Report on

APPENDIX 1
Preparing the data set

Prepared by

Laima Lukoseviciute

January 27, 2026

Table of contents

Introduction

This report presents a data preparation for the EDA step for Coresignal jobs data. The primary goal of the further analyses will be to identify trends, relationships, and interesting angles within the programming language landscape of 2025.

The initial raw data was gained from the Coresignal Multi-Source Jobs Dataset, which aggregates listings from major global job boards. Since the Job listings for 2025 have more than 60 mln job postings I have decided to extract the job titles in the US, only where the job title contains selected keywords (plural of the words was also accepted):

`developer` OR `analyst` OR `programmer` OR `programming` OR `scientists` OR `data` OR `researcher` OR `engineer` OR `engineering`.

The data was extracted using this SQL query that can be seen below.

```
SELECT
  title, description, company_name, company_industry, state, created_at,
FROM
  `oxy-analytics.raw_external_cosi_core.multisource_job`
WHERE
  created_at > "2024-12-31" AND
  country = "United States" AND
  REGEXP_CONTAINS(title, r"(?i)\b(developers?|analysts?|programmers?|programming|
  scientists?|data|researchers?|engineers?|engineering)\b")
```

The analyzed dataset has 6 primary features and over 2 062 382 observations, ranging from January 1, 2025 to December 31, 2025. For a detailed breakdown of the features, please refer to the Table 1.

Table 1: The description of variables for data.

Variable Name	Type	Description
title	STRING	The professional title of the job listing.
description	STRING	The full text of the job post, used for keyword extraction.
company_name	STRING	The name of the hiring organization.
company_industry	STRING	The sector the company operates in (e.g., Tech, Finance).
state	STRING	The US state of the job location.
created_at	TIMESTAMP	The date when the job listing was added to the database.

A preview of the analysed dataset is presented below in Table 2.

Table 2: Raw data pre-view first 5 rows. Note: if the table is not visible

	title	description	company_name
0	Scientist (non-PhD), Tumor Immunogenicity	Why Patients Need You Pfizer's purpose is to d...	NaN
1	Sr. Scientist, CMC Analytical	About Loyal Loyal is a clinical-stage veterina...	Loyal
2	Transit Coordinator - Financial Analyst	Posted: Oct 1, 2025 Transit Coordinator - Fina...	National Grants Mana
3	Senior Manager, Capacity Engineering, North Am...	Description As the Capacity Engineering lead f...	Amazon
4	Business Intelligence Engineer	Job Title: Business Intelligence Engineer Loca...	IntelliSavvy

Data Preparation

The extracted data totals approximately 10 GB. To ensure the system processes this volume of information efficiently, I have partitioned the data into 11 separate files. After the programming language data is extracted, the "description" column will be removed. This adjustment allows the information from all files to be combined into a single dataset for the analysis and saved to the file `data/processed/jobs_proc_2025_no_desc.csv`.

The function `extract_programming_languages` extracts these programming languages: Python, SQL, Java, JavaScript, TypeScript, C++, C#, Objective-C, C, R, Go, Swift, PHP, Ruby, Kotlin, Rust, Matlab, Scala, Perl, Dart, Bash, Assembly. The detection of all the langueges except “Go” are handaled with regular expressions. To process the language “Go” I will be using LLM to interpret the meaning of the word “go” in the description.

Below you can see the list of all the languages found and the corresponding quantities of job postings where that language was menationed.

```
(2062297, 27)
```

Table 3: Raw data pre-view first 5 rows. Note: if the table is not visible in pdf, please see html version of the report.

	title	Python	SQL	Java	JavaScript	TypeScript	C++
0	Scientist (non-PhD), Tumor Immunogenicity	0	0	0	0	0	0
1	Sr. Scientist, CMC Analytical	0	0	0	0	0	0
2	Transit Coordinator - Financial Analyst	0	0	0	0	0	0
3	Senior Manager, Capacity Engineering, North Am...	1	1	0	0	0	0
4	Business Intelligence Engineer	1	1	0	0	0	0

NOTE:

The `extract_programming_languages` function utilizes the Ollama Large Language Model (LLM) to identify the meaning of the word “Go”. To ensure the most consistency of these results and minimize variability in model output, the temperature parameter is set to 0.

Below I will evaluate the error rate for the other programming languages. Which of them mean the language and which of them do not. The list of languages can be seen below. “Go_verified” will not be checked.

```
['Python',
 'SQL',
 'Java',
 'JavaScript',
 'TypeScript',
 'C++',
 'C#',
 'Objective-C',
 'C',
 'R',
 'Swift',
 'PHP',
 'Ruby',
 'Kotlin',
 'Rust',
 'Matlab',
 'Scala',
 'Perl',
 'Dart',
 'Bash',
 'Assembly',
 'Go_verified']
```

Below at Table 4 you can see the preciton evaluation for each language. These precision evaluations are not very precice. I just evaluated it by going through the raw files, python was always a programming language in the given sample, not somwthing else. Thus I will implement them only for languages that need them. Like “R”, “C”, and “Assembly”. Because there are a lot of engineers that are required to work with “assembly lines”.

Table 4: Precision evaluation of the programming languages.

	language	precision
0	Python	0.74
1	SQL	0.59
2	Java	0.70
3	JavaScript	0.40
4	TypeScript	0.63
5	C++	0.70
6	C#	0.74

Table 4: Precision evaluation of the programming languages.

	language	precision
7	Objective-C	0.56
8	C	0.32
9	R	0.64
10	Swift	0.58
11	PHP	0.40
12	Ruby	0.57
13	Kotlin	0.72
14	Rust	0.73
15	Matlab	0.70
16	Scala	0.61
17	Perl	0.41
18	Dart	0.50
19	Bash	0.23
20	Assembly	0.17

```
/var/folders/1_/mwhsmc3x39g2h4ny87__zzsr0000gn/T/ipykernel_71775/2410304175.py:3: FutureWarning: Settin
  top_programming_languages["C"]= top_programming_languages["C"]*df_precision[df_precision["language"]=
/var/folders/1_/mwhsmc3x39g2h4ny87__zzsr0000gn/T/ipykernel_71775/2410304175.py:4: FutureWarning: Callin
  top_programming_languages["R"]= top_programming_languages["R"]*df_precision[df_precision["language"]=
/var/folders/1_/mwhsmc3x39g2h4ny87__zzsr0000gn/T/ipykernel_71775/2410304175.py:5: FutureWarning: Callin
  top_programming_languages["Assembly"]= top_programming_languages["Assembly"]*df_precision[df_precisio
```

```
Python         401168.00
SQL            387326.00
Java           176246.00
JavaScript     158261.00
C++            100799.00
Bash            96966.00
C#              79789.00
TypeScript      60446.00
R               53583.36
Go_verified     39960.00
C               34499.20
Matlab          33412.00
Scala           20810.00
Perl            19576.00
Ruby            19265.00
Swift           17892.00
Kotlin          16056.00
Rust            15245.00
PHP             13544.00
Assembly        10778.68
Objective-C      4487.00
Dart             1707.00
dtype: float64
```

Observations and Features

To make sure the function did what it was suppose to do let's do the exploration of the dataset's structure. I will examine the characteristics of each column to ensure data integrity and understand the available information.

Below you can see the list of all the columns in the processed dataframe. All the programming languages were included.

```
Index(['title', 'company_name', 'company_industry', 'state', 'created_at',
       'Python', 'SQL', 'Java', 'JavaScript', 'TypeScript', 'C++', 'C#',
       'Objective-C', 'C', 'R', 'Swift', 'PHP', 'Ruby', 'Kotlin', 'Rust',
       'Matlab', 'Scala', 'Perl', 'Dart', 'Bash', 'Assembly', 'Go_verified'],
      dtype='object')
```

Below Table 5 you can see the description of categorical data. We can see that there are 598373 unique title in the data and 107161 unique companies.Top industry is Software Development, and top state is California. At Table 6 you can see that the data covers 2025.

Table 5: Description of the categorical data

	title	company_name	company_industry	state
count	2062297	2039563	1782048	1459598
unique	598373	107161	427	138
top	Financial Analyst	Jobs via Dice	Software Development	California
freq	6863	68134	240228	215729

Table 6: Description of the date data

	created_at
count	2062297
mean	2025-06-22 23:52:19.825892096+00:00
min	2025-01-01 00:14:21+00:00
25%	2025-03-25 02:28:17.284700928+00:00
50%	2025-06-25 07:36:12.390136064+00:00
75%	2025-09-12 19:41:54.569914880+00:00
max	2025-12-19 09:18:48.440245+00:00

Duplicate and Missing Values

In this section I will analyse if the data set has any duplicated observations or missing values. From the outputs below we can see that data have 337390 missing values in column `state`, 175057 in column `company_industry`, and 175057 in `company_name`. The full breakdown can be seen below in Table 7.

Table 7: Missing values in the data set by column.

	column_name	no_values_missing	percentage_values_missing
3	state	602699	29.22
2	company_industry	280249	13.59
1	company_name	22734	1.10
0	title	0	0.00
15	Swift	0	0.00
25	Assembly	0	0.00
24	Bash	0	0.00
23	Dart	0	0.00
22	Perl	0	0.00
21	Scala	0	0.00
20	Matlab	0	0.00
19	Rust	0	0.00
18	Kotlin	0	0.00
17	Ruby	0	0.00
16	PHP	0	0.00
13	C	0	0.00
14	R	0	0.00
12	Objective-C	0	0.00
11	C#	0	0.00
10	C++	0	0.00
9	TypeScript	0	0.00
8	JavaScript	0	0.00
7	Java	0	0.00
6	SQL	0	0.00
5	Python	0	0.00
4	created_at	0	0.00
26	Go_verified	0	0.00

Table 7: Missing values in the data set by column.

column_name	no_values_missing	percentage_values_missing

720741 were duplicated values. I will keep missing values, and will do the analyses with them in mind, and I will remove the duplicated values, since it is the same job posting.

```
Number of duplicated values:  720720
```

Outliers

In this section let's look for some obvious outliers or other descrepencies in the data. The job `title` contains some obvious outliers, like resercher that is not related to data, but rather the academic enviroment, and egnineering manager mignt not need any knowlage of the programming languages, thus I will remove all the rows that have no mentions of any programming languages. I will also filter out the job

```
Number of rows and columns after filtering jobs (rows) that have no mentions of programming languages:
(581120, 27)
```

Also some single letter languages like C and R might be caught due to the fact that there are typos like: "C ompletely" or "R esponcible" in the job descriptions.

```
array(['Tennessee', 'Washington', 'California', 'Utah', nan, 'Wisconsin',
       'Texas', 'Idaho', 'Indiana', 'Michigan', 'Massachusetts',
       'New York', 'Illinois', 'Arizona', 'North Carolina', 'Virginia',
       'Georgia', 'New Jersey', 'Ohio', 'Iowa', 'Minnesota', 'Colorado',
       'Maryland', 'Oklahoma', 'Arkansas', 'Pennsylvania', 'Florida',
       'Nevada', 'Missouri', 'New Mexico', 'Hawaii',
       'District of Columbia', 'Connecticut', 'Wyoming', 'Oregon',
       'Puerto Rico', 'Kansas', 'United States', 'Rhode Island',
       'New Hampshire', 'Alabama', 'Delaware', 'Mississippi', 'Vermont',
       'South Carolina', 'MN', 'Montana', 'Kentucky', 'North Dakota',
       'Nebraska', 'South Dakota', 'Maine', 'SC', 'Louisiana',
       'Metropolitan Area', 'West Virginia', 'Alaska', 'MA', 'Carolina',
       'County', 'DC', 'GA', '   ', 'TX', 'St Croix', 'San Juan', 'WI',
       'US Virgin Islands', 'indiana', 'Dededo Municipality',
       'Eastern District', 'Barrigada Municipality', 'WA',
       'Sarasota Area', 'MD', 'Gurabo Municipio', 'Grants Pass Area',
       'Guam', 'Virgin Islands', 'Provincia de Las Palmas',
       'Cayey Municipio', 'ND', 'Floride', '    ', 'Nowy Jork', 'OH',
       'DE', 'Guaynabo', '         ', 'states',
       'Estados Unidos', 'American Samoa', 'e Região', '      '],
      dtype=object)
```

These are states and US territories, which have left after the cleaning.

```
55
```

```
array(['Alabama', 'Alaska', 'Arizona', 'Arkansas', 'California',
       'Colorado', 'Connecticut', 'Delaware', 'District of Columbia',
       'Florida', 'Georgia', 'Guam', 'Hawaii', 'Idaho', 'Illinois',
       'Indiana', 'Iowa', 'Kansas', 'Kentucky', 'Louisiana', 'Maine',
       'Maryland', 'Massachusetts', 'Michigan', 'Minnesota',
       'Mississippi', 'Missouri', 'Montana', 'Nebraska', 'Nevada',
       'New Hampshire', 'New Jersey', 'New Mexico', 'New York',
       'North Carolina', 'North Dakota', 'Ohio', 'Oklahoma', 'Oregon',
       'Pennsylvania', 'Puerto Rico', 'Rhode Island', 'South Carolina',
       'South Dakota', 'Tennessee', 'Texas', 'Utah', 'Vermont',
       'Virgin Islands', 'Virginia', 'Washington', 'West Virginia',
       'Wisconsin', 'Wyoming'], dtype=object)
```

```
broad_industry_group
Tech, Data & Telecom                       256058
Professional, Legal & Business Services     82789
Miscellaneous                               67950
```

```
Manufacturing, Industrial & Defense         60538
Finance, Insurance & Real Estate            47937
Healthcare, Pharma & Wellness               18538
Education, Government & Non-profit          13834
Logistics, Travel & Construction            10287
Consumer, Retail & Agriculture               8723
Energy, Utilities & Environment              8306
Media, Entertainment & Arts                  6160
Name: count, dtype: int64
```

Feature Engineering

To provide a more structured view of the recruitment landscape, I am standardizing the job titles within the dataset. Currently, the data contains approximately 600 000 unique job titles, which is a level of detail that can obscure broader market trends.

By grouping these titles into five primary categories, the analysis becomes more accessible for identifying high-level patterns. These categories include:

- Manager: Roles focused on leadership and project oversight.
- Engineer: Positions centered on building and maintaining technical systems.
- Analyst: Roles dedicated to interpreting data and providing insights.
- Scientist: Research-oriented positions, including Data Scientists and Researchers.
- Developer: Traditional software creation and programming roles.

This categorization simplifies the comparison of programming language requirements across different professional functions. The final dataframe can be found in file `data/processed/jobs_filtered_2025_no_desc.csv`.

```
(581120, 33)
```

	title	manager	engineer	analyst	scientist	developer	company_nar
3	Senior Manager, Capacity Engineering, North Am...	1	1	0	0	0	Amazon
4	Business Intelligence Engineer	0	1	0	0	0	IntelliSavvy
5	HAZARDOUS SUBSTANCES ENGINEER	0	1	0	0	0	California Dep
9	Senior ASIC Synthesis Engineer	0	1	0	0	0	NVIDIA
12	Product Security Engineer	0	1	0	0	0	Grammarly

Summary

This stage of the project focused on transforming around 10 GB of raw job posting data from the Coresignal Multi-Source Jobs Dataset into a structured, analysis-ready format. To ensure the data remained manageable on local hardware while maintaining depth, the following steps were completed:

- Targeted Extraction: I have narrowed the scope to over 2 062 382 observations from 2025 specifically within the United States, filtering for key technical roles such as developers, analysts, and engineers.
- Data Integrity: The dataset was refined by removing 720 741 duplicate entries and filtering out job titles that lacked any programming language mentions (e.g., academic researchers or pure management roles).
- Technical Standardization: Using the extract_programming_languages function, we identified mentions of 22 programming languages. Ambiguous terms like "Go" were processed using a local Large Language Model (Ollama) with a temperature of 0 to ensure high reproducibility and minimize false positives.
- Categorization: To understand market trends, I have engineered five high-level job categories: Manager, Engineer, Analyst, Scientist, and Developer.

The resulting processed dataset contains 602 359 high-quality job postings, significantly reduced from the initial data, allowing for efficient and high-impact EDA.

Suggestions for Further Improvements

The error rate calculations for the LLM used for the interpretation programming languages could be done. I think this will be marginal improvement and will not change the results that much, but just for the sake of being precice and redusing the error this should be done.