Anomaly Detection Techniques to Find Influential Users

Shawn L. Watts

Western Governors University

Table of Contents

[A. Project Highlights 3](#_Toc169687363)

[B. Project Execution 5](#_Toc169687364)

[C. Data Collection Process 8](#_Toc169687365)

[C.1 Advantages and Limitations of Data Set 9](#_Toc169687366)

[D. Data Extraction and Preparation 10](#_Toc169687367)

[E. Data Analysis Process 10](#_Toc169687368)

[E.1 Data Analysis Methods 11](#_Toc169687369)

[*Summary of Statistical and Machine Learning Methods* 11](#_Toc169687370)

[E.2 Advantages and Limitations of Tools and Techniques 13](#_Toc169687371)

[E.3 Application of Analytical Methods 16](#_Toc169687372)

[F Data Analysis Results 18](#_Toc169687373)

[F.1 Statistical Significance 18](#_Toc169687374)

[F.2 Practical Significance 19](#_Toc169687375)

[F.3 Overall Success 19](#_Toc169687376)

[G. Conclusion 19](#_Toc169687377)

[G.1 Summary of Conclusions 19](#_Toc169687378)

[G.2 Effective Storytelling 20](#_Toc169687379)

[G.3 Recommended Courses of Action 21](#_Toc169687380)

[H Panopto Presentation 21](#_Toc169687381)

[References 22](#_Toc169687382)

# A. Project Highlights

Which users are influencing others the most with their reviews and recommendations of video games on the Steam platform?

In this fictitious scenario based on real a real company, Valve Corporation and Steam gaming platform need help discovering user patterns in their recommendation and review systems data to better understand who their most influential users are. They have asked for a compact list of the most influential users for study by themselves and other game development companies that use the Steam platform to sell their games. The hope is that this will help game companies develop better games and marketing strategies to increase sales. Since this data is not known currently, they have asked to make some discovery and provide them a list of users with proper rational and methods. The value of this study could improve success for all parties.

The scope of the project will be limited to acquiring de-identified user data with minimal features. The only hard deliverable here is the final list of users in Excel format, no more than one thousand. If the client is not satisfied, a review process will ensue, and a new iteration will begin. Bringing in identifiable data and doing further analysis will be the responsibility of the client. These attributes are not provided and thus would be impossible to analyze.

Rapid Application Development was used as the project management strategy. Data was collected that was needed to complete the project. A CSV file was downloaded with the help of a browser from the Kaggle website. A WSL Ubuntu environment was used to isolate our workspace in Windows 11. Miniconda was utilized to create a python and Jupyter Notebook environment for performing and displaying the analysis. Standard data science python packages were used including, Pandas, Polars, Seaborn, Matplotlib, Scikit-Learn, and full list below:

* **Windows 11**: This is the main operating system of my laptop.
* **Windows Subsystem for Linux (WSL) with Ubuntu 20.04.6 LTS**: I want to isolate my process from my main OS. I will run the project on a Linux virtual machine.
* **Miniconda**: This is a free installer for a minimal version of conda. This is a package and environment management system. We will want to create our conda environment separate from any others. We will also need it to install Jupyter Notebooks and other python packages.
* **Python 3.10.14**: This is the python version we will use for this project.
* **GitHub**: GitHub is where we will keep all our work. We need some place to store our code and changes and be able to roll back to a previous version if needed.
* **Jupyter Notebook**: This tool helps present your data process, visualizations, and run most any python script needed for this project. Our EDA and final file creation will be all here.
* **Matplotlib**: For visualization of data.
* **NumPy**: It has many mathematical functions that will be useful for data analytics.
* **Pandas**: It helps read and store data in a structure that is ready to be consumed by python.
* **Scikit-learn**: This is a popular machine learning library for python, and we will use it for PCA, t-SNE, and Isolation Forests.
* **Seaborn**: This extends Matplotlib with a richer feature set and a much better presentation.
* **SciPy**: This has many uses. Among them is the test we plan to use to compare skewed distributions. Another is the bootstrapping we will use to simulate normal distributions for our skewed data.
* **Polars**: A faster implementation and alternative for large data sets. It will be used to import raw data.
* **OpenPyXL**: Needed to write our file to Excel. It extends Pandas and adds an Excel method to export the data frame.

Raw data was explored and assessed to understand what needed to be done. The data was summarized at the user level. Cleaning was completed after assessment. A Single PCA component was created over the relevant features and the right five percent of users were sampled based on the PCA component. An Isolation Forest model was created with the appropriate contamination rate set and features to obtain enough users for our final sample and label them as outliers or not. The Mann-Whitney U Test was used to compare the outliers with the non-outliers. Finally, each features’ distribution was independently compared to the sample visually and the result file created for the client.

# B. Project Execution

Rapid Application Development process was employed to deliver this project. Consisted of a design session, code, review, and iterations until client was satisfied and a sign-off was obtained.

The project was originally slated to take one hundred ours and start on June 24th. Instead, the client wanted to finish before the July 4th weekend, so the start time was moved up to June 10th. After starting the project, it was realized that the tasks were not as complicated as first thought and they took much less time than originally expected. The project was therefore completed in thirty and a half hours. Almost a third of the original estimate. The largest areas of variance were the design meeting and client review sessions. Also, there were no other iterations needed and the client signed off on the first draft. See the deliverables and timeline below.

*Project Goals, Objectives, and Deliverables*

* Goal 1: The goal is to select a set of users with the most influence among the others.
  + Objective 1.0: Client discussions centering around plan, design, and expectations.
    - Deliverable 1.0.1: Meeting notes and transcript with client.
    - Deliverable 1.0.2: Project plan and timelines.
  + Objective 1.1: Acquire the data.
    - Deliverable 1.1.1: The data will be downloaded in its original CSV format.
    - Deliverable 1.1.2: The data will be saved in parquet format for faster processing.
  + Objective 1.2: Analyze available features and clean the data.
    - Deliverable 1.2.1: Features will be selected with descriptive statistics, correlation, and basic understanding of the data structure used to get a sense of the data.
  + Objective 1.3: Sample enough data to run machine learning algorithms on.
    - Deliverable 1.3.1: A sample of data with the highest impactful users will be selected and stored in memory.
    - Deliverable 1.3.2: The sample will be much smaller than the original for faster processing during machine learning.
  + Objective 1.4: Reduce the data to two dimensions and visualize for an understanding of the outliers.
    - Deliverable 1.4.1: PCA will be used to reduce dimensions to two dimensions and a scatter plot displayed to show the pattern of our sample.
    - Deliverable 1.4.2: t-SNE will be used to reduce to two dimensions and a scatter be displayed of the results to see how it agrees with the PCA version.
  + Objective 1.5: Create an Isolation Forest model.
    - Deliverable 1.5.1: An isolation forest model shall be created using the relevant features and appropriate contamination parameter.
    - Deliverable 1.5.2: A pickle file shall be saved from the model to disk.
  + Objective 1.6: Select the outliers and ensure the count is below one thousand.
    - Deliverable 1.6.1: A dataset shall be created of the outliers predicted by the model.
    - Deliverable 1.6.2: The dataset shall be saved as an Excel file per the client.
  + Objective 1.7: Meet with clients and deliver the result.
    - Deliverable 1.7.1: Obtain sign-off on work or feedback about revisions.
    - Deliverable 1.7.2: Wrap up project or deliver a new plan based on feedback.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Milestone or deliverable | Duration  (hours or days) | Projected  start date | Anticipated  end date | Deviation |
| Deliverable 1.0.1 | 16 hours | *June 24th 2024* | *June 25th 2024* | *This project was moved up to start on June 10th. Only took 4 hours to meet with client.* |
| Deliverable 1.0.2 | 24 hours | *June 26th 2024* | *June 28th 2024* | *Project planning took less time. Only 4 hours.* |
| Deliverable 1.1.1 | 2 hours | *July 1st 2024* | *July 2nd 2024* | *File acquisition only took 0.5 hours.* |
| Deliverable 1.1.2 | 2 hours | *July 1st 2024* | *July 2nd 2024* | *Saving data to parquet only took 0.5 hours.* |
| Deliverable 1.2.1 | 8 hours | *July 1st 2024* | *July 2nd 2024* | *There was no deviation here. It was as expected.* |
| Deliverable 1.3.1 | 7.5 hours | *July 3rd 2024* | *July 3rd 2024* | *Sampling the data only took 2 hours.* |
| Deliverable 1.3.2 | 0.5 hours | *July 3rd 2024* | *July 3rd 2024* | *Verifying the sample was smaller did not deviate.* |
| Holiday | 0 hours | *July 4th 2024* | *July 7th 2024* | *Started earlier, should not factor in.* |
| Deliverable 1.4.1 | 4 hours | *July 8th 2024* | *July 8th 2024* | *PCA visual took 2 hours.* |
| Deliverable 1.4.2 | 4 hours | *July 8th 2024* | *July 8th 2024* | *t-SNE visual to 2 hours.* |
| Deliverable 1.5.1 | 15 hours | *July 9th* | *July 10th 2024* | *Isolation forest only took 4 hours to train and validate.* |
| Deliverable 1.5.2 | 1 hours | *July 10th 2024* | *July 10th 2024* | *Only took 0.5 hours to save to pickle file.* |
| Deliverable 1.6.1 | 6 hours | *July 11th 2024* | *July 11th 2024* | *Only took 2 hours to review the results.* |
| Deliverable 1.6.2 | 2 hours | *July 11th 2024* | *July 11th 2024* | *Only took 0.5 hours to save the final file.* |
| Deliverable 1.7.1 | 6 hours | *July 12th 2024* | *July 12th 2024* | *Meeting with clients was not necessary after sending them the results.* |
| Deliverable 1.7.2 | 2 hours | *July 12th 2024* | *July 12th 2024* | *No further planning was needed as the client signed off on the work.* |

# C. Data Collection Process

Recommendations.csv is a tabular file that has the most records which identify whether the user recommends the game, how many people found the recommendation helpful or funny, the review date, and number of hours user played the game. This is the file that we will be using for the project. The data was downloaded from Kaggle collected directly from the Steam website using screen scraping by the author. The CSV file was imported using Pandas and then saved as a parquet file in case needed to re-run the process.

This dataset has the number of recommendations made along with the number of products and whether the user recommended it or not. Along with that it has important measures for how many other people thought the recommendation was helpful and/or funny. This is appropriate because we are looking for those users that have the most influence.

The data is exceptionally clean and up to date. There are no missing values. It was curated well. There are a few issues that will need to be taken care of. One, there are values in the columns that are zero, and those rows should be removed. Two, some total hours played do not seem like they could be accurate. No other cleaning was different than planned and the items above were executed before running any processes. No other privacy, security, or governance issues came up that were unexpected.

## C.1 Advantages and Limitations of Data Set

One advantage of this data set is its shear coverage of data. It has been curated well and there are over fourteen million users that have been collected from the Steam website itself since the beginning of the platform. Another is that Steam captures pieces of information that are especially useful for analysis for this particular use case and that is the “helpful” and “funny” columns. Every time someone makes a comment, others can say whether they found it helpful and/or funny.

A disadvantage of this data is there is not much in the way identifiable information in which to glean even more insights. For example, we cannot stratify by gender or region. These would be great to characterize the users and potentially weed them out based on certain attributes.

# D. Data Extraction and Preparation

The data was downloaded from Kaggle [here](https://www.kaggle.com/datasets/antonkozyriev/game-recommendations-on-steam?resource=download&select=recommendations.csv) using a browser. It was then saved in a Jupyter Notebook project. The CSV file was uploaded using Pandas and then saved as a parquet file in case the process needed to be re-run. Pandas is a common library for loading data from many different file types. It can consume CSV and parquet. Parquet if processed very quickly compared to CSV and consumes much less space. Thus, the time to load is reduced and space saved.

After obtaining the data, it was examined using Pandas methods, info() and describe(). Info() produces the schema of the data along with missing values and types. Describe() produces a table of descriptive statistics for each numerical type. After viewing the described output, it was determined that any value with zero was to be removed. Also, hours played were too high in some cases, so those were removed as well. The data was then visualized for each column using the hist() function of Panadas. A histogram was rendered for each column and every distribution was skewed heavily right which aligns well with the mean and medians provided by the describe() function.

Finally, the data was summarized on all the columns to the user\_id level resulting in over fourteen million user records. After the summarization, the data was examined for the appropriateness of features and a PCA component was created to summarize those features. Then, a sample of the right most five percent selected based on that new column. At this point, the data is clean, selected, and ready to be further analyzed.

# E. Data Analysis Process

## E.1 Data Analysis Methods

Raw data was explored, summarized, and cleaned. A single PCA component was created from the relevant features. Everything right of the ninety fifth percentile was selected as our primary sample using the single PCA component.

An Isolation Forest algorithm was created with the appropriate contamination rate set and features to obtain enough users for our final sample and labeled them as outliers or not. The contamination rate we chose was appropriate to find enough outliers to satisfy our requirements.

Finally, the Mann-Whitney U Test was used to compare the outliers with the non-outliers. This was an appropriate tool to compare two skewed distributions. A p-value of zero was found and which was less than the alpha of .05 which gave compelling evidence in support of the alternative hypothesis, thus concluding analysis and delivering the users. For completeness, the test was run and visualized for each feature independently.

## *Summary of Statistical and Machine Learning Methods*

**Standardization**

* Unsupervised statistical method for scaling data. Fitted before use of PCA.

**PCA**

* Dimension reduction unsupervised.
* Principal Component Analysis algorithm
* Features included in the fitting of the PCA component will rely on correlation metrics of less than 0.60 and removal of features that do not make sense. Feature importance can be viewed to see how each feature is weighed. Social features were heavily weighted, and the others removed. The process was repeated once more to determine what the features were.
* No benchmark for this. In a different scenario, the number of components could be determined by the amount of variance explained by the components. This could be viewed using a scree plot, but this is not necessary for this application as only one PCA component was needed to decide the tail of the data.

**Percentile Method**

* n = (P/100) x N where P=percentile and N = number of values in the dataset sorted from greatest to least using our single PCA component.
* The percentile method was used to sample everything right-of and include the 95th percentile.
* The count of the sample size was measured to ensure it was large enough.
* Greater than 10,000 was sufficient.

**Isolation Forest**

* Unsupervised
* Isolation Forest algorithm
* The count of outliers is important to measure as the number of users to be returned is less than a thousand.
* Between 500 and 1,000 users should be sufficient. We returned ~700 users.

**Mann-Whitney U Test**

* The Mann-Whitney U test was used to validate that the sample of outliers was greater than the non-outliers.
* The metrics computed from the Mann-Whitney U test are the U-Statistic and the p-value.
* An alpha of 0.05 was used to measure this.
  + H0: There is no difference in the means between the outlier users from the non-outlier users.
  + H1: The outlier user means are greater than the non-outlier user means.

Standardization is recommended for PCA. PCA operates under the assumption that the data is normally distributed. The data was heavily skewed right. PCA is helpful here to gain a single value that maintained the relative positioning based on multiple features.

Once the user sample was selected, Isolation Forest was used to separate out the strong cases from the others. The contamination parameter was set appropriately to ensure we had enough users in the final output. Isolation Forests is a well-known method for discovering outliers in a dataset. PCA and t-SNE with two components was used to visualize the outlier vs non-outliers and it was observed there was a clear separation of the data. More importantly the outliers were more on the right side of the distribution. After verifying the distributions were different with eth Mann-Whitney U test, the data was bootstrapped to get a normal distribution, so the differences could be visualized.

## E.2 Advantages and Limitations of Tools and Techniques

*Tools*

* **Windows 11**
  + Advantage: Everyone is familiar with and has it
  + Disadvantages: Requires a license and not open source
* **Windows Subsystem for Linux (WSL) with Ubuntu 20.04.6 LTS**
  + Advantage: Great for isolating your main OS from your process thus not crashing the main OS.
  + Disadvantage: Steep learning curve.
* **Miniconda**:
  + Advantage: Great for creating separate environments for running different versions of python and packages.
  + Disadvantage: Miniconda is a bare bones approach and requires knowledge of what you need upfront. Conda could be a better choice as most all tools will be installed by default and has a user interface to simplify packages and environments.
* **Python 3.10.14**
  + Advantage: Very recognized as a data science platform.
  + Disadvantage: Requires programming skills and practice. This version may be missing some bug fixes, optimizations, or other helpful newer features.
* **GitHub**
  + Advantage: Great for storing, versioning, and sharing your work.
  + Disadvantage: Large file storage can cost, and the learning curve can be steep.
* **Jupyter Notebook**
  + Advantage: Very well recognized notebook environment and well documented.
  + Disadvantage: Knowing how to launch and manage this as a server can be challenging.
* **Matplotlib**
  + Advantage: Great for quick visualizations.
  + Disadvantage: Does not produce pretty output by default. Can take more work.
* **NumPy**
  + Advantage: Great for managing arrays and the math functions to support them.
  + Disadvantage: If only relying on this, then a lot of code would be required to manage.
* **Pandas**
  + Advantage: Easily load data from a variety of data source types.
  + Disadvantage: Can be slow with large data.
* **Scikit-Learn**
  + Advantage: Easy to implement classical machine learning.
  + Disadvantage: Not great for deep learning scenarios and can be slow with large datasets.
* **Seaborn**
  + Advantage: Produces nice looking visuals by default.
  + Disadvantage: Less control over finer details of the visual. Must use Matplotlib for that.
* **SciPy**
  + Advantage: Has many statistical functions.
  + Disadvantage: Some algorithms are slower when compared to other libraries.
* **Polars**
  + Advantage: Fast loading of data and manipulation.
  + Disadvantage: So different from Pandas that you must relearn data frames. Luckily there is a to\_pandas() function, but of course than the Pandas data frame will perform worse.
* **OpenPyXL**
  + Advantage: Easily export a data frame to an Excel file.
  + Disadvantage: Can be complex for inexperienced users and slow with larger datasets.

*Techniques*

* **Descriptive Statistics**
  + Advantage:Well defined and easy to implement.
  + Disadvantage:Can oversimplify results and sensitive to outliers.
* **Dimension Reduction**
  + Advantage: Summarize multiple features to less or a single dimension.
  + Disadvantage: Can potentially cause data loss which could cause results to be invalid.
* **Machine Learning**
  + Advantage: Can handle a lot of data and features and produce insights impossible for humans to do manually.
  + Disadvantage: Can be harder to interpret and training can take a while.
* **Bootstrapping**
  + Advantage: Can take an abnormal distribution and make it more normal.
  + Disadvantage: Can take a while to perform and ensuring you have enough samples is important and not the same for every problem.

## E.3 Application of Analytical Methods

*Steps*

1. Summarization is needed to aggregate the data to the user level. Spot checking counts is sufficient to validate that this is working.
2. PCA is required to be able to sort and select high end users. It requires n\_components to be set, but this is already determined to be one as that would be sufficient to represent the magnitude of the notable features. Data is standardized for each feature prior to running PCA.
3. Data needs to be sampled and reduced before feeding into the Isolation Forest algorithm. A simple method of ranking on the single PCA created and selecting the right five percent is employed using the percentile method.
4. Isolation Forest is implemented to find outliers in the sample. Some basic arithmetic is performed to ensure that the contamination parameter was high enough to select enough users.
5. PCA and t-SNE are used to reduce the dimensions to two for visualizing the differences between outlier and non-outliers. They both take a parameter of n\_component equal to two, so that a scatter plot could be displayed for each.
6. A function is created to compare two distributions which bootstrap both distributions for the mean with confidence\_level equal to 0.9 and n\_resamples equal to 20,000. The number of samples are verified by the shape of the graphs which should be normal in appearance. Each distribution is then plotted as a histogram on the same graph and the confidence interval printed beneath to compare. Each column is run through this function and thus six graphs are rendered to ensure each outlier is on average higher than non-outliers. Since this is a visual, there was not much validation other than visual assessment. However, the confidence intervals below the graph show that the highs and lows do not overlap for each distribution giving even more evidence.
7. For our last check, the Mann-Whitney U test is performed on each feature for outlier and non-outliers. This test simply takes two distributions and produces a p-value. If the p-value meets the threshold, the null hypothesis can be rejected or accepted.

# F Data Analysis Results

## F.1 Statistical Significance

**Isolation Forest**

* Unsupervised
* Isolation Forest algorithm
* The count of the outliers
* Between 500 and 1,000 users was the benchmark and 721 were found and were within range.

**Mann-Whitney U Test**

* The Mann-Whitney U test was used to validate that the sample of outliers is greater than the non-outliers.
* The metrics computed from the Mann-Whitney U test are the U-Statistic and the p-value.
* An alpha of 0.05 was set to measure this:
  + H0: There is no difference in the means between the outlier users from the non-outlier users.
  + H1: The outlier user means are greater than the non-outlier user means.
* Each feature was measured and found to have a p-value of ~0.

There was sufficient evidence for every feature and thus the null hypothesis was rejected and the alternative accepted that the outlier user means were greater than the non-outlier user means. With a count of outlier users within range, the final list was validated.

## F.2 Practical Significance

The solution will provide answers the client is expecting. It will reduce the sample users to a size that the client can handle and can begin marketing to. Follow-up metrics by the client can provide insight into the success of marketing efforts. PCA and Isolation Forests are well documented and are used in many applications and thus proven. Comparing the outlier with non-outlier distributions is widespread practice and can be evaluated using the Mann-Whitney U Test. If needed, the client could use this process again to grab the next set of users. Simply perform exact operations except filtering out the first batch of outliers.

## F.3 Overall Success

For this project to be a success, the number of influential users delivered must be one thousand or less. The users should be the most significant and most impactful. The final dataset is evaluated using the Mann-Whitney U test to determine if the outliers are greater than the non-outliers with a p-value of less than the alpha 0.05. This will be the right tail version of the test.

This project was efficient at identifying users believed to be of high value to the client. Visuals corroborated expectations a gave insights into the direction of the analysis. The tests were appropriate and yielded expected results.

# G. Conclusion

## G.1 Summary of Conclusions

Overall, the project produced the data the client is looking for. The Isolation Forest algorithm selected enough outliers to fit the final criteria, 721, and all features came in under the 0.05 alpha for the Mann-Whitney U test. All visuals gave insight to the accuracy of the data as well. It was thus concluded that the list delivered to the client met expectations.

## G.2 Effective Storytelling

* **Histogram**
  + Histograms are used to view each variable and what the distribution looks like. This informs how skewed or normal the distribution is. The hist() function of Pandas is used to create a histogram of all the columns. All columns are noticeably heavily skewed right. This expects appropriate tests for comparing skewed distributions later.
* **Correlation Plot**
  + A correlation plots of all the features show the co-linearity of features. The graph is analyzed and adjusted as needed to the final selection. All correlation coefficients should be at least under 0.60 which is the benchmark for this project. This process is repeated until co-linearity has been reduced appropriately based on domain knowledge and the correlation coefficients. The built in corr() function of Pandas styled as a heatmap is used to render this visual.
* **Scatter Plot**
  + Scatter plots are created to view the components of the PCA and t-SNE models trained for two components. Color codes for outlier and non-outliers will be generated. The Seaborn scatter plot function is used to create the scatter plots of the components.
* **Side-by-side bar graphs of Bootstrapped means**
  + The Matplotlib hist() function is used to graph the outlier and non-outliers as histograms on the same graph for comparison. Seeing the difference between these distributions demonstrated how much greater each outlier user is to the non-outliers.

## G.3 Recommended Courses of Action

Follow-up should be done by the client. Each user should be profiled using all appropriate dimensions the client has for each user and classified for campaigns. This may require more machine learning, domain knowledge, or a little of both. This should be the next step for the client as the data that was used was inadequate to segment the users. This will be important for informing messaging, and approaching the users with product, offerings, or even studying user behavior. If the client provides more data, than the project could be re-run with more features and yield even better results in finding the most impactful users.

Next, the client should perform this on the next set of users as well. A repeat of this process should yield similar traits between the two samples and can be beneficial in reaching more users. It is understood that this is a first attempt at approaching these users, so it would be appropriate to expand this base once the first recommendation has been executed and tested. The process will yield the next most influential users in the data. Since data is constantly being added, running monthly or weekly could be appropriate as well.

# H Panopto Presentation

[Panopto Presentation - D502 - Shawn Watts](https://wgu.hosted.panopto.com/Panopto/Pages/Viewer.aspx?id=5e83a62d-ffd5-4dac-b71e-b194014cd2c8)

 

# References

Kozyriev, Anton. (2023). Game Recommendations on Steam [Data set]. Kaggle. <https://doi.org/10.34740/KAGGLE/DS/2871694> Scribbr. (2022, December 21).

Glackin, C. E. W., & Adivar, M. (2023). Using the power of machine learning in sales research: process and potential. Journal of Personal Selling & Sales Management, 43(3), 178–194.

Elbaghazaoui, B. E., Amnai, M., & Fakhri, Y. (2022). Data Profiling and Machine Learning to Identify Influencers from Social Media Platforms. Journal of ICT Standardization, 10(2), 201–218.