

DataFlippers : A drag and drop interface to build machine learning models

Charles Bennington ¹

¹ Department of Computer Science, Gonzaga University, USA

DOI: [10.xxxxxx/draft](https://doi.org/10.xxxxxx/draft)

Software

- [Review](#) 
- [Repository](#) 
- [Archive](#) 

Editor: [Open Journals](#) 

Reviewers:

- [@openjournals](#)

Submitted: 01 January 1970

Published: unpublished

License

Authors of papers retain copyright and release the work under a Creative Commons Attribution 4.0 International License ([CC BY 4.0](#))

Summary

Machine learning is often taught at the upper undergraduate levels, with programming often cited as a prerequisite for learning Machine learning([Bart et al., 2016](#); [Brunner & Kim, 2016](#); [Thayer, 2020](#)). This creates an barrier to entry that bars some novices, who want to learn Machine learning, but are unable to do so due to a lack of programming knowledge. DataFlippers aims to teach novices the core concepts of statistical machine learning through a intuitive drag and drop interface inspired by scratch([Resnick et al., 2009](#)).

Statement of need

Programming as a barrier to machine learning

Many educators cite that programming is a barrier to entry to machine learning ([Bart et al., 2016](#); [Brunner & Kim, 2016](#); [Thayer, 2020](#)). This is because many tools and libraries for machine learning are called programmatically. Building a GUI to interface with theses tools would allow for novices to learn machine learning, without the pre-requisite of knowing how to code.

Machine learning literacy

This lack of accessible entry points limits the potential for widespread machine learning literacy. As AI increasingly permeates various aspects of modern life understanding its underlying principles becomes essential. AI literacy empowers individuals to critically evaluate these systems, fostering informed decision-making and promoting responsible technological development([Jain, 2021](#); [Provost & Fawcett, 2013](#)). Moreover, a basic grasp of AI models can demystify complex technologies, enabling students to navigate a world shaped by intelligent systems with greater confidence and agency.([Hsu & Chen, 2025](#))

Therefore, there's an urgent need for tools that prioritize accessibility and intuitive learning. A low barrier to entry is paramount; users should be able to explore core data science concepts without needing prior programming experience. This necessitates a paradigm shift away from code-centric approaches towards user-friendly interfaces that abstract the complexities of programming while preserving the fundamental principles of data analysis.

State of the field

There are several no-code, low code platforms available on the internet. However these are often designed with power users in mind, and are often expensive, making them frustrating to novices. Many of these softwares are geared toward commercial data science use, with a focus on integration with common business tools.

Name	Description	Target Audience	License	Drag and drop
DataBricks	Generative AI	Businesses	Paid / Commercial	No
Power BI	Visualization Interface	Businesses	Paid / Commercial	Yes
Rapid Miner	Training / Visualization	Data Scientists	Free for individuals	No
JASP	Statistics / Visualization	Upper Undergraduate / Graduate	Free	No

37 The platforms that are free and tailored to students, such as JASP(JASP Team, 2025) is
 38 catered to teaching students statistics, rather than expressly teaching them machine learning.

39 Software design

40 Inspiration

41 Scratch(Resnick et al., 2009), a visual programming language designed for children, offers a
 42 compelling model for accessible computational learning. Its intuitive drag-and-drop interface
 43 allows beginners to grasp fundamental programming concepts without needing to decipher
 44 complex syntax. Scratch's interface has been proven to be effective at teaching novices
 45 programming concepts, and assist learners when the transition to "real" programming(Armoni
 46 et al., 2015).

47 Scratch was constructed on 4 pillars, which makes up it's main philosophy. DataFlippers
 48 follows these pillars closely. 1. **Passion** Learners are more motivated when they are learning
 49 things that they care about. To follow this pillar, DataFlippers uses crafted example datasets,
 50 which focus on topics that are intriguing to children. This is discussed more in the educational
 51 materials section. 2. **Peers** DataFlippers is intended to be shared with friends. For this
 52 purpose DataFlippers has it's own file type, and every chart and graph has the ability to be
 53 inspected and saved and shared. 3. **Play** The drag a drop nature implies a certain type of
 54 play. This makes is easy to novices to quickly try things, without the overhead of writing code.
 55 4. **Projects** DataFlippers is focused around datasets, allowing for novices to center machine
 56 learning projects around a specific dataset.

57 Drag and drop design

58 The drag and drop interface is designed to reduce the syntactic complexity of programming
 59 down to drag and drop blocks, with the layout and design imitating underlying python libraries.
 60 This ensures that users have an low floor to learning, while also paving the way for them to
 61 transition to writing code later.

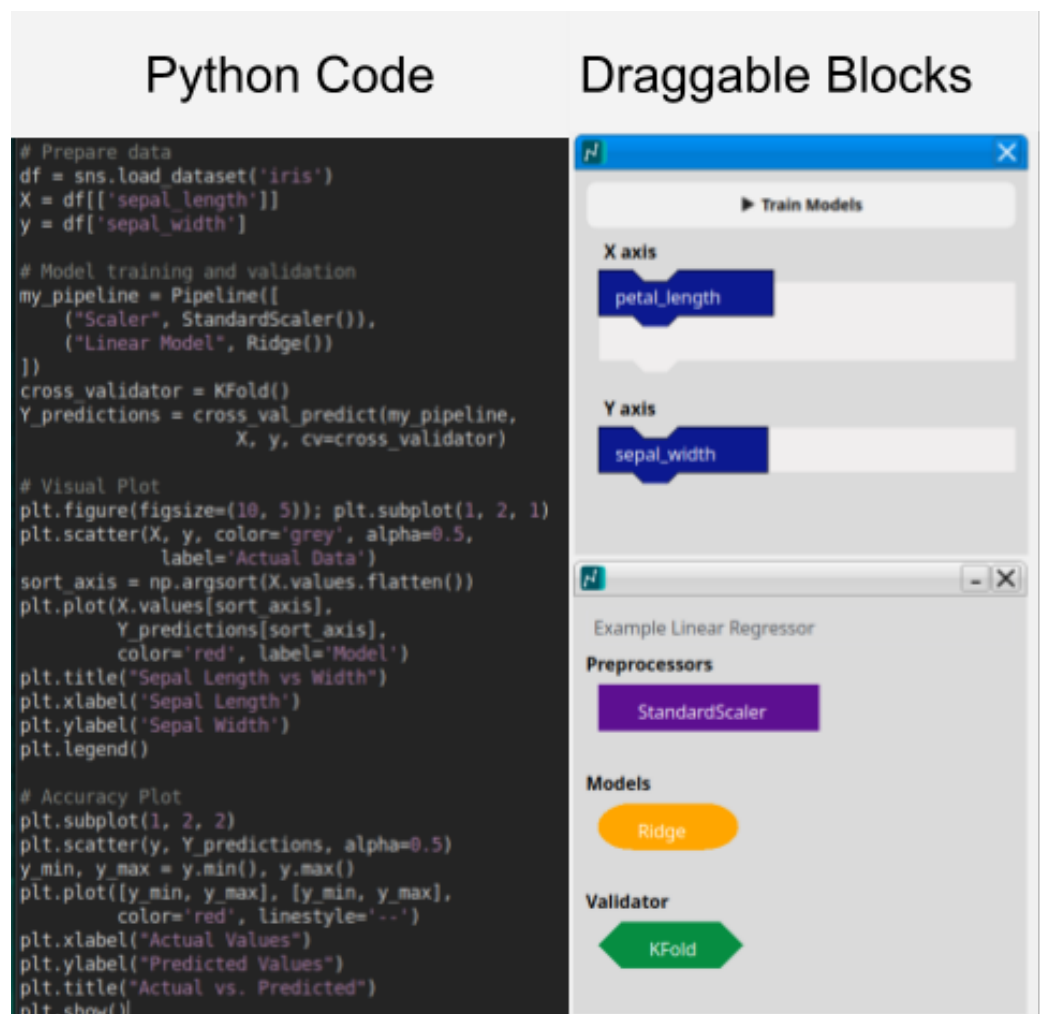


Figure 1: Image showing the difference between python blocks, and the drag and drop DataFlippers interface. Both the code and the DataFlippers blocks result in the same thing, however the blocks are much easier for novices to understand.

Each drag and drop block is modeled after basic shapes, this gives the user visual signifier, indicating where each block should be dropped on the interface. This ensures the interface is intuitive to people without machine learning experience.

Training multiple models at once is a core feature, allowing for quick model comparison. This enables common user desires within data science, where data scientists often compare and contrast models. Another purpose of this feature is to allow users to learn the differences between certain models.

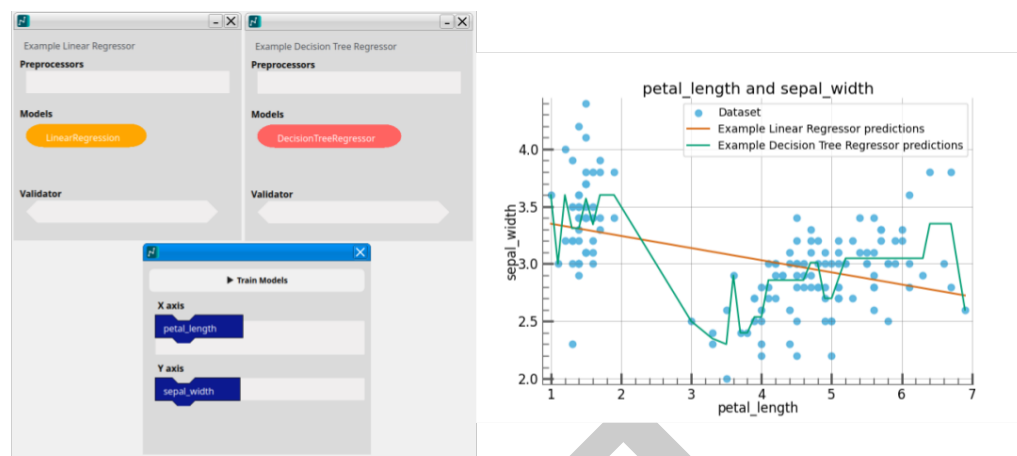


Figure 2: Image showing the model comparison. This would allow novices to understand and visually see the differences between different types of machine learning models. This example shows a comparison between a DecisionTreeRegressor, and a LinearRegressor.

69 Users are also enabled to perform basic statistical analysis, this can be done by removing all of
70 the pipeline blocks, and only plotting via the “Inputs and Outputs” block. The software also
71 automatically runs a descriptive statistics plotting on all columns inputted, whenever a model
72 training job is submitted.

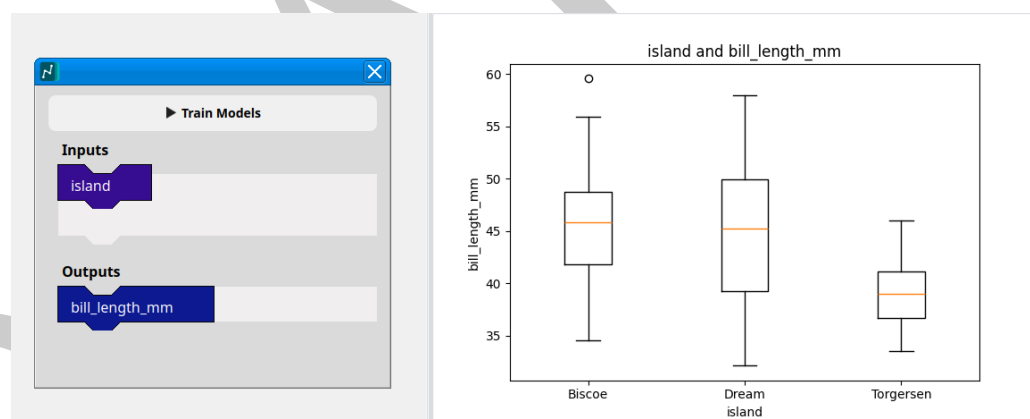


Figure 3: Image showing the basic statistical capabilities of the DataFlippers framework. This shows the island type versus the bill length on the penguins.

73 The interface also allows the user to input manual predictions, allowing for novices to interact
74 with their newly created AI models. This tab enables the user to export their saved models as
75 software, which is where a user can save their trained model, and access it later. DataFlippers
76 also enables the exporting as pickle, which fulfills the needs of potential power users, by
77 allowing them to interface with the python object directly, if desired.

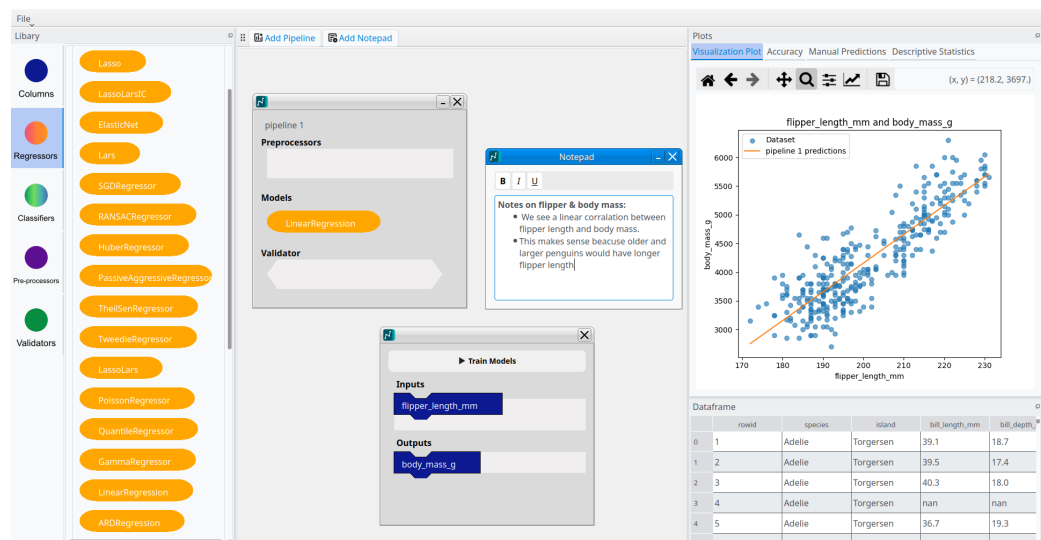


Figure 4: Image of the fullscreen DataFlippers project page. The example dataset here is the penguins dataset.

Software architecture and core project libraries

The language for this software is python, this is because python possesses several libraries, such as pandas (team, 2020), matplotlib (Hunter, 2007), and scikit-learn [Pedregosa et al. (2011);], which are standard tools for teaching machine learning (Burridge & Fekete, 2022).

The core machine learning models are provided by scikit-learn [Pedregosa et al. (2011);]. There are several reasons for choosing scikit-learn, one of them is portability. scikit-learn is very common within the entry level machine learning field, so this means that skills that novice learn from DataFlippers could translate easily to programmatic skills, if the novice decides to learn programming. Additionally, the scikit-learn is very friendly for first time users, featuring extensive documentation, which can be viewed by hovering on each items tooltip. Pandas (team, 2020) is used for handling the data, this is because pandas supports the importing of multiple standard filetype formats, such as csv, excel, and parquet. Matplotlib is utilized for the plotting features, this is because matplotlib interfaces well with the PyQt background, as well as being another standard data science library.

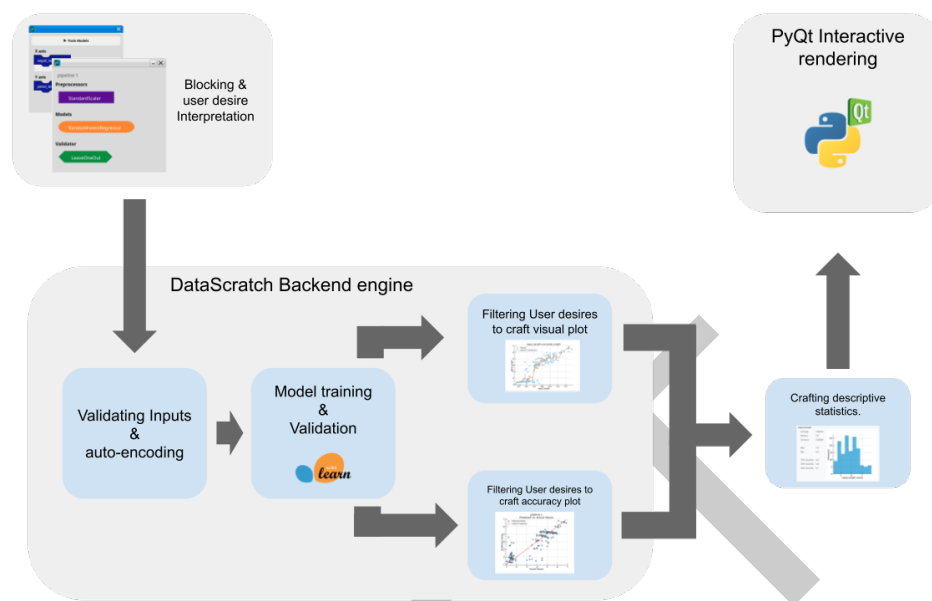


Figure 5: Diagram of the overall software architecture.

Research impact statement

Installable executables

The current codebase also supports the automatic building of installers for both windows and linux. In linux, this is in the form of a debian package, and for windows, it is the form of a executable. This shows near term significance in the form of reproducible materials. These are both available from the main DataFlippers page.

User documentation and educational materials

The main DataFlippers page features a walk thought tutorial on how to use the software, and a brief explanation of the the different components on the screen.

Dataframe Basics

Each “row” in the dataset corresponds to an item in that dataset. Since this example is using the `penguins` dataset, each row corresponds to a penguin.

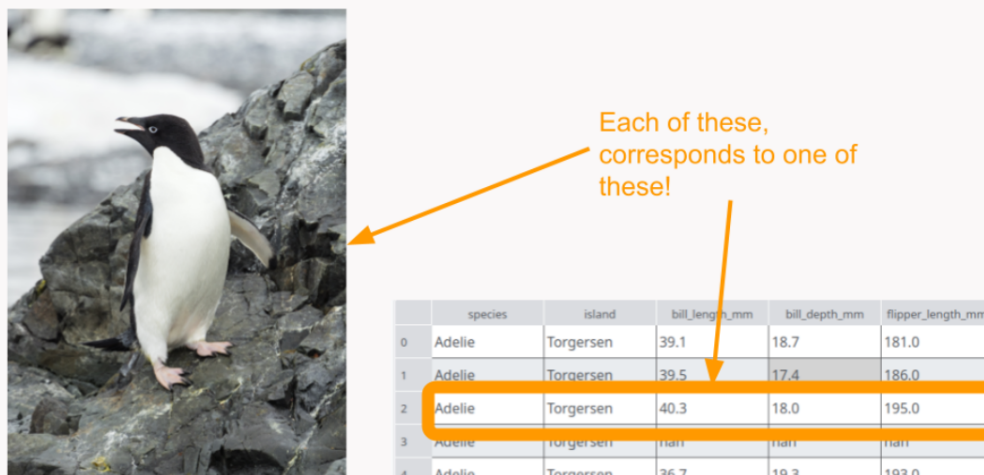


Figure 6: A screenshot from the DataFlippers website “My first AI model”.

Guided active learning greatly improves student engagement (Rossi et al., 2021), to accommodate this desire DataScratch comes pre-loaded with several example datasets, which have been crafted to be usable to a wide range of users, allowing novices to get learning right away, without having to procure a dataset first. These datasets are a blend of commonly used data science teaching datasets such as `palmer_penguins` (Horst et al., 2020), `diamonds` dataset (Waskom, 2021) and the `iris` dataset (Fisher, 1936), as well as datasets that would be enticing to a younger audience, such as a dataset containing information about `pokemon` (Banik, 2018) and a dataset about `minecraft` biome statistics (Pilot, 2026). These example datasets come with 3-5 lab questions each, which can be accessed from the main DataFlippers website. The lab questions can be categorized as the following.

- Questions encouraging novices to be critical of the models they are creating. Below are some examples.
 - Make a box plot to compare the gender of the waiter versus the total tip. Do men or women get more tips? Does this seem fair?
 - Using the model constructed in step 1, do a single prediction, compare a total bill of 200\$ if the waiter is Male versus Female. Does your model have gender bias? Discuss whether the model's behavior could be categorized as sexist.
- Questions asking typical data science inquiries. Below are some examples.
 - What is the relationship between `flipper_length_mm` and `body_mass_g`?
 - Create a linear AI model that takes in `flipper_length_mm` and `body_mass_g` as inputs, and predicts the gender of the penguins

AI usage disclosure

~95% of this project was written by humans, and ~5% would be what is consider “AI assisted”. Google was used to search for API documentation. The built in AI overview on google cannot be deactivated, and thus the AI overview was used upon each google search. Oftentimes, the google AI overview provided false information, and was ignored in the later stages of the project due to a lack of verifiability. Generative AI was later used during the proofreading stage of writing the paper, with it helping catch several grammatical errors.

References

- Armoni, M., Meerbaum-Salant, O., & Ben-Ari, M. (2015). From scratch to “real” programming. *ACM Transactions on Computing Education (TOCE)*, 14(4), 1–15.
- Banik, R. (2018). *The complete pokemon dataset*. Kaggle. <https://www.kaggle.com/datasets/rounakbanik/pokemon?resource=download>
- Bart, A. C., Tibau, J., Tilevich, E., Shaffer, C. A., & Kafura, D. G. (2016). Implementing an open-access, data science programming environment for learners. *COMPSAC*, 728–737.
- Brunner, R. J., & Kim, E. J. (2016). Teaching data science. *Procedia Computer Science*, 80, 1947–1956.
- Burridge, J., & Fekete, A. (2022). Teaching programming for first-year data science. *Proceedings of the 27th ACM Conference on on Innovation and Technology in Computer Science Education Vol. 1*, 297–303.
- Fisher, R. A. (1936). *Iris*. UCI Machine Learning Repository.
- Horst, A. M., Hill, A. P., & Gorman, K. B. (2020). *Palmerpenguins: Palmer archipelago (antarctica) penguin data*. <https://doi.org/10.5281/zenodo.3960218>
- Hsu, T.-C., & Chen, M.-S. (2025). Effects of students using different learning approaches for learning computational thinking and AI applications. *Education and Information Technologies*, 30(6), 7549–7571.
- Hunter, J. D. (2007). Matplotlib: A 2D graphics environment. *Computing in Science & Engineering*, 9(3), 90–95. <https://doi.org/10.1109/MCSE.2007.55>
- Jain, L. C. (2021). *Smart innovation, systems and technologies*. Springer: Singapore.
- JASP Team. (2025). *JASP (Version 0.95.4)[Computer software]*. <https://jasp-stats.org/>
- Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., & Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12, 2825–2830.
- Pi1lot. (2026). *Minecraft-chunks*. Kaggle. <https://www.kaggle.com/datasets/pi1lot/minecraft-chunks?resource=download>
- Provost, F., & Fawcett, T. (2013). Data science and its relationship to big data and data-driven decision making. *Big Data*, 1(1), 51–59.
- Resnick, M., Maloney, J., Monroy-Hernández, A., Rusk, N., Eastmond, E., Brennan, K., Millner, A., Rosenbaum, E., Silver, J., Silverman, B., & others. (2009). Scratch: Programming for all. *Communications of the ACM*, 52(11), 60–67.
- Rossi, I. V., Lima, J. D. de, Sabatke, B., Nunes, M. A. F., Ramirez, G. E., & Ramirez, M. I. (2021). Active learning tools improve the learning outcomes, scientific attitude, and critical thinking in higher education: Experiences in an online course during the COVID-19 pandemic. *Biochemistry and Molecular Biology Education*, 49(6), 888–903.
- team, T. pandas development. (2020). *Pandas-dev/pandas: pandas (latest)*. Zenodo. <https://doi.org/10.5281/zenodo.3509134>
- Thayer, K. M. (2020). *Practical knowledge barriers in professional programming* [PhD thesis].
- Waskom, M. L. (2021). Seaborn: Statistical data visualization. *Journal of Open Source Software*, 6(60), 3021. <https://doi.org/10.21105/joss.03021>