Technical report of data pipeline for analysis of US census microdata

ANTONI SKUBISZ - 36145142

1 INTRODUCTION

A census is an official count of the population of a particular area or country at a specific point in time. Its aim is to collect data regarding the demographic, social, and economic characteristics of the population of interest. It is an essential tool for governments and researchers to understand the needs of the population and to plan the allocation of resources accordingly. Moreover, the data can be used by data scientists for predictive modeling, data validation or enrichment, market research, and some interesting visualisations. In 1962, the U.S. Census Bureau, in an attempt to meet the demand from social science research, sampled 1 in 1000 samples from the decennial census, removed sensitive information, and released the dataset to researchers as a Census Microdata [1]. Nowadays, every year, the U.S. Census Bureau releases the Census Microdata as a subset of individual or household records from the original Census database. Data is accessible through the API on the official website, and our data pipeline will be built upon it. The motivation for our analysis is an overview of trends that have emerged in the US over the last ten years. For example, changes in the average age of US citizens, average rent, or average time to travel to work. Additionally, we aim to answer a question: "Is it possible to accurately predict a person's salary based on their characteristics?". A machine learning model can be developed to answer this question, and its usefulness can span across a range of applications, from labor market analysis to personal finance management. For instance, an accurate model could help an individual to set realistic goals based on their expected income, or identify areas where they can improve to increase their earning potential. This report will describe the details of our data pipeline and the design rationale behind it.

2 BACKGROUND

The availability and the quantity of the data from Census Microdata exploded in recent years. According to the article from 2013 [1], the number of individual records accessible to researchers has surged from 100 million in 2000 to a staggering 750 million in 2013. It was then projected that this number could surpass 2 billion by 2018. Such accessibility facilitated the development of many pieces of research.

Recently, the new measure of doubled-up homelessness was presented in [2]. This type of homelessness occurs when people live with other individuals or families due to financial reasons. Official statistics usually do not consider this type of homelessness and only include individuals who are living on the streets. The new measure's calculation is based on US Census Microdata and it was estimated that 3.7 million households were doubled up. Owing to this research, homelessness can be better understood and new programs and policies can be implemented to address this issue.

Another example could be an article from 2012 [3] where the author used microdata to examine the relationship between physical activity and happiness. It was found that people who more frequently participated in sports activities were happier than those who did not exercise as much. Such study suggests that promotion of sports is a promising strategy for the improvement of general well-being among people.

The article from 2015 [4] analyses the impact of school starting age policy on crime rates in the US by using microdata. The findings suggest that individuals who started school later due to entry

Author's address: Antoni Skubisz - 36145142.

policies were more likely to engage in criminal activity later in life compared to those who started school earlier. It can be concluded that policymakers should take that into account and consider the long-term effects of school starting age.
These studies are examples of the significant value that analysis of Census Microdata can bring to society. Regarding the model for salary prediction, in a study by [5], the authors developed a salary prediction model for students to improve their motivation for studying and encourage a positive outlook. However, we have not come across any research that has attempted to build a similar model for the general population.

3 DATASET AND CHARACTERISTICS

To answer our research question, we identified US Census Microdata to be a perfect fit for our goal. Firstly, it contains a comprehensive list of available variables (which counts 525) and their data types so it will certainly contain the ones that we might find relevant to our model. Secondly, it can be sourced from simple to use, public API. Thirdly, it is a well-trusted, official government institution so the data should be valid and free of any bias or inaccuracies that might be present in other sources. Fourthly, Census Microdata is collected at regular intervals, which allows us to track changes in population characteristics over time, and this will be important for our visualisations of trends in the US. Finally, the dataset is large and diverse enough to allow us to train a reliable machine learning model. The only downfall of the data is that it is provided in a non-standard format. Essentially, it is a mixture of comma-separated format with double arrays. This means that the data will require some special care before loading into the database. Now, we will explore some typical characteristics that occur in big data:

- **Volume** - The original dataset does exhibit a large number of records, which is enough to break our ingestion pipeline. Hence, to solve this issue we restricted our universe by using **yoep** variable, which represents the year of entry of the record. Consequently, we collected data from 2011 to 2021.
- **Velocity** - Due to the nature of our data, it did not show signs of velocity as it is not updated in real time. Once you make an API call, you get the same output as you would in the last month.
- **Variety** - The entirety of the data gets collected from one source in a semi-structured format, and then it gets transformed into a structured one as we load it into a database. Hence, the data does not showcase variety.
- **Veracity** - As mentioned before, the data comes from an official government website, hence, there are no concerns about the trustworthiness of the data. Moreover, inside the pipeline, there are checks placed to ensure that the data follows the right format.

The one limitation that we experienced was connected to the mentioned volume of the data. Essentially, the number of operations needed to transform the data was exceeding the capacity of RAM on the machine. The solution to that problem will be discussed in the future section. The variables that we queried from API, that we found to be interesting to analyse in terms of change in trend, are: **gender**, **age**, **educational level**, **marital status**, **military status**, **gross rent**, **gross rent as a percentage of household income**, **household language**, **travel time to work**, **travel time to work** and **year of entry of datapoint**. Out of these variables that we will visualise, we chose the first four to be the features in our machine learning model.

4 DESIGN APPROACH

The purpose of our pipeline is visualisation of the data and using it for a machine learning model. It is important to note that we deal with data that is already collected and stored, so it is not generated

in real time, as this will dictate our future design choices. During the data ingestion stage, we source it from 3rd party using REST API. The type of ingestion is batched due to the nature of the data and the fact that it does not need to be processed in real time. The alternative choice would be streaming ingestion, which occurs more frequently in the world for example in social media or environment sensors. The ingestion pipeline does not have any features to deal with potential schema evolution as it is very unlikely that it will happen. When new census data is posted, it gets a new unique URL so the result of our API query should not change. Now, we will move on to our choices for methods for data integration and transformation. We followed ETL (Extract, Transform, Load) structure, which means that after getting the data from our source, we transform it into a desired format, and load it into a target system, which in our case is a data warehouse. The transformation step consists only of format standardisation. Received data is in a quite non-standard format and it is not possible to insert such format into our data warehouse, hence various steps (that will be described in detail in the next section) had to be taken. The visualisation of the received and desired format can be seen in Figure 1.

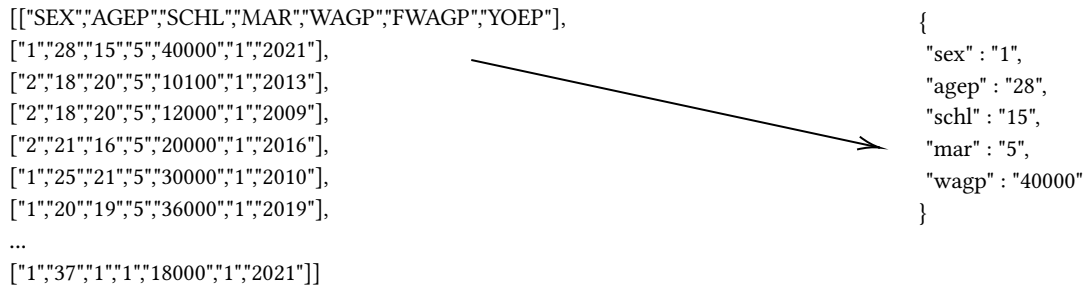

Fig. 1. Example of transformation procedure based on the first row of data

The advantages of ETL include data quality and centralised data storage. ETL ensures the aforementioned quality by the set of transformations (normalisation, cleaning) that are performed before loading. It is most beneficial when data consistency is important. ETL pipeline also allows ingesting data from multiple sources and tracking the lineage of the data easily. The drawbacks are time and resource consumption. In essence, transformations applied might not be well suited for big volumes of data, and there might be a delay between the collection of the data and when it is ready for analysis. The alternative choice is ELT, however, in our case, it was not useful as we prioritised the consistency of data over time. The next step is storage in our data warehouse which we chose to be a relational database. This is the most optimal choice for us due to the structured format of the data, security, and potential scalability in the future. The drawback of this approach is that it can be slower than other types of databases when dealing with complex queries. In the processing step, the data gets queried from the data warehouse in batch and fed into a machine learning model (details in the next section). The model is then used for the prediction table, which is then sent to the data warehouse. Lastly, in the serving stage, we connect RDBMS with an interactive dashboard. It allows non-technical people to explore the findings and analytics performed. The whole process is summarised in Figure 2.

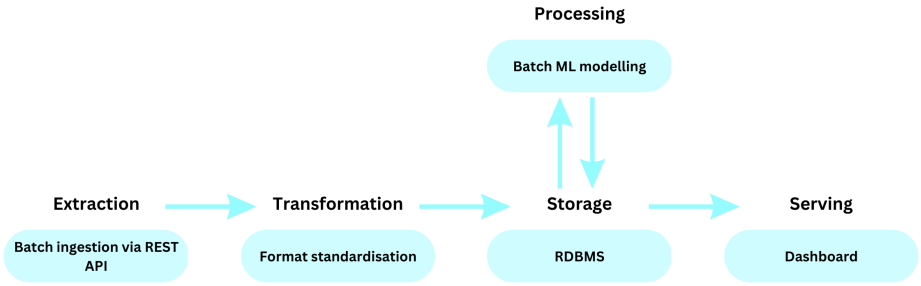

Fig. 2. Diagram of the stages of the pipeline

5 IMPLEMENTATION

Now, we will show the software and methods that we used to implement our pipeline. The overview of all of them is presented in Figure 3. Everything has been "dockerised" and the instructions on how to run each step has been put in */code/instructions.txt* file in the submission.

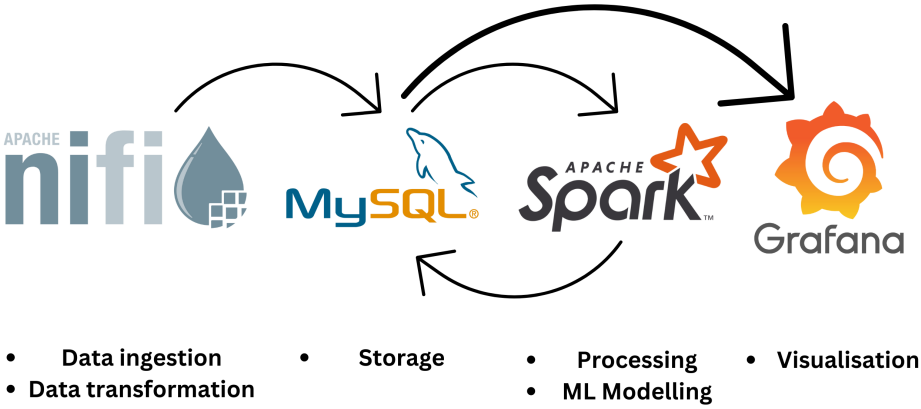

Fig. 3. Diagram of tools used at each step of the pipeline

5.1 Apache Nifi

The pipeline starts with Apache Nifi. We decided that Nifi will be the most optimal choice for the ingestion and transformation step due to its great features in terms of data provenance and tracking data lineage. We created two separate process groups, one for loading data from the past decade for the visualisation purpose, and the other to load only data that will be relevant to the ML model. In the latter, we only considered records that were entered between 2000 and 2021 and their salary is between 10,000$ and 1,000,000$. It was done to ensure that data is recent enough and to remove outliers. The flow starts with the InvokeHTTP processor which extracts data from public API and gets saved into a FlowFile content. We are aware that the format that is easily inserted into a MySQL table is JSON, hence a series of transformations needs to be performed. Firstly, SplitText processor is used to split each line of the content into separate FlowFiles. The first one, which contains column names (look Figure 1), gets discarded. The content of each FlowFile gets saved to

its attribute, and from there, expression language can be used to perform string manipulation. All the unnecessary square brackets and quotation marks get removed so that the attribute content is simply comma separated. Now, the values can be easily separated into new attributes (whose names correspond to the column name) by using getDelimitedField in the UpdateAttribute processor. The last transformation step is using AttributeToJSON processor which outputs the data in the desired format. Our approach to transformation was reasoned by its simplicity and the ability to track progress at each step. During the execution of the processors, we noticed that splitting content into lines and saving content into attributes were especially time consuming, which might be a potential downside of our approach. An alternative choice would be to run a Python script as a part of our flow using ExecuteStreamCommand processor. However, it required an additional setup of Python, so we decided that our approach will be more straightforward. After transformation, the data in JSON format is ready to be inserted into our MySQL database using PudDatabaseRecord processor.

5.2 MySQL

We chose MySQL as our location of data storage, as it is one of the most popular open-source RDBMS, and we gained some experience with operationalising it during workshops. Additionally to MySQL, we used Adminer as a database management tool that provides GUI. Its main advantage is a more user-friendly way to browse and edit data, as compared to the command-line interface. After examination of Adminer it can be noted that inside of *us_census* database, we made 3 tables: *data*, *earning_data*, and *predictions*. The first one contains the data that is going to be visualised, the second one has the data that will be fed into a machine learning model, and the last one has results of predictions of ML model and we will talk about it in the next subsection.

5.3 Apache Spark

After data is stored in MySQL, it can be easily queried inside Apache Spark by using JDBC (Java Database Connectivity). The reasons why Spark was chosen for the processing step include scalability, easiness of use, and the rich set of ML libraries. Apache Spark is known for being able to process big amounts of data and that it can be scaled up quite easily, if needed. Programming resembles Python very closely, hence it was a natural choice for the processing step. The data that we extracted for modeling contains 32,476 rows of 4 features and 1 target variable. It is a typical regression problem, hence we split the data into training and test sets by 70/30 ratio and try 4 different models. The first one is Linear Regression. It assumes a linear relationship between the input variables and the salary and is a good starting point for modeling. Next, we have Generalised Linear Regression which is a more flexible variation of the previous model, as it can handle non-linear relationships. We chose Gamma family in this model as it assumes positive values that are skewed to the right, which sounds reasonable for salary. Another algorithm that we will try is Decision Tree. Some pros of this model include the ability to capture non-linear relationships and robustness to outliers. Lastly, we have Random Forest which is an extension of the aforementioned model, based on the idea of combining multiple decision trees. Such variations tend to be less likely to overfit and give better generalisation. We train each model on the training set and evaluate results on the test set using RMSE and MAPE. The results are presented in Table 1.

	RMSE	MAPE
Linear Regression	56312.1	333.81%
Generalised Linear Regression	56574.7	335.92%
Decision Tree	51849.3	290.39%
Random Forest	**51776.3**	**290.01%**

Table 1. Error metrics on a test set for each attempted model

Random Forest did the best, hence we use this as the final model. Our approach to operationalise the visualisation of the model is as follows. We created a new table that contains the Cartesian product of values:

- **Sex** - 1 to 2
- **Age** - 0 to 99
- **Education** - 0 to 24
- **Marital Status** - 1 to 5

Consequently we have a table with 25,000 rows ($2 \times 100 \times 25 \times 5$) with the first row being 1,0,0,1, the next one 1,0,0,2 and so on. ML model can be easily applied to predict such table, and the result gets inserted into our database. We found this approach appealing as it can be easily integrated with our dashboard's feature to give it an interactive feeling.

5.4 Grafana

Grafana was our choice for visualisation software. It is an open-source platform that supports flexible data sources, which made it easy for us to connect it with the MySQL database. It allows technical people to create customized dashboards to display data in a way that is meaningful to non-technical people. Grafana is most popular when dealing with time-series data, however, we found it useful in our case as well. Its 'variables' feature allowed us to make the visualisation of the ML model interactive. In essence, the user can specify the values of the features and the dashboard will display prediction for those specified values. The drawback of Grafana is that it requires the data to be queried in a specific format, which might be difficult to achieve in some cases.

6 EVALUATION

6.1 Salary predicting model

As it could be seen in Table 1, the error metrics are not great. There might be various reasons for that. Firstly, we might not have enough data. While around 32,500 data points might sound like a lot, it is also important for the data to be diverse enough and the general rule is that the more data available for training, the better the prediction abilities of the model. Secondly, the features that we chose might not be enough to fully capture the behaviour of the target variable. Moreover, some features seem to be overcomplicated, for example, **education** has values ranging from 0 to 24, where values from 4 to 15 are representing the education level of Grade 1 to Grade 12. We can expect the model to perform better if we restrict samples to only include educational levels such as high school, bachelor's, master's, and doctorate. We also found the model to be giving unreasonable estimates for unrepresented in the training set examples. For instance, it predicts a married 5-year-old boy with a master's degree to be earning a salary of $64,000$. Some restrictions for the input variables could be put in place to deal with it. However, when it comes to examples that are possible, the estimate sound quite reasonable. For example, a 25-year-old married male is expected to make: $39,900$ if he finished high school, $77,400$ if he has a bachelor's degree, $81,500$

if he has a master's degree, and 74, 600$ if he has a doctorate degree. In Figure 4 we present how the prediction panel looks like, with the example input mentioned before.

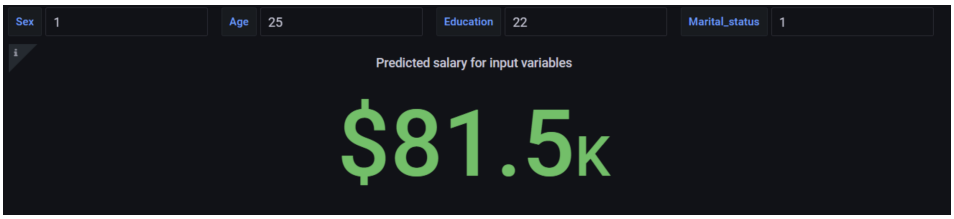

Fig. 4. Example of input into our model and the resulting prediction

6.2 General visualisation

Firstly, we will show the sample of created visualisations in Figure 5.

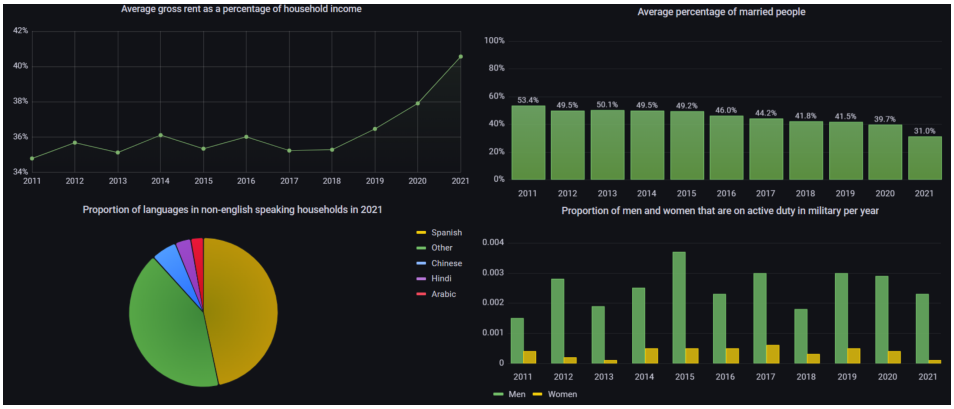

Fig. 5. Examples of visualisations created in the dashboard

As we can see, the type of plots created vary from line chart to pie chart. Starting with the upper left corner, we see that average rent as a percentage of household income increased abruptly in recent years. This might be due to the recent increase in prices for electricity or inflation in general. Authorities after looking at such data might prepare some strategies to tackle this problem. In the upper right corner, we see a bar chart that informs that the percentage of married people is steadily decreasing in recent years. Such observation might be an interesting fuel for a social study of what is the reason behind it. The bottom left corner shows the pie chart presenting the proportion of the language spoken in non-English households in 2021. The full dashboard also contains the second pie chart but with 2011 data. A comparison of these two can give an idea of the origin of the influx of minorities in the last decade in the US. Lastly, in the bottom right corner, we have a proportion of the population in the military, grouped by gender. There is neither an increasing nor decreasing trend, however, there tend to be a higher percentage of males in military. In general, our dashboard presents a few more interesting visualisations, from which more conclusions about society can be made.

7 CONCLUDING REMARKS

To conclude, we have designed and implemented the pipeline with the purpose of visualisation and prediction. Due to the nature of the data we focused on building a pipeline designed for batched data, rather than streaming. Because of that, it is not suitable for events such as schema evolution. The main limitation of our work was the computing resources, as we had to restrict the dataset that we source to make it able to run. With more computing, we would be able to get more observations which would likely make our model more accurate. The possible improvements that could be considered next time include data enrichment. In essence, we might source some additional data from various sources to provide better insights, or to validate our current ones.

REFERENCES

[1] S. Ruggles, "Big microdata for population research," *Demography*, vol. 51, no. 1, pp. 287–297, 2014.
[2] M. K. Richard, J. Dworkin, K. G. Rule, S. Farooqui, Z. Glendening, and S. Carlson, "Quantifying doubled-up homelessness: Presenting a new measure using us census microdata," *Housing Policy Debate*, pp. 1–22, 2022.
[3] H. Huang and B. R. Humphreys, "Sports participation and happiness: Evidence from us microdata," *Journal of economic Psychology*, vol. 33, no. 4, pp. 776–793, 2012.
[4] J. M. McAdams, "The effect of school starting age policy on crime: Evidence from us microdata," *Economics of Education Review*, vol. 54, pp. 227–241, 2016.
[5] P. Khongchai and P. Songmuang, "Implement of salary prediction system to improve student motivation using data mining technique," in *2016 11th International Conference on Knowledge, Information and Creativity Support Systems (KICSS)*. IEEE, 2016, pp. 1–6.