AppJusto - Audit of manage and allocate algorithms of work

Heitor Gouvêa
Independent Security Researcher
research@heitorgouvea.me
www.heitorgouvea.me

[1] - This document is classified as public.

1. Summary

This paper presents an audit performed on the management and work allocation algorithm of the application called "AppJusto". The purpose of the audit is to verify the presence of bias in the algorithms that may result in unequal outcomes for workers from historically or currently disadvantaged groups. Key points of attention are highlighted, including the context of biased algorithms, the importance of mitigating bias in delivery apps, the methodology used in the audit, the analysis findings, and future recommendations to improve app fairness.

The current audit found no bias in AppJusto's management and work allocation algorithm that could negatively impact workers from disadvantaged groups, but highlights the importance of adopting future efforts to maintain this condition and suggests expanding the scope to other areas of the application and hiring an information security assessment service.

2. Background

Algorithm bias refers to tendencies or prejudices embedded in algorithms. These biases can arise from uneven data sets or subjective decisions by developers, resulting in biased outcomes and potential unfair discrimination. Algorithms in general are designed to process information and make decisions based on available data. However, if this data is biased, or if the algorithm itself contains biases, such circumstances may engender discriminatory or unfair outcomes.

For example, algorithms used in resume screening systems, credit ratings, court cases, and more can be subject to bias. If training data or algorithm rules favor certain groups over others, disparities and unfairness can occur. Therefore, it is critical to consider bias in any context where algorithms are used to make decisions. Becoming aware of bias and implementing practices to mitigate it are important to ensure fairness and impartiality in automated processes. Delivery apps, such as those used by food delivery companies or transportation services, often rely on algorithms to make decisions about the allocation of orders, the assignment of drivers or couriers, and even the pricing of services. However, these algorithms can be affected by bias.

An example of bias in delivery apps is the possibility of geographic discrimination. Algorithms that determine service availability or priority of service may be influenced by specific geographic characteristics, resulting in disparities in access to delivery services in certain regions. Another type of bias is that related to personal preferences. Algorithms can learn from user's order history data or ratings and, based on this data, make personalized recommendations. However, if this historical data is biased or reflects specific preferences of certain groups of users, the algorithm can perpetuate and amplify these biases by delivering disparate recommendations to distinct groups of individuals.

3. Objective

To address these issues, it is essential that delivery app companies are aware of the potential for bias in their algorithms and take steps to mitigate it. This may entail regularly analyzing the data, conducting fairness tests, being transparent in setting the rules, and adjusting the algorithms to ensure fairness, justice, and equity for both users and drivers or couriers involved in the delivery process.

Through this work, the objective is to audit the job management and allocation algorithm of the application called "AppJusto". This audit aims solely to verify and evidence the implementation of the work management and allocation algorithm, emphasizing the search for biases that may result in unequal results for workers from groups historically or presently facing some form of disadvantage.

---

4. Auditor cover letter

As an independent researcher, Heitor Gouvêa has no affiliations with companies or investors, which allows him to maintain a neutral stance while evaluating product manufacturers, to ensure that audits focus on quality and deliver results that exceed expectations; With a background in Software Engineering and 7 years of experience in the offensive security industry, Gouvêa's research focus is the discovery of vulnerabilities in applications/services and development of exploits - having already reported vulnerabilities for amazing companies like Activision, Nubank (and other Brazilian fintechs), Adobe, Oracle.

The author(s) received no financial support for the research, authorship, and/or publication of this article.

---

5. Scope

The audit period started on *07/14/2023* and ended on *07/21/2023*. In this particular time period, the code contained within the repository bears the following name: *"firebase-functions"* [5] was audited, which is understood as the back-end of the entire AppJusto application. All efforts were focused on algorithm flows in relation to job management and allocation, leaving out issues related to combating fraud, privacy and pricing of races.

---

6. Methodology

The code review approach was followed throughout the course of this project/work, for this purpose the read access to the code repository described in the "Scope" section was provided. Then, the search and validation of the statements was applied, which AppJusto communicates directly to its users through proprietary channels such as website and application.

7. Findings

In the table below, we have a list of the main test cases, i.e. the hypotheses that were elaborated and what were the results obtained through the manual verification of the code.

Hypothesis	Type	Result
The code responsible for allocating races may introduce discriminatory biases based on personal characteristics of users, such as gender, ethnicity, age or geographical origin.	Discriminatory allocation	Pass
The race allocation algorithm may result in under-representation of certain geographic regions, leading to longer waiting times or reduced availability of services in these areas.	Geographical under-representation	Pass
The code that selects drivers may introduce biases by preferring drivers based on non-performance-related characteristics such as gender, age or physical appearance.	Preference of drivers based on personal characteristics	Pass
The code may treat users unequally on the basis of personal characteristics, offering additional benefits or restrictions based on factors not relevant to the provision of the service.	Unequal treatment of users	Pass
The race allocation code may favor drivers near request locations, resulting in unequal access to services for more distant or underserved areas.	Preferential allocation by proximity	Pass
The allocation code may introduce biases by considering vehicle characteristics such as age, model or condition, resulting in unequal treatment between drivers based on these criteria.	Discrimination based on vehicle characteristics	Pass
The allocation code may give preference to users with high scores or ratings, potentially excluding or limiting opportunities for users with lower ratings, leading to unfair discrimination.	Prioritization of users based on scores	Pass
The code may not provide clear and understandable information on the criteria and processes used for the allocation of races, making it difficult to identify possible bias or discrimination.	Lack of transparency in the allocation algorithm	Pass
The race allocation code may result in inequity in serving different geographic regions, leading to longer waiting times or reduced availability of services in specific areas.	Geographical bias	Pass
The code may prioritize regular or frequent users over new users, resulting in unequal treatment and difficulties for users who are starting to use the service.	Prioritization bias of regular users	Pass

Table 1: description of the tests and classification of the results obtained.

8. Future work

Understanding that software is a changing artifact over time, it is important to emphasize that future efforts should be adopted so that AppJusto continues to acquire an increasing level of maturity in the aspects discussed here. The main recommendations are the following:

- The implementation of unit tests should be considered, so it will be possible to notice infringements still at the development stage, without affecting end users if something goes wrong;
- A policy of carrying out recurring audits, i.e. at least one within 12 months, should also be adopted;
- It is strongly recommended to expand the scope in future audits, for example in the flows: registration, driver approval and pricing;
- Hiring an information security assessment service.

9. Conclusion

Based on all the information presented here, we can affirm that at the present time AppJusto's job management and allocation algorithm is free of biases that may negatively impact workers from groups historically or currently at some kind of disadvantage. At the same time, we understand that there are still future efforts to be adopted by the development team so that the condition remains so, as mentioned in the "Future work" section.

10. References

1. https://appjusto.com.br
2. https://dev.appjusto.com.br/trabalho-decente
3. https://github.com/appjusto/firebase-functions
4. Standards for protecting at-risk groups in AI bias auditing: https://www.ibm.com/downloads/cas/DV4YNKZL
5. The algorithm audit: Scoring the algorithms that score us: https://philarchive.org/archive/DAVTAA-21