TepesAI

1. Contributions

FutureInternet[1] is a scholarly, peer-reviewed, open access journal on Internet technologies, being amongst the top websites where projects papers are published.

The most popular projects, similar to what TepesAI is demanding, are the projects where tax fraud is caught using neural networks.

A group of students from the African Center of Excellence in Data Science, University of Rwanda, managed to show that Artificial Neural Networks perform well in identifying tax fraud with an accuracy of 92% and a precision of 85%. One of the advantages provided by predictive models for tax fraud detection purposes consists of their utilization to calculate tax avoidance probabilities at the individual level. Their study also reveals that the time of the business that shows the difference in time from when a business was registered to the time of audit was also revealed to be a feature that is related to tax fraud.

Another group of Spanish students at Instituto de Estudios Fiscales, Universidad Rey Juan Carlos,Madrid, Spain did a research based on contributing to the detection of tax fraud concerning personal income tax returns (IRPF, in Spanish) filled in Spain, through the use of Machine Learning advanced predictive tools, by applying Multilayer Perceptron neural network (MLP) models.The results showed that the selected model has an efficiency rate of 84.3%, implying an improvement in relation to other models utilized at that time(2019) in tax fraud detection.

[1] https://www.mdpi.com/journal/futureinternet

Regarding the used technologies in these studies, it has been confirmed that neural networks offer low-cost algorithmic solutions and facilitate analysis, as it is not necessary to consider various statistical assumptions: Matrix homogeneity, normality, incorrect processing of data, and so on. Sigmoid, ReLu, softmax, softsign, linear, hard-sigmoid, softplus, and others were tested using the grid search to determine which activation function is best suited for income tax fraud detection. As for the model evaluation binary cross-entropy was used, as it compares predicted probabilities to actual classes by measuring the distance between the prediction and actual output. A confusion matrix is also used to check how the comparisons are displayed.

Moving on, our team will try to analyze these results, while trying to find similarities to TepesAI requirements, in order to proceed with choosing the best parameters and functions for gaining the highest possible accuracy.

2. Important names in the field, research teams

Since, to achieve this project's goal, we will be using web crawlers, it is important to mention the WebCrawler search engine, and its most important developer, Brian Pinkerton. Created in 1994, and originally a desktop application, WebCrawler was the first full-text crawler-based search engine. In 1996, it was the second most visited website on the web, and it frequently crashed due to server overloads, before being taken over by rivals Yahoo, Excite and others[2].

In 1998, alongside the effective launch of Google as a company, the Googlebot crawler debuted on the market. It's main task is crawling through all existing pages on the web to populate Google's search result pages (or SERP)[3]. It has a component for simulating a user on a desktop, and one for mobile devices. Googlebot accesses each site once every few seconds. It is designed to be run by thousands of machines simultaneously, to improve performance and scalability. It can crawl the first 15MB of any HTML file or other supported text files, while JavaScript, CSS and other resources such as videos will be fetched separately. After it has crawled the aforementioned 15MB, it stops, and considers the specific

[2] https://en.wikipedia.org/wiki/WebCrawler
[3] https://soft-surge.com/a-brief-history-of-web-crawlers/

page for indexing[4]. Since Google is the most used search engine globally, it can be argued that Googlebot is one of the most important implementations of a web crawler in the entire field, standing its ground as a leader since its inception in 1998.

3. Related Articles and books + Relevant links

- https://www.transparency.org/en/projects/integrity-watch-europe-online-tools-for-the-fight-against-political-corruption-in-europe
- https://www.giz.de/de/downloads/Blockchain_Anticorruption-2020.pdf
- https://www.u4.no/publications/are-blockchain-technologies-efficient-in-combatting-corruption.pdf
- https://www.u4.no/publications/artificial-intelligence-a-promising-anti-corruption-tool-in-development-settings
- https://ieeexplore.ieee.org/abstract/document/7881382/
- https://www.sciencedirect.com/science/article/abs/pii/0167487095000274
- https://academy.datawrapper.de/article/135-how-to-extract-data-out-of-pdfs
- https://ieeexplore.ieee.org/abstract/document/9183326
- Neural Networks and Deep Learning, A Textbook, by Charu C. Aggarwal
- Deep Learning with Python, by Francois Chollet
- Hands on Deep Learning Algorithms with Python, by Sudharsan Ravichandiran,
- Deep Learning (Adaptive Computation and Machine Learning Series), by Ian Goodfellow, Yoshua Bengio and Aaron Courville.

4. Resources and tools available

Some of the open-source tools and resources available online that we plan on using are:

I. Crawling / Data processing / AI Model creation & training:

- Flask[5]
- Selenium[6]
- Tesseract[7]

[4] https://developers.google.com/search/docs/crawling-indexing/googlebot
[5] https://flask.palletsprojects.com/en/2.2.x/
[6] https://selenium-python.readthedocs.io/
[7] https://github.com/tesseract-ocr/tesseract

- Slate[8]
- Tensorflow[9]

II. Data storage

- PostgreSQL[10] / MongoDB[11]

III. API Exposure

- Spring[12]

IV. UI/UX

- Angular[13] / React[14]

[8] https://pypi.org/project/slate/
[9] https://www.tensorflow.org/
[10] https://www.postgresql.org/
[11] https://www.mongodb.com/
[12] https://spring.io/
[13] https://angular.io/
[14] https://reactjs.org/

V. IDEs

- Intellij Idea[15]
- PyCharm[16]

This list is, however, merely an estimation at this point. Once we begin to implement components, we may need to use more resources than the ones mentioned above.

[15] https://www.jetbrains.com/idea/
[16] https://www.jetbrains.com/pycharm/