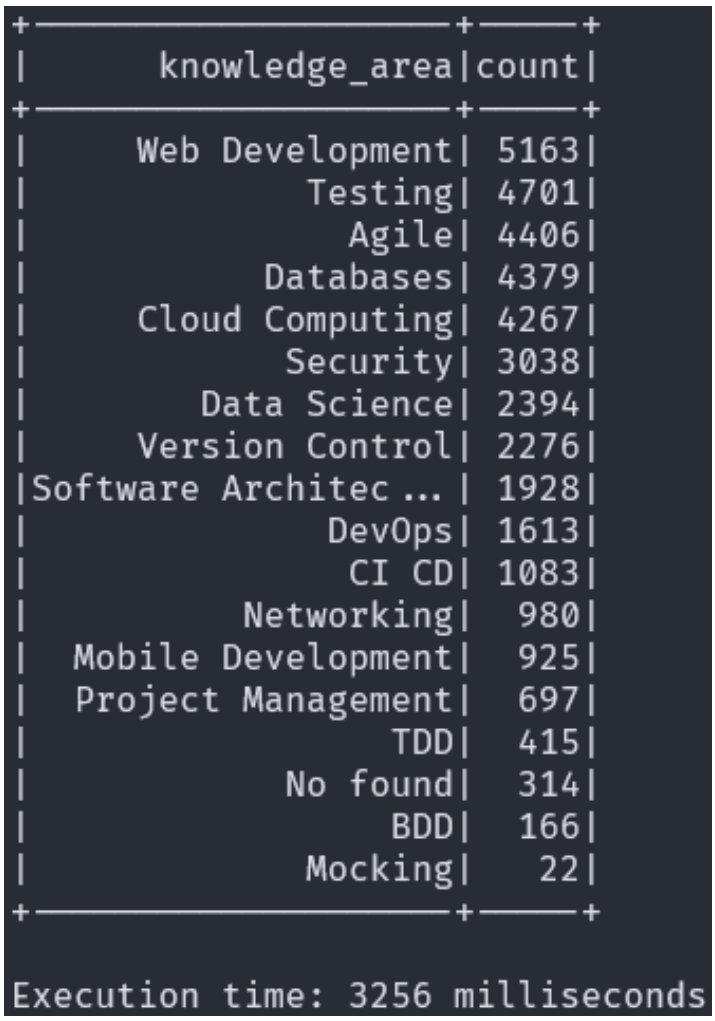


Figura 5.14: Respuesta de la query 7 con Spark SQL con UDF.

En la Figura 5.13 y Figura 5.14 podemos ver la respuesta de la consulta. En cuanto a tiempos de ejecución en este caso con UDF es ligeramente menor.

5.2. Consultas con Spark MLlib

Aplicamos modelos de Machine Learning con ayuda de la librería MLlib de Spark.

```
root
 ├── job_location: string (nullable = true)
 ├── search_city: string (nullable = true)
 ├── search_country: string (nullable = true)
 ├── job_level: string (nullable = true)
 ├── job_type: string (nullable = true)
 ├── job_name: string (nullable = true)
 ├── languages: string (nullable = true)
 ├── certifications: string (nullable = true)
 ├── soft_skills: string (nullable = true)
 ├── programming_languages: string (nullable = true)
 ├── technologies: string (nullable = true)
 ├── academic_degrees: string (nullable = true)
 ├── knowledge: string (nullable = true)
 ├── n_languages: double (nullable = false)
 ├── n_certifications: double (nullable = false)
 ├── n_soft_skills: double (nullable = false)
 ├── n_programming_languages: double (nullable = false)
 ├── n_technologies: double (nullable = false)
 ├── n_knowledge: double (nullable = false)
 ├── n_academic_degrees: double (nullable = false)
 ├── num_job_level: double (nullable = false)
 ├── num_job_type: double (nullable = false)
 ├── num_search_country: double (nullable = false)
```

Figura 5.15: Esquema de datos de entrada.

```
Training Data count: 7468
Test Data count: 1899
```

Figura 5.16: Creando el train y test data 80 % y 20 %.

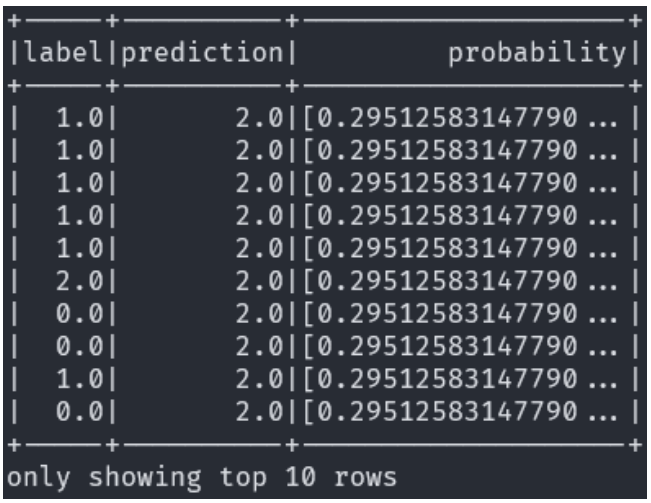
5.2.1. Consulta ML 1

Aplicamos el modelo de LogisticRegression con múltiples clases tomando como etiqueta *num_job_type*.

num_job_type	n_languages	n_certifications	n_soft_skills	n_programming_languages	n_technologies	n_knowledge	n_academic_degrees
1.0	0.0	0.0	2.0	4.0	0.0	3.0	118.0
1.0	0.0	0.0	2.0	0.0	0.0	1.0	110.0
2.0	0.0	0.0	1.0	6.0	4.0	4.0	116.0
2.0	0.0	0.0	1.0	6.0	3.0	5.0	116.0
2.0	0.0	0.0	2.0	6.0	8.0	5.0	116.0
2.0	0.0	0.0	5.0	0.0	0.0	2.0	116.0
2.0	0.0	0.0	1.0	3.0	5.0	3.0	116.0
2.0	0.0	0.0	1.0	8.0	9.0	5.0	116.0
2.0	0.0	0.0	5.0	1.0	0.0	3.0	67.0
2.0	0.0	0.0	5.0	1.0	0.0	3.0	67.0

only showing top 10 rows

Figura 5.17: Los primeros 10 registros de los datos de entrada.



```
+-----+----------+--------------------+
|label|prediction|         probability|
+-----+----------+--------------------+
|  1.0|       2.0|[0.29512583147790 ... |
|  1.0|       2.0|[0.29512583147790 ... |
|  1.0|       2.0|[0.29512583147790 ... |
|  1.0|       2.0|[0.29512583147790 ... |
|  1.0|       2.0|[0.29512583147790 ... |
|  2.0|       2.0|[0.29512583147790 ... |
|  0.0|       2.0|[0.29512583147790 ... |
|  0.0|       2.0|[0.29512583147790 ... |
|  1.0|       2.0|[0.29512583147790 ... |
|  0.0|       2.0|[0.29512583147790 ... |
+-----+----------+--------------------+
only showing top 10 rows
```

Figura 5.18: Predicciones de los 10 primeros registros.

```
False positive rate by label:
label 0: 0.0
label 1: 0.0
label 2: 1.0
True positive rate by label:
label 0: 0.0
label 1: 0.0
label 2: 1.0
Precision by label:
label 0: 0.0
label 1: 0.0
label 2: 0.46063202999464 38
Recall by label:
label 0: 0.0
label 1: 0.0
label 2: 1.0
F-measure by label:
label 0: 0.0
label 1: 0.0
label 2: 0.6307297396406307
Accuracy: 0.46063202999464 38
False positive rate: 0.46063202999464 38
True positive rate: 0.46063202999464 38
F-measure: 0.2905343203486569
Precision: 0.212181867056986 44
Recall: 0.46063202999464 38
```

Figura 5.19: Métricas antes del ajuste de parámetros.

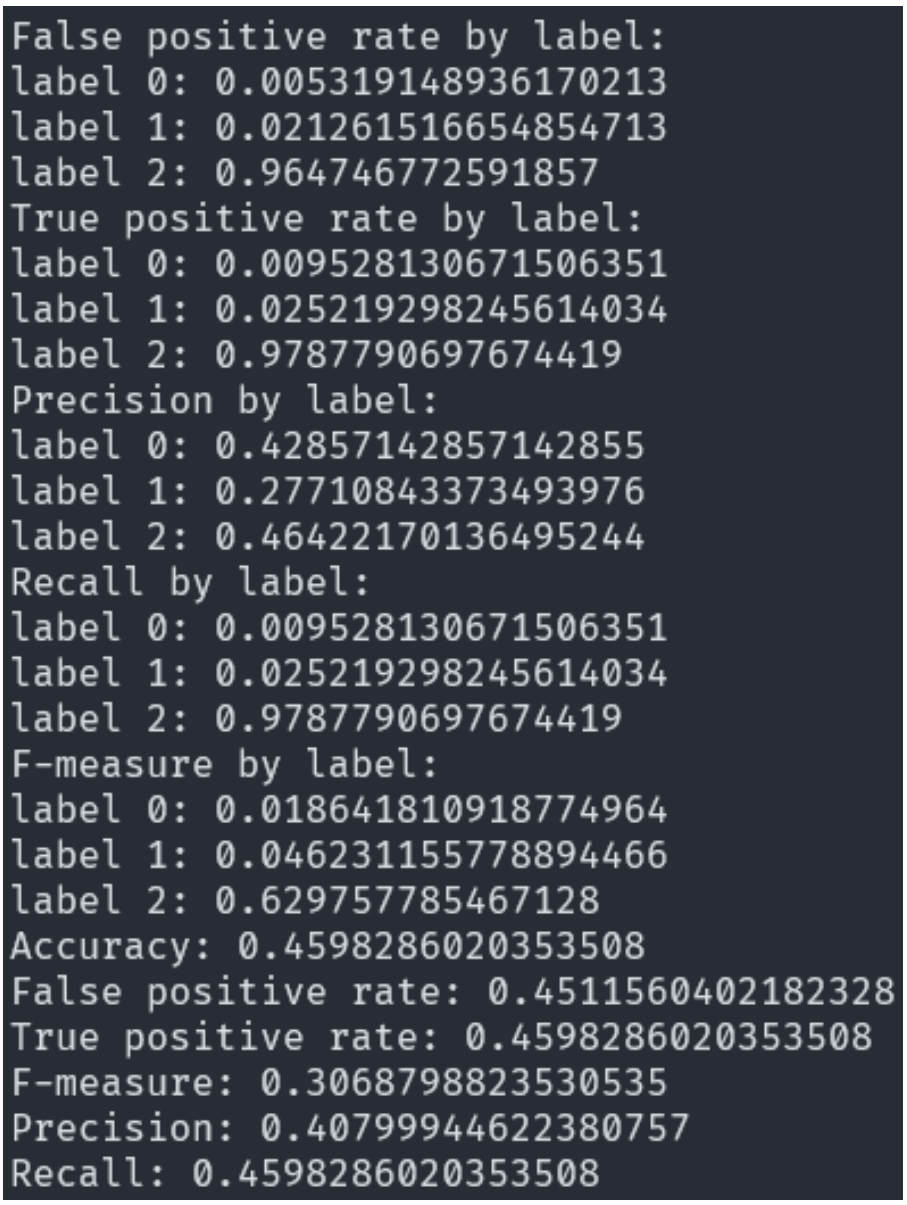


Figura 5.20: Métricas después del ajuste de parámetros.

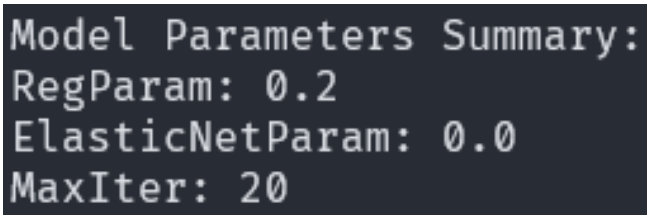
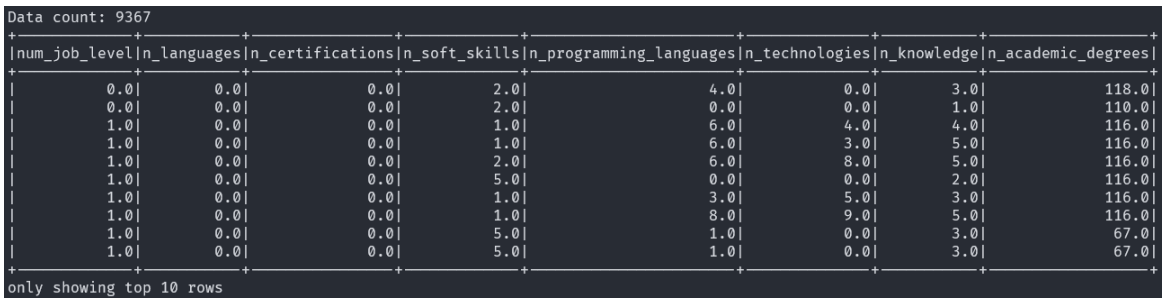


Figura 5.21: Resumen de los mejores parámetros encontrados.

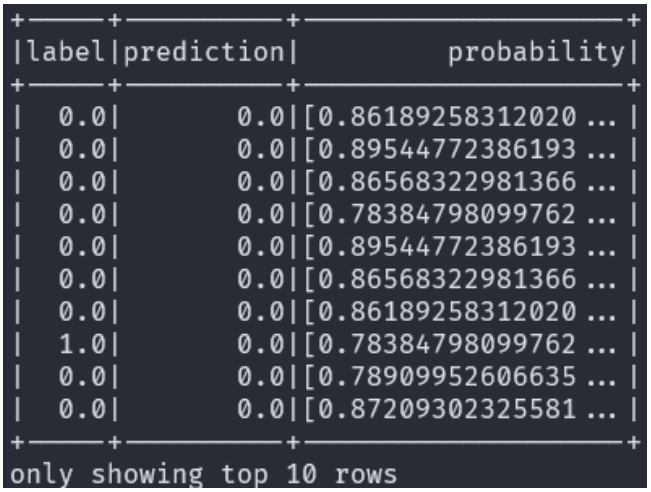
5.2.2. Consulta ML 2

Aplicamos el modelo de DecisionTreeClassifier con dos clases tomando como etiqueta *num_job_level*.



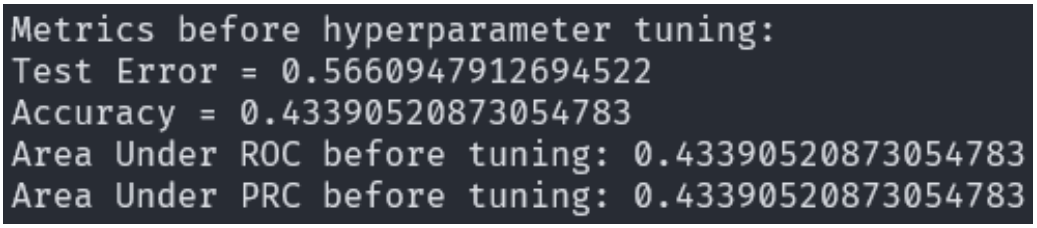
```
Data count: 9367
+-------------+-----------+----------------+-------------+----------------------+--------------+-----------+-------------------+
|num_job_level|n_languages|n_certifications|n_soft_skills|n_programming_languages|n_technologies|n_knowledge|n_academic_degrees|
+-------------+-----------+----------------+-------------+----------------------+--------------+-----------+-------------------+
|          0.0|        0.0|             0.0|          2.0|                   4.0|           0.0|        3.0|              118.0|
|          0.0|        0.0|             0.0|          2.0|                   0.0|           0.0|        1.0|              110.0|
|          1.0|        0.0|             0.0|          1.0|                   6.0|           4.0|        4.0|              116.0|
|          1.0|        0.0|             0.0|          1.0|                   6.0|           3.0|        5.0|              116.0|
|          1.0|        0.0|             0.0|          2.0|                   6.0|           8.0|        5.0|              116.0|
|          1.0|        0.0|             0.0|          5.0|                   0.0|           0.0|        2.0|              116.0|
|          1.0|        0.0|             0.0|          1.0|                   3.0|           5.0|        3.0|              116.0|
|          1.0|        0.0|             0.0|          1.0|                   8.0|           9.0|        5.0|              116.0|
|          1.0|        0.0|             0.0|          5.0|                   1.0|           0.0|        3.0|               67.0|
|          1.0|        0.0|             0.0|          5.0|                   1.0|           0.0|        3.0|               67.0|
+-------------+-----------+----------------+-------------+----------------------+--------------+-----------+-------------------+
only showing top 10 rows
```

Figura 5.22: Los primeros 10 registros de los datos de entrada.



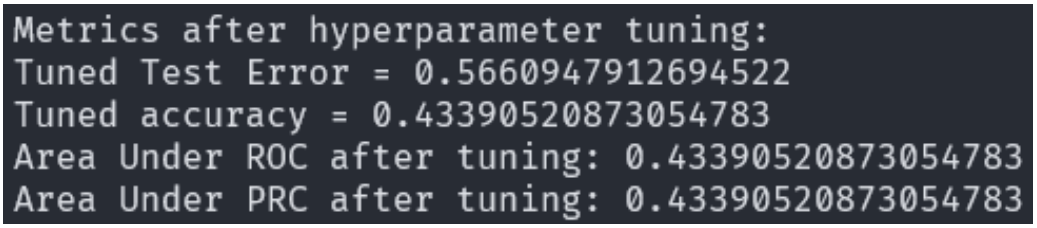
```
+-----+----------+--------------------+
|label|prediction|         probability|
+-----+----------+--------------------+
|  0.0|       0.0|[0.86189258312020 ... |
|  0.0|       0.0|[0.89544772386193 ... |
|  0.0|       0.0|[0.86568322981366 ... |
|  0.0|       0.0|[0.78384798099762 ... |
|  0.0|       0.0|[0.89544772386193 ... |
|  0.0|       0.0|[0.86568322981366 ... |
|  0.0|       0.0|[0.86189258312020 ... |
|  1.0|       0.0|[0.78384798099762 ... |
|  0.0|       0.0|[0.78909952606635 ... |
|  0.0|       0.0|[0.87209302325581 ... |
+-----+----------+--------------------+
only showing top 10 rows
```

Figura 5.23: Predicciones de los 10 primeros registros.



```
Metrics before hyperparameter tuning:
Test Error = 0.5660947912694522
Accuracy = 0.43390520873054783
Area Under ROC before tuning: 0.43390520873054783
Area Under PRC before tuning: 0.43390520873054783
```

Figura 5.24: Métricas antes del ajuste de parámetros.



```
Metrics after hyperparameter tuning:
Tuned Test Error = 0.5660947912694522
Tuned accuracy = 0.43390520873054783
Area Under ROC after tuning: 0.43390520873054783
Area Under PRC after tuning: 0.43390520873054783
```

Figura 5.25: Métricas después del ajuste de parámetros.

```
Model Parameters Summary:
Best maxDepth: 5
Best impurity: gini
```

Figura 5.26: Resumen de los mejores parámetros encontrados.

5.2.3. Consulta ML 3

Aplicamos el modelo de RandomForestClassifier con múltiples clases tomando como etiqueta *num_search_country*.

Data count: 9367

num_search_country	n_languages	n_certifications	n_soft_skills	n_programming_languages	n_technologies	n_knowledge	n_academic_degrees
3.0	0.0	0.0	2.0	4.0	0.0	3.0	118.0
3.0	0.0	0.0	2.0	0.0	0.0	1.0	110.0
3.0	0.0	0.0	1.0	6.0	4.0	4.0	116.0
3.0	0.0	0.0	1.0	6.0	3.0	5.0	116.0
3.0	0.0	0.0	2.0	6.0	8.0	5.0	116.0
3.0	0.0	0.0	5.0	0.0	0.0	2.0	116.0
3.0	0.0	0.0	1.0	3.0	5.0	3.0	116.0
3.0	0.0	0.0	1.0	8.0	9.0	5.0	116.0
3.0	0.0	0.0	5.0	1.0	0.0	3.0	67.0
3.0	0.0	0.0	5.0	1.0	0.0	3.0	67.0

only showing top 10 rows

Figura 5.27: Los primeros 10 registros de los datos de entrada.

label	prediction	probability
1.0	0.0	[0.78499201617239 ...
1.0	0.0	[0.78499201617239 ...
1.0	0.0	[0.78499201617239 ...
1.0	0.0	[0.72768390799408 ...
3.0	0.0	[0.87127881663451 ...
3.0	0.0	[0.82335308543183 ...
0.0	0.0	[0.78911632530651 ...
0.0	0.0	[0.75876516039763 ...
0.0	0.0	[0.87452852715009 ...
0.0	0.0	[0.85793482384415 ...

only showing top 10 rows

Figura 5.28: Predicciones de los 10 primeros registros.

```
Metrics before hyperparameter tuning:
Test Error = 0.18483412322274884
Accuracy = 0.8151658767772512
F1 Score = 0.8151658767772512
Weighted Precision = 0.8151658767772512
Weighted Recall = 0.8151658767772512
```

Figura 5.29: Métricas antes del ajuste de parámetros.

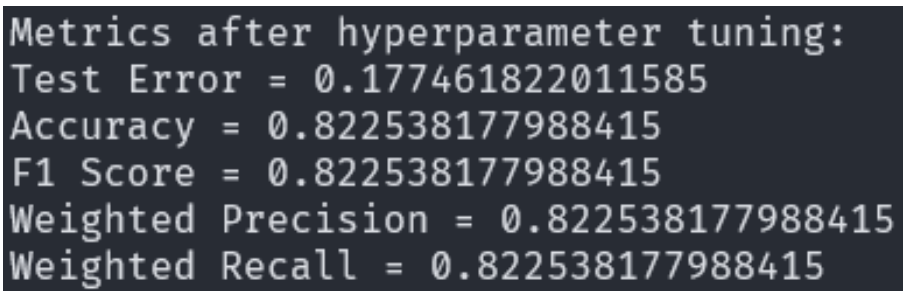


Figura 5.30: Métricas después del ajuste de parámetros.

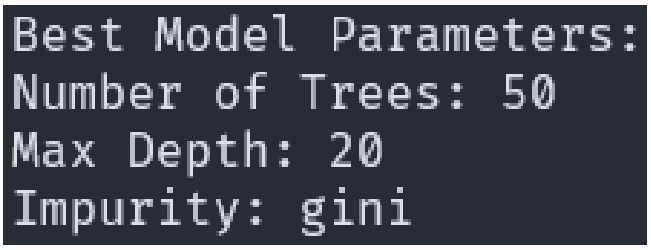


Figura 5.31: Resumen de los mejores parámetros encontrados.

Capítulo 6

Conclusiones

- Se logró explorar y visualizar las tendencias de requerimientos en las propuestas de trabajo para la carrera de Ciencia de la computación en la plataforma de LinkedIn mediante consultas con Scala y Spark.
- Se obtuvo que el tiempo de ejecución de las consultas realizadas con Spark SQL y UDF eran en su mayoría menor que el de las consultas con Scala y Spark.
- Se reconoció que el modelo de Random Forest de la librería MLlib obtuvo un valor de accuracy bueno al momento de clasificar el valor de *num_search_country* con respecto a los features numéricos generados en la limpieza y manipulación de los datos. Mientras, que en los modelos de Regresión Logística y Decision Tree se obtuvo un valor de accuracy menor al momento de clasificar los valores de *num_job_type* y *num_job_level*.

Capítulo 7

Anexo Código

El código relacionado a la limpieza y exploración de datos lo pueden revisar en Google Colaboratory https://drive.google.com/file/d/1R-pBUjueDs4-MpBmtTDUWLnKTMcyyfje/view?usp=sharing.

Bibliografía

[1] Apache Spark. Apache spark™ - mllib: Machine learning guide, 2024. Accedido el 12 de Junio del 2024.

[2] Apache Spark. Apache spark™ - rdd programming guide, 2024. Accedido el 12 de Junio del 2024.

[3] Apache Spark. Apache spark™ - spark sql, 2024. Accedido el 12 de Junio del 2024.

[4] Apache Spark. Apache spark™ - unified analytics engine for big data, 2024. Accedido el 12 de Junio del 2024.

[5] ASANICZKA. Linkedin software engineering jobs dataset, 2024. Accedido el 6 de Junio del 2024.

[6] Databricks. User-defined functions (udfs), 2024. Accedido el 12 de Junio del 2024.

[7] Google Cloud. ¿qué es apache spark?, 2024. Accedido el 12 de Junio del 2024.