Innovating Feature Selection in Data Science

by
Nikita Rubocki

A THESIS

submitted to

Oregon State University

Honors College

in partial fulfillment of
the requirements for the
degree of

Honors Baccalaureate of Science in Computer Science
(Honors Associate)

Presented April 14, 2022
Commencement June 2022

AN ABSTRACT OF THE THESIS OF

Nikita Rubocki for the degree of Honors Baccalaureate of Science in Computer Science presented on April 14, 2022. Title: Innovating Feature Selection in Data Science

Abstract approved:_____

Aimee Lougee

Data science is a rapidly growing industry permeating throughout every aspect of society. Everything collects data these days, and people use this data to find meaningful patterns leading to benefits ranging from more intuitive marketing to better cancer detection. However, increased data collection also leads to increased complexity, and data science works to manage this complexity through various techniques and machine learning/artificial intelligence models. But data science faces two significant issues: too many features in a dataset and long model training times. To help combat these issues, the author developed a tool called Ensemble Feature Importance Ranker (EFIR). This paper analyzes the accuracies and limitations of this innovative tool through a series of experiments on linear regression datasets. Preliminary results and metrics show high accuracy in finding the most impactful features, overall proving that EFIR identifies the key features in linear regression datasets. In short, EFIR leads to better data and faster model training times under various conditions.

Key Words: data science, machine learning, artificial intelligence, feature selection, feature importance, dataset analysis, ensemble feature importance ranker, EFIR

Corresponding e-mail address: rubockin@oregonstate.edu

Innovating Feature Selection in Data Science

by
Nikita Rubocki

A THESIS

submitted to

Oregon State University

Honors College

in partial fulfillment of
the requirements for the
degree of

Honors Baccalaureate of Science in Computer Science
(Honors Associate)

Presented April 14, 2022
Commencement June 2022

Honors Baccalaureate of Science in Computer Science project of Nikita Rubocki presented on April 14, 2022.

APPROVED:

_____

Aimee Lougee, Mentor, representing Data Science and Analytics

_____

Yong Bakos, Committee Member, representing Computer Science

_____

Jill Hubbard, Committee Member, representing Computer Science

_____

Toni Doolen, Dean, Oregon State University Honors College

I understand that my project will become part of the permanent collection of Oregon State University, Honors College. My signature below authorizes release of my project to any reader upon request.

_____

Nikita Rubocki, Author

Table of Contents

Abstract

Data science is a rapidly growing industry permeating throughout every aspect of society. Everything collects data these days, and people use this data to find meaningful patterns leading to benefits ranging from more intuitive marketing to better cancer detection. However, increased data collection also leads to increased complexity, and data science works to manage this complexity through various techniques and machine learning/artificial intelligence models. But data science faces two significant issues: too many features in a dataset and long model training times. To help combat these issues, the author developed a tool called Ensemble Feature Importance Ranker (EFIR). This paper analyzes the accuracies and limitations of this innovative tool through a series of experiments on linear regression datasets. Preliminary results and metrics show high accuracy in finding the most impactful features, overall proving that EFIR identifies the key features in linear regression datasets. In short, EFIR leads to better data and faster model training times under various conditions.

1 Introduction

The world has exploded with information, and we are collecting and analyzing this information at an unprecedented scale. Known as Big Data, this growing industry is worth over $70 billion and produces around 2 exabytes of information *daily* (that's 2,000 trillion bytes)[1]. Data accumulation is everywhere: computers, phones, cookies, social media, newsfeeds, emails, at-home assistants, wearables, etc. all gather information [2]. Even new dishwashers come with data collection capabilities [3].

The sheer amount of data generated every day is unfathomable, bringing with it intricate and insightful trends hidden within the noise. Because the human mind simply cannot comprehend or understand nuances within this sea of data, we created complex and "intelligent" computers to understand it for us. This is the birthplace of machine learning (ML) and artificial intelligence (AI)[4]. By possessing all this data, we hold the potential to estimate trends and manage predictions, but accurately finding meaningful conclusions from data is difficult. To tackle these challenges, an emerging industry using ML/AI models rapidly grew into mainstream technology. Today, we call this industry data science.

Data science faces two main problems: too many features within a dataset and long model training times [5], [6]. The first problem manifests in a myriad of ways. A dataset "feature" refers to a column in a dataset, which can be almost anything: height, weight, median income, etc. (as opposed to the rows, or "observations", of a dataset). More features lead to less understanding because complexity inherently increases. Though ML models assist in analyzing these complex problems, a tradeoff exists between a model's complexity and interpretability, leading to all sorts of real-world problems [7]–[10]. In other words, the more complex a dataset, the less likely a model will find meaningful trends within the dataset, leading to mediocre performance. Furthermore, including too many features causes models to become unsolvable. Known as the "curse of dimensionality", this phenomena causes model and numerical instabilities, causing unusable models [11]. The feature problem is also exacerbated because most collected data contains noisy, dirty, irrelevant, or redundant features, causing further drops in performance [12].

The second problem data science faces is long model training times, which directly correlates to the problem of too many features due to simple physics. Because of finite hardware capacity, ML models can only analyze a certain amount of data at a time. The more data fed to a ML model, the longer it takes to learn the data. Though there are many companies in the space working to optimize and increase hardware and computational capacity, data accumulation is exponentially increasing while continued optimization yields diminishing returns [13]. Furthermore, increased model times lead to delays in results and increase the drain on limited energy resources.

While companies focused on making bigger and faster models, data scientists turned towards making the data itself cleaner and leaner using various "feature selection" techniques. Feature selection encompasses a family of methods geared towards reducing the dimensionality of the feature space. In other words, they attempt to decrease the number of features necessary to train a model. Though useful, these processes often fall

short of expectations and only work in certain situations. The Background section of this paper covers this topic in more detail.

For now, know that many feature selection processes are too specialized and limited in their capacity to find the key features in a dataset. A new data science tool known as Ensemble Feature Important Ranking (EFIR) aims to change that. The author developed EFIR to better understand how certain features contributed to the outcomes of a model. This tool uses an ensemble approach to help users identify the most important features in a dataset and therefore discard irrelevant features. These insights alleviate the issues associated with high dimensional feature spaces (i.e., too many features in a dataset), leading to more effective datasets and faster model training times. As such, EFIR addresses the above data science problems by facilitating leaner, more understandable datasets, and therefore better ML models with faster training times.

The following paper details how EFIR helps solve these major problems and assesses the true efficacy of the tool on linear regression datasets. First, the Background provides a brief overview of feature selection methods and explains the origins of EFIR. Then, the Methods section goes over the datasets and processes used to evaluate the thesis. Finally, the paper concludes with an analysis of the results and a conclusion on the effectiveness of the tool.

2 Background

Before diving into the specifics of EFIR, let us look at similar methods and understand why innovation in the space is necessary.

2.1 Feature Selection

As previously stated, creating faster ML/AI models is becoming increasingly difficult while ML/AI chip advances only gain marginal returns due to hardware limits [13]. The rate of data growth is simply outpacing hardware innovation and how much data can flow through the physical components of a computer. Because of these physical constraints, datasets became another target in reducing long model learning times. Plus, dirty datasets lead to poor model performance regardless [14]. Therefore, efficiently leaning and analyzing datasets is critical to the data science industry.

The inherent problem with dataset analysis is determining which features are important to an ML model. In other words, given a target label (the goal a model learns), a user wants to know which features correlate most strongly with the target. For instance, imagine a dataset on lung cancer where each row contains data about a patient and the target label shows whether the patient has cancer. Doctors would like to know which features, such as height, weight, smoking amount, etc. are most strongly linked with a positive cancer outcome. In this case, height is probably unimportant, while the amount the patient smokes is likely indicative of cancer. As such, features that do not correlate to the target (i.e., “noisy” features) are removeable. In general, this improves model accuracy, lowers training time, and increases model generalizability [15].

The main method of identifying contributing features in a dataset is called feature selection [16]–[18]. Many ML methods/approaches come with the ability to perform feature selection, including Lasso/Ridge Regression [19], Information Gain [20], Relief [21], Fisher Score [22], Elastic Net [23], and many more. There are many studies emphasizing the use of feature selection methods for improving models for a variety of different problems [24]–[27].

2.2 Too Many Solutions

Though detailing each of the mentioned techniques goes beyond the scope of this paper, it is important to acknowledge each method works in a specific domain. For example, linear regression models use Lasso Regression. Though it is very adept at this job, Lasso Regression falls short in solving problems where no linear relationship exists between the input features and the target label. As another example, decision trees such as Information Gain suffer from feature selection bias where features with many values are favored over those with few values [28]. In other words, a feature such as height, with varying values, would gain precedence over a binary feature, which only has two possible options, simply because height has more variety.

In short, there are many different ways of selecting features, even within one single algorithm [29]. However, though no single method works for all problems (see the "no free lunch theorem" [30]), data science champions generalizability. Current feature selection methods cannot meet this task, but a new method would work by choosing important features from any given range of datasets with varying trends and features. One possible solution is an ensemble approach [31]. The ensemble method relies on the idea of "majority rules," where many inputs are considered and weighed for an overall voted output. Many ensemble methods show model prediction improvement and solve an array of challenges [32]–[35].

Through an ensemble feature selection program, features could be ranked in order of importance. A feature is important if it highly relates to the target label and unimportant if no relation exists. Though other studies and methods explore various feature importance and ranking techniques [36]–[38], no study or tool combines different, opposing methods of feature selection in an ensemble fashion. As such, EFIR was created under the assumption that including multiple feature selection approaches would allow coverage across a variety of models, capturing different decision boundaries into one single, more accurate solution.

2.3 Ensemble Feature Important Ranker (EFIR)

If ensemble methods improve model performance [31], it is natural to assume a similar outcome for feature selection. This line of thinking led to the creation of EFIR, an ensemble feature ranking program. By leveraging existing ML models, EFIR takes in a dataset and outputs a list of features ranked in importance.

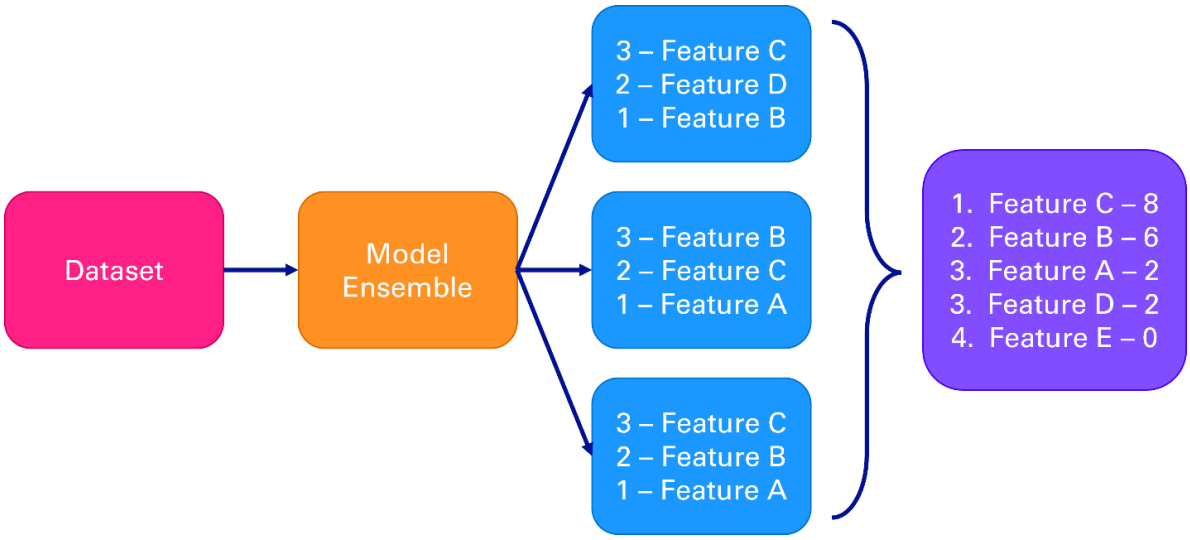

Figure 1: An overview of the EFIR process

The models used in EFIR are Random Forest [39], XGBoost [40], and Linear/Logistic Regression [41], [42]. Both Random Forest and XGBoost are types of decision trees. Random forest is a simpler model robust against overfitting, building its trees independently (parallel) of each other in a bagging technique. XGBoost is more complex and handles unbalanced datasets well, building its trees sequentially through gradient boosting. Linear regression uses ordinary least squares to fit the equation and is used in regression tasks, while logistic regression uses maximum likelihood estimation and is used for classification tasks. Like previously mentioned, they fail when the relationship between features and the target is not linear/logistic.

Each model also runs through a series of feature selection methods, which are permutation importance [43], drop column [44], error analysis [45], and sequential feature selection [46]. See Table 1 for a brief overview of each method.

Table 1: Descriptions of the feature selection methods used in EFIR

Method	Description
Permutation	Calculates the importance of each feature by permuting the values of a single feature, running the altered dataset through the model, and comparing the difference in accuracy (or another metric) between this outcome and a baseline result.
Drop Column	Works similarly to Permutation by dropping an entire column and running this altered dataset, rather than altering the values within a column.
Big/Small Error	Finds the highest and lowest accuracy (or another metric) for observations the model predicted. In other words, find the observation with the best guess and the worst guess. The

	internal feature coefficients used to make the best and worst predictions are pulled for use in EFIR.
Forwards Sequential Selection	Begins by choosing one feature at a time and evaluating performance, going through all *n* features of the dataset. FSS chooses the feature with the best performance and adds it to the dataset. Then it begins on two-feature subsets, adding the next best feature. FSS repeats this cycle with *n*-feature subsets until outcome improvements cease.
Backwards Sequential Selection	Works similarly to FSS, albeit by starting with all *n* features and removing features one at a time.

Internally, this ensemble of models and methods runs through the dataset. Each model in EFIR contains an internal ranking of features which can be pulled out after training. For example, Random Forest contains a method called "feature_importances_", and Linear Regression has "coef_". These methods show which features the model deems important to the outcome.

The heart of EFIR lies in the subsequent ranking system, which "normalizes" these model rankings into one overall score. Because each model maintains a unique internal ranking system, the results from one model cannot be directly compared against a different model. So, EFIR converts each internal ranking into a general score, which is comparable against every other model's internal feature ranking. The result is a list of feature importance scores detailing which features are most highly correlated to the target label.

This paper sets out to prove that EFIR will correctly identify relevant features under a variety of conditions in linear regression datasets.

3 Methods

EFIR takes in a dataset, runs the ensemble on the dataset, and outputs a list of features ranked from most important to least important. To verify this functionality, the tool must be tested on datasets where the most important features are already known. Then, by possessing the answer to each dataset, the efficacy of EFIR can be assessed using custom metrics that compare the results with the correct dataset answers.

3.1 Data

As real-world datasets are complicated and rarely possess certain important features (thus highlighting the need for innovative feature selection tools), they are not useful in evaluating the accuracy of EFIR. Synthetic datasets, however, provide an easy and clear-cut way to assess the tool. By creating a diverse set of data with known important features, different boundaries of EFIR can be explored and measured.

All the synthetic datasets were generated using Scikit-Learn's make_regression method, which generates datasets using a Linear Regression model [47]. The number of features

used, the ratio of important features to unimportant features, and the number of dataset observations were varied to create a range of datasets. Overall, two experiments with two different sets of variables were run and analyzed. See Tables 2 and 3 for a detailed list of the values used in each experiment.

Table 2: Variables and corresponding values used to generate synthetic datasets in Experiment 1

Variable	Values
Features	3, 5, 10, 15, 25, 50, 75, 100, 150
Importance Ratio	0.01, 0.05, 0.1, 0.25, 0.5, 0.75, 0.9, 0.95, 0.99
Observations	10, 100, 1000, 2000, 5000, 7500, 10000

Table 3: Variables and corresponding values used to generate synthetic datasets in Experiment 2

Variable	Values
Features	10, 30, 50, 70, 90, 110, 130
Importance Ratio	0.05, 0.2, 0.35, 0.5, 0.65, 0.8, 0.95
Observations	100, 1600, 3100, 4600, 6100, 7600, 9100

Experiment 1 set out to solve the main thesis statement: EFIR can identify the important features in linear regression datasets. The first set of variables were used for general EFIR testing. Based on the metrics, which are discussed in the following section, the useability of EFIR could be evaluated.

Experiment 2 was created to make results more generalizable and robust towards real-world situations. Which dataset aspects would cause the most influence in the results: the number of features, the ratio of important features to unimportant features, or the number of observations? To answer this question, the second set of variables follow a more rigid, equal step pattern so a Multiple Linear Regression (MLR) model could be run on the results and provide insights into the relationship between the variables and the outcome. MLR is a statistical test that determines if a linear relationship exists between any of the features and the outcome variables. Evaluating the impact of these three variables provides insight into the boundaries of EFIR.

To ensure repeatability, all datasets were set with a seed of 42. Note that the importance ratio rounds up when applied to features to get the total number of important features, since the make_regression method cannot take decimals as parameter values. For example, 3 features multiplied by an importance ratio of 0.01 equals 0.03, which rounds up to 1 important feature. The number 1 is then used in the function to set the total number of important features in the dataset.

Furthermore, some generated datasets were removed due to obvious conflicts. For example, datasets with more features than observations were withdrawn because of common data science practices (see the curse of dimensionality [48]). Datasets with 100%

important features due to rounding were also removed, as the point of EFIR is rendered moot: if all features are important, there is no reason to run an algorithm to assess feature importance.

Finally, since data science happens in the real-world, a dataset with relatively well-known important features will be used to assess EFIR. The dataset chosen for this task is the California Housing dataset [49].

3.2 Metrics

To evaluate EFIR, two different metrics called overlap and distance were created. See Table 4 and Equations 1 and 2 for details about each metric.

Table 4: Metrics used to evaluate EFIR

Metric	Description
Overlap	A percentage between 0-100% (where 100% is a perfect score) measuring how many important features are in the top rankings out of the total number of important features. For example, if there are 3 important features and 10 features total, then those 3 features must show up in the top three spots of the rankings for a perfect 100% overlap.
Distance	A value (where higher numbers equal further distances) measuring the difference between the mean of importance score for each feature cluster (important vs. unimportant). In other words, measure how far apart the mean of important feature scores is from the mean of unimportant feature scores.

Equation 1: Method for calculating the overlap metric

$$overlap = \frac{Imp. Features\ in\ top\ rankings}{Total\ number\ of\ Imp.\ Features}$$

Equation 2: Method for calculating the distance metric

$$distance = \frac{abs(Imp.\ Features\ mean\ -\ Unimp.\ Features\ mean)}{Max\ score - Min\ score}$$

Used in conjunction with each other, overlap and distance provide a picture of EFIR's accuracy. Overlap informs the tool's ability to properly select which features are important, considering the important features are already known. Distance pairs nicely by further detailing how well EFIR separates important features from unimportant features.

4 Results

All scripts, datasets, and plots can be found at the associated GitHub link [50].

Note: EFIR may run with or without Sequential Feature Selection (the slowest method in the ensemble). Due to time constraints, these experiments were run without SFS.

4.1 Experiment 1 – General Analysis

The first test of EFIR compares the results output from the tool with the true important features from each dataset. To analyze the results, facet grids were created with importance ratio as the x-axis, overlap as the y-axis, observation numbers as the lines, and each feature number as a grid plot. In essence, each individual plot shows a slice of the results from the perspective of the feature variable. See Figure 2 for the overlap results of Experiment 1.

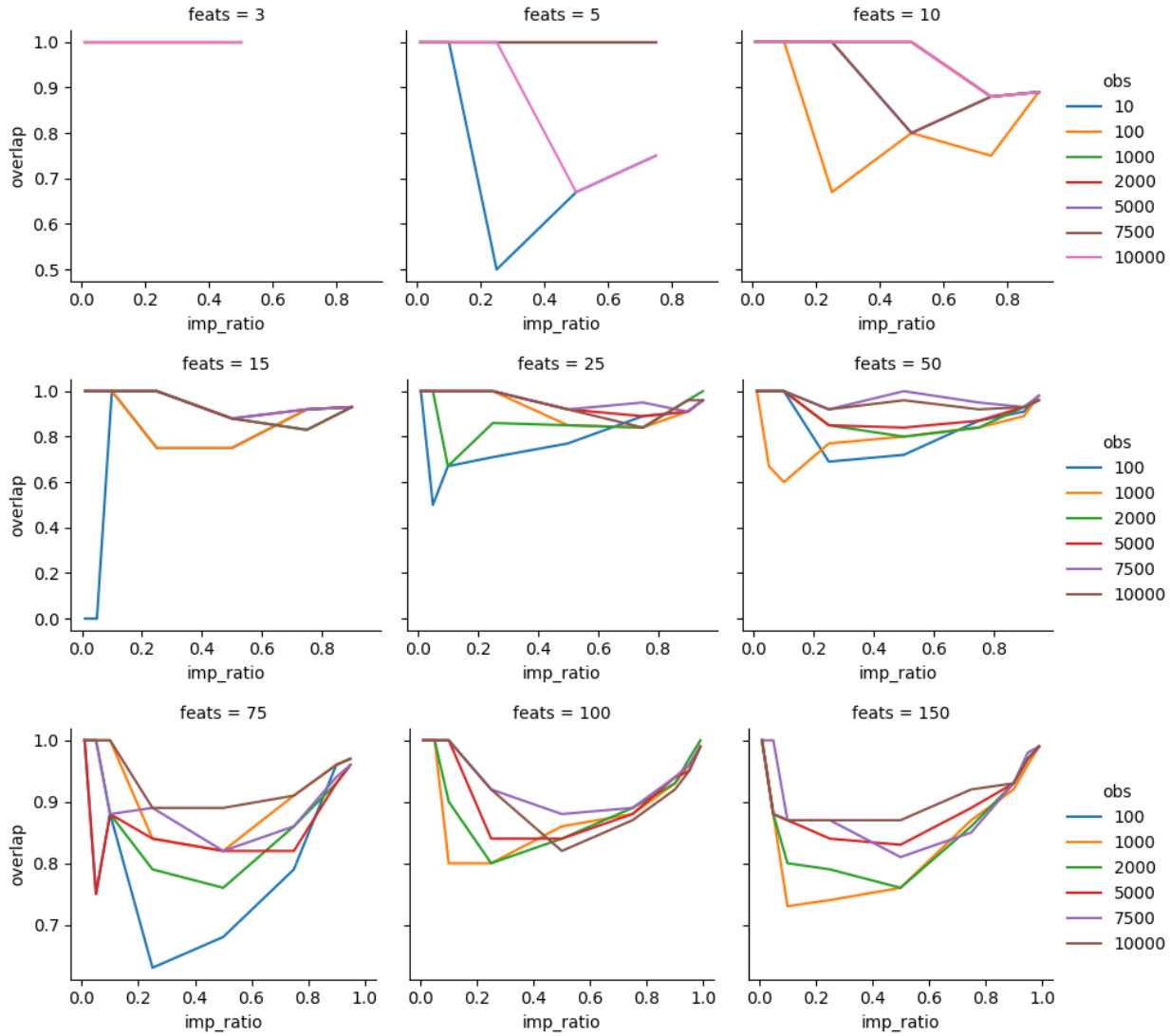

Figure 2: Overlap results from Experiment 1

The overlap results tell an interesting story. Each graph generally follows a parabolic pattern which grows more prominent as the number of features increases. The dip of each parabola occurs when importance ratio is around 50%. However, even in the dips, overlap scores are still around 65-85%. This means EFIR is accurately identifying the impactful features, becoming better as datasets become more comparable to real world sizes.

Intuitively, this makes sense – models need enough data to extrapolate meaningful results (though this varies among models and what problem they are solving [14]). So, the results understandably show more unconventional patterns when there is not enough data to learn trends in these smaller datasets. In this scenario, smaller datasets are those where observations are less than or equal to 1000.

The parabolic patterns occur because more features cause individual feature importance to decrease. Remember that overlap calculates how many important features are ranked at the top compared to all important features. When there are fewer important features, EFIR picks them out with 85-100% accuracy. As more are added, the overall importance of each feature decreases, decreasing EFIR's accuracy. This occurs because each internal feature ranking takes a larger slice of the "importance" cake. There is a fixed amount of importance divided among features, so as the important ratio increases, each feature receives a smaller slice of the overall importance.

This leads to another subtle trend – as the importance ratio increases, each feature becomes less important when viewed through EFIR. The unimportance phenomenon relates to skipping datasets with 100% important features. Viewed another way, imagine a basketball coach looking for the tallest people to play on the team. The coach checks school height records and weeds out everyone under 5'8". Though the height of the tallest players does not change, the average height of the team increases. Applied to these datasets, when all features are important, then each feature becomes less prominent overall, even though they remain good predictors of the target label.

Finally, overlap increases again as importance ratio nears 100% due to simple statistics. When almost all features are important, receiving an almost perfect overlap score becomes easy. For instance, if 9 out of 10 features are important, then at worst 8 important features are in the top 9 spots. This gives a high overlap score of 88.89%. As such, the overlap scores increase as the importance ratio increases.

After considering the overlap results, EFIR can differentiate the importance of features under most conditions tested. However, the tool should also be able to reasonably separate important features from unimportant features. To see these results, facet grids are used to show the distance metric. All axises remain the same as above except for the y-axis, which now shows distance. See Figure 3 for the distance results of Experiment 1.

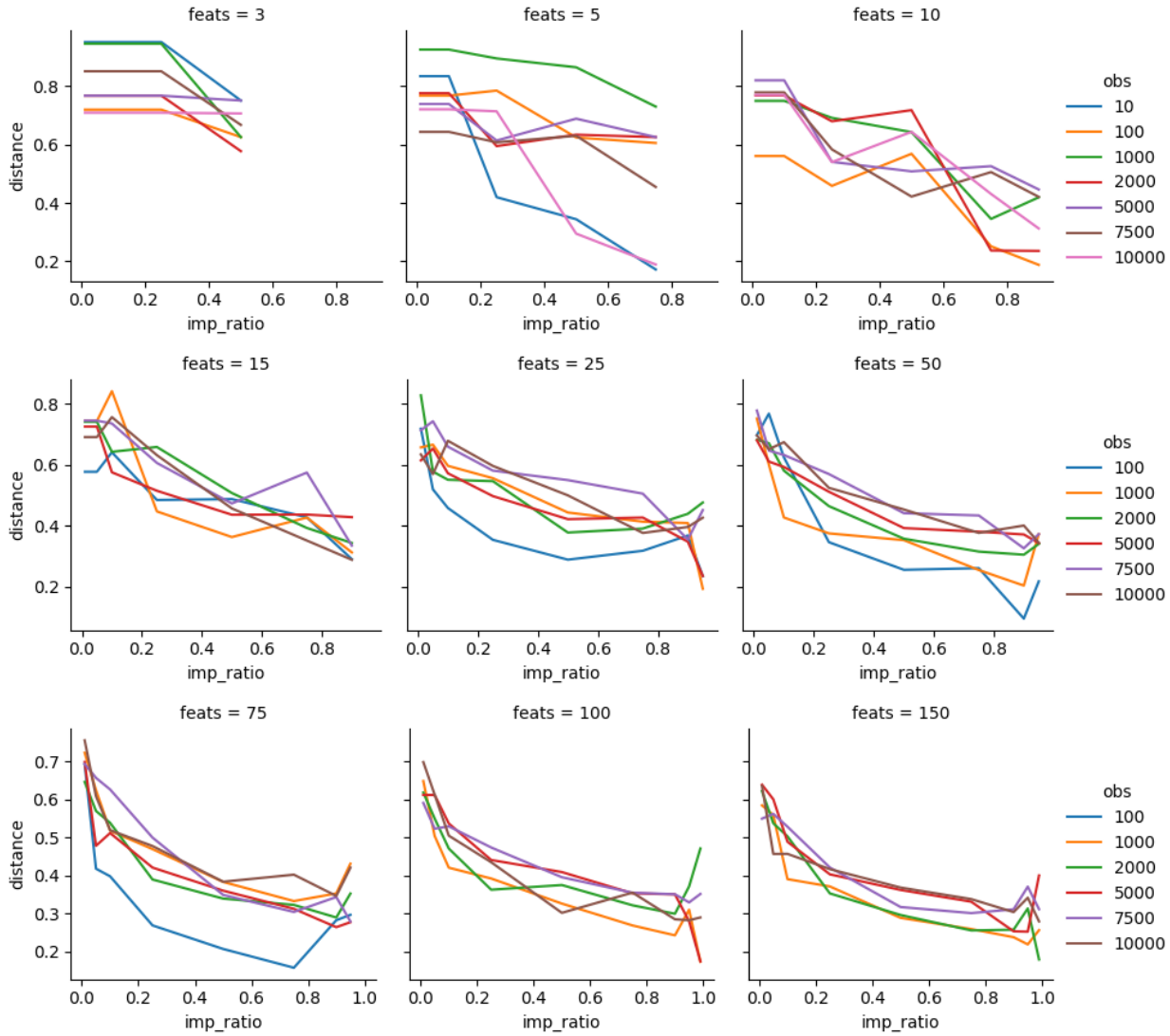

Figure 3: Distance results from Experiment 1

Recall that distance measures the difference in average scores between each feature cluster. Based on the graphs, distance decreases as the importance ratio increases. Referring to the importance cake analogy, the tool is again working as expected. When the number of important features increases, each feature receives a smaller overall importance. As such, the distance in importance scores would subsequently decrease, as each feature becomes closer in importance to every other feature. EFIR accurately exposes this trend through its distance scores. By seeing results such as these, a dataset is likely as lean as it should be when distance becomes small. As distance decreases, EFIR confirms that a dataset contains important predictive features and is ready for model training.

Note that these trends happen regardless of number of features or observations. For example, every feature plot shows a general downward trend. Within each plot, the number of observations does not have influence either – though the overall distance between observations changes, each line still trends downward. Furthermore, exceedingly

small datasets once again tend to show more erratic behavior. For instance, the Feats = 3-10 plots show much larger differences between observations and show some spikes in the downwards trendline. Another strange pattern shows up in the lower observations, such as 10, 100, and 1000. These smaller datasets tend to perform the worst and jump around the most, such as in Feats = 75 plot.

Overall, Experiment 1 confirms EFIR works. Through various synthetic linear regression datasets, the tool accurately predicts which features are important and shows when datasets already possess highly impactful features.

4.2 Experiment 2 – Multiple Linear Regression

The main intent behind EFIR is generalizability. The idea is for the tool to work on any dataset, whether large or small, a classification or regression task, clean or noisy, etc. The graphs from Experiment 1 suggest the number of features and observations had negligible impact on distance and overlap, with importance ratio being the driving force. To understand and quantify the effect each dataset condition had on the target label (either overlap or distance), the results from Experiment 2 were run through a Multiple Linear Regression (MLR) model from Python's statsmodels library [51]. This type of model outputs various statistics, such as feature coefficients and the R^2 value, which are useful for evaluating the variable-outcome relationship.

After accumulating all the results from Experiment 2 and adding a constant, the dataset ran through an MLR model. Table 5 shows the coefficients and P values by running the model with overlap as the target value, and Table 6 is similar but with distance as the target value.

Table 5: Multiple Linear Regression results from Experiment 2 for overlap

Overall Model		Feature	Coefficients	P>|t|
R^2	0.159	Constant	0.9028	0.000
F-Statistic	20.04	Features	-0.0006	0.000
Prob (F-Stat)	6.28e-12	Import. Ratio	-0.0019	0.893
		Observations	0.00000866	0.000

Table 6: Multiple Linear Regression results from Experiment 2 for distance

Overall Model		Feature	Coefficients	P>|t|
R^2	0.721	Constant	0.6215	0.000
F-Statistic	274.3	Features	-0.0014	0.000
Prob (F-Stat)	7.26e-88	Import. Ratio	-0.3075	0.000
		Observations	0.00001101	0.000

The overall regression was not statistically significant for overlap (R^2 = 0.159, F(3, 318) = 20.04, p = 6.28e-12) but was statistically significant for distance (R^2 = 0.721, F(3, 318) = 274.3, p = 7.26e-88). The R^2 from overlap indicates 15.9% of the variability of the target

was accounted for by the three features, while distance accounts for 72.1% of the variability. This shows a much stronger linear correlation between the variables and distance compared to overlap.

These results confirm the results shown in the facet grids. The overlap grids showed a parabolic trendline in the plots. Because the importance ratio had a high p-value of 0.893, it failed to reject the null hypothesis that no linear relationship existed. Furthermore, the small coefficient values associated with each variable, especially compared to the constant, prove no linear correlation exists between the variables and overlap.

However, a strong linear correlation exists between the variables and distance. In particular, the importance ratio has the greatest absolute value among all the other coefficients, proving it has the highest influence on the target variable. For every one value increment, the importance ratio produces a -0.3075 decrease in the target when all else is held constant, which is a much larger magnitude than the number of features (-0.0014) or observations (0.00001101). Furthermore, the p-value for every variable is low, signifying an underlying pattern instead of chance randomness.

This concludes that importance ratio is the largest driving factor to distance in relation to the other variables. In other words, EFIR is robust against dataset size under these linear regression experiments. Regardless of the number of features or observations, the tool will be able to distinguish the most impactful features. Further testing will need to happen to determine the most important aspect for overlap, but based on the facet grids, importance ratio will still possess the most influence.

Overall, Experiment 2 proves that EFIR can evaluate almost any linear regression dataset regardless of the number of features and observations through using an MLR model and the output statistics.

4.3 California Housing Data

Though synthetic data provides a clean way of testing EFIR, datasets come from the real world, where messy, irrelevant, and noisy features are inherent. As such, knowing which features correlate to the target label becomes a much harder problem.

To test EFIR in the wild, the California Housing dataset will be used [49]. This is a regression task dataset using housing related features to solve Median House Value for real estate in California. See Table 7 for an overview of the dataset.

Table 7: An overview of the California Housing dataset

California Housing Dataset	
Observations	20640
Missing Values	None
Target Label	Median House Value

Features	Median Income, House Age, Average Rooms, Average Bedrooms, Population, Average Occupants, Latitude, Longitude

Two sources detailing different feature selection methods were chosen to illustrate how various methods lead to different results and how EFIR holds up against them. The first source is an independent data scientist named Ivan Pupkin using this dataset [52]. The second is from the scikit-learn documentation, the Python package used throughout this paper [53]. See Table 8 for the comparison of results.

Table 8: Comparison of important features between various sources

Source	Important Features
Ivan Pupkin	1. Median Income 2. Latitude 3. Average Rooms 4. House Age
Scikit-Learn	• Longitude • Latitude • Median Income
EFIR	1. Median Income – 66 2. House Age – 43 3. Longitude – 40 4. Latitude – 37 5. Average Rooms – 34 6. Average Occupancy – 32 7. Average Rooms – 32 8. Population – 25

Pupkin is shown with numbers since he quantifies the feature importance, while Scikit-Learn receives bullet points as order of importance is never clarified. EFIR shows the entire result output from running the dataset. As shown in the table, EFIR captures all the features the two sources conclude as important. Median Income, Latitude, Longitude, Average Rooms, and House Age show up as the top five important features based on EFIR.

This information can be used to clean out a dataset from the top-down or bottom-up. For instance, a data scientist may remove the bottom 30% of the features shown in the EFIR rankings. Then they can run this leaner dataset through a model and compare results to the original dataset, removing more features if accuracy (or another metric) keeps improving. Alternatively, they may start with the top 30% of features and add features if accuracy (or another metric) keeps improving. As Median Income maintains the highest ranking by 23 points compared to the differences between the other feature rankings, a top-down approach would likely be most beneficial for this dataset. Either way, the dataset becomes leaner, and the data scientist possesses better dataset insight.

EFIR utilizes the ensemble method to ensure many different boundaries are considered when ranking features. As such, the tool does not impose the same limitations as other feature selection methods. Because of its robust nature, EFIR more accurately reports the important features in a dataset, as shown through the California Housing example. This provides data scientists with better intuition when determining which features to remove from a dataset, leading to improved datasets and faster model training times.

5 Conclusion

Through this research, EFIR proved effective in determining the important features of linear regression datasets, showing robustness towards obtaining lean data and running faster models. On these types of datasets, the tool works correctly, provides a broader view of feature importance compared to other feature selection methods, and shows value in reducing dataset complexity and model training times.

Though successful at the datasets in the experiments, the tool is not a perfect solution towards feature selection: running results takes time, the internal models used are not always accurate, and further investigation of the tool is recommended. There are multiple avenues for further testing and validating the efficacy of EFIR, such as

- Finding a better model for overlap to statistically prove how the features, observations, and importance ratio affect overlap scores
- Using other synthetic dataset generators to create and assess non-linear datasets
- Starting with a clean dataset and adding dirty features at various levels to assess noise robustness
- Analyzing the computational savings of reduced datasets on model training
- Critical review of the efficacy of the evaluation metrics
- Testing other real-world datasets

Data science is a growing industry quickly running into bigger and more problematic limitations. Two major problems faced by data scientists involve datasets with too many features and long model training times. There are a multitude of algorithms created to solve the first problem and many companies actively work on new ways of creating faster hardware to solve the second, yet both problems persist today.

EFIR was created to combat those challenges. This paper explained how the tool solves these problems and proves its efficacy in finding the important features of regression datasets. Through this process, a data scientist can identify key features in a dataset and remove the rest, both decreasing the complexity of the dataset and improving model training times. More importantly, EFIR is not limited by the size or task of the dataset – the tool is robust and generalizable to most regression data. These results provide a glimmer of hope towards improving the data science pipeline in the future.

6 References

[1] M. Connall, "Top 20 Big Data Statistics for 2022," *Sigma Computing*, Jun. 24, 2020. https://www.sigmacomputing.com/blog/top-20-big-data-statistics/ (accessed Feb. 07, 2022).

[2] "What is the Internet of Things (IoT)?," *Oracle*. https://www.oracle.com/internet-of-things/what-is-iot/ (accessed Feb. 07, 2022).

[3] R. Midrack, "How Smart Is Your Dishwasher?," *Lifewire*. https://www.lifewire.com/smart-dishwasher-4159822 (accessed Feb. 07, 2022).

[4] S. Brown, "Machine learning, explained," *MIT Sloan*. https://mitsloan.mit.edu/ideas-made-to-matter/machine-learning-explained (accessed Feb. 07, 2022).

[5] J. Phoenix, "Top Data Science Problems and How to Avoid Them," *Just Understanding Data*, Feb. 02, 2021. https://understandingdata.com/top-data-science-problems-and-how-to-avoid-them/ (accessed Feb. 07, 2022).

[6] J. Brownlee, "Why Training a Neural Network Is Hard," *Machine Learning Mastery*, Feb. 28, 2019. https://machinelearningmastery.com/why-training-a-neural-network-is-hard/ (accessed Feb. 07, 2022).

[7] R. S. Baker and A. Hawn, "Algorithmic Bias in Education," *Int. J. Artif. Intell. Educ.*, Nov. 2021, doi: 10.1007/s40593-021-00285-9.

[8] J. Condliffe, "The Week in Tech: Algorithmic Bias Is Bad. Uncovering It Is Good.," *The New York Times*, Nov. 15, 2019. Accessed: Feb. 07, 2022. [Online]. Available: https://www.nytimes.com/2019/11/15/technology/algorithmic-ai-bias.html

[9] H. Ledford, "Millions of black people affected by racial bias in health-care algorithms," *Nature*, vol. 574, no. 7780, pp. 608–609, Oct. 2019, doi: 10.1038/d41586-019-03228-6.

[10] M. Stewart, "The Limitations of Machine Learning," *Medium*, Jul. 29, 2020. https://towardsdatascience.com/the-limitations-of-machine-learning-a00e0c3040c6 (accessed Feb. 07, 2022).

[11] M. Verleysen and D. François, "The Curse of Dimensionality in Data Mining and Time Series Prediction," in *Computational Intelligence and Bioinspired Systems*, Berlin, Heidelberg, 2005, pp. 758–770. doi: 10.1007/11494669_93.

[12] D. Haughton, M. A. Robbert, L. P. Senne, and V. Gada, "Effect of Dirty Data on Analysis Results".

[13] N. Thompson, "Deep Learning's Diminishing Returns," *IEEE Spectrum*, Sep. 24, 2021. https://spectrum.ieee.org/deep-learning-computational-cost (accessed Feb. 07, 2022).

[14] A. L'Heureux, K. Grolinger, H. F. Elyamany, and M. A. M. Capretz, "Machine Learning With Big Data: Challenges and Approaches," *IEEE Access*, vol. 5, pp. 7776–7797, 2017, doi: 10.1109/ACCESS.2017.2696365.

[15] J. Tang, S. Alelyani, and H. Liu, "Feature Selection for Classification: A Review".

[16] G. H. John, R. Kohavi, and K. Pfleger, "Irrelevant Features and the Subset Selection Problem," in *Machine Learning Proceedings 1994*, W. W. Cohen and H. Hirsh, Eds. San Francisco (CA): Morgan Kaufmann, 1994. doi: 10.1016/B978-1-55860-335-6.50023-4.

[17] I. Guyon and A. Elisseeff, "An Introduction to Variable and Feature Selection".

[18] G. Chandrashekar and F. Sahin, "A survey on feature selection methods," *Comput. Electr. Eng.*, vol. 40, no. 1, Jan. 2014, doi: 10.1016/j.compeleceng.2013.11.024.
[19] R. Muthukrishnan and R. Rohini, "LASSO: A feature selection technique in predictive modeling for machine learning," presented at the 2016 IEEE International Conference on Advances in Computer Applications (ICACA), Oct. 2016. doi: 10.1109/ICACA.2016.7887916.
[20] C. Lee and G. G. Lee, "Information gain and divergence-based feature selection for machine learning-based text categorization," *Inf. Process. Manag.*, vol. 42, no. 1, Jan. 2006, doi: 10.1016/j.ipm.2004.08.006.
[21] R. J. Urbanowicz, M. Meeker, W. La Cava, R. S. Olson, and J. H. Moore, "Relief-based feature selection: Introduction and review," *J. Biomed. Inform.*, vol. 85, Sep. 2018, doi: 10.1016/j.jbi.2018.07.014.
[22] Q. Gu, Z. Li, and J. Han, "Generalized Fisher Score for Feature Selection," *ArXiv12023725 Cs Stat*, Feb. 2012, Accessed: Dec. 14, 2021. [Online]. Available: http://arxiv.org/abs/1202.3725
[23] Q. Li and N. Lin, "The Bayesian elastic net," *Bayesian Anal.*, vol. 5, no. 1, Mar. 2010, doi: 10.1214/10-BA506.
[24] R. Rodríguez-Pérez and J. Bajorath, "Feature importance correlation from machine learning indicates functional relationships between proteins and similar compound binding characteristics," *Sci. Rep.*, vol. 11, no. 1, Jul. 2021, doi: 10.1038/s41598-021-93771-y.
[25] J. Hua, W. D. Tembe, and E. R. Dougherty, "Performance of feature-selection methods in the classification of high-dimension data," *Pattern Recognit.*, vol. 42, no. 3, Mar. 2009, doi: 10.1016/j.patcog.2008.08.001.
[26] B. H. Menze *et al.*, "A comparison of random forest and its Gini importance with standard chemometric methods for the feature selection and classification of spectral data," *BMC Bioinformatics*, vol. 10, no. 1, Jul. 2009, doi: 10.1186/1471-2105-10-213.
[27] D. Oreski, S. Oreski, and B. Klicek, "Effects of dataset characteristics on the performance of feature selection techniques," *Appl. Soft Comput.*, vol. 52, Mar. 2017, doi: 10.1016/j.asoc.2016.12.023.
[28] A. P. White and W. Z. Liu, "Technical Note: Bias in Information-Based Measures in Decision Tree Induction," *Mach. Learn.*, vol. 15, no. 3, pp. 321–329, Jun. 1994, doi: 10.1023/A:1022694010754.
[29] E. Lewinson, "Explaining Feature Importance by example of a Random Forest," *Medium*, Aug. 26, 2021. https://towardsdatascience.com/explaining-feature-importance-by-example-of-a-random-forest-d9166011959e (accessed Dec. 09, 2021).
[30] Y. C. Ho and D. L. Pepyne, "Simple Explanation of the No-Free-Lunch Theorem and Its Implications," *J. Optim. Theory Appl.*, vol. 115, no. 3, Dec. 2002, doi: 10.1023/A:1021251113462.
[31] M. Re and G. Valentini, *Ensemble Methods*. 2012, pp. 563–593. Accessed: Dec. 14, 2021. [Online]. Available: https://ui.adsabs.harvard.edu/abs/2012amld.book..563R
[32] O. Sagi and L. Rokach, "Ensemble learning: A survey," *WIREs Data Min. Knowl. Discov.*, vol. 8, no. 4, p. e1249, 2018, doi: 10.1002/widm.1249.
[33] S. Wang and X. Yao, "Diversity analysis on imbalanced data sets by using ensemble models," in *2009 IEEE Symposium on Computational Intelligence and Data Mining*, Mar. 2009, pp. 324–331. doi: 10.1109/CIDM.2009.4938667.

[34] J. D. Wichard and M. Ogorzalek, "Time series prediction with ensemble models," in *2004 IEEE International Joint Conference on Neural Networks (IEEE Cat. No.04CH37541)*, Jul. 2004, vol. 2, pp. 1625–1630 vol.2. doi: 10.1109/IJCNN.2004.1380203.
[35] L. Zhou, K. K. Lai, and L. Yu, "Least squares support vector machines ensemble models for credit scoring," *Expert Syst. Appl.*, vol. 37, no. 1, pp. 127–133, Jan. 2010, doi: 10.1016/j.eswa.2009.05.024.
[36] G. König, C. Molnar, B. Bischl, and M. Grosse-Wentrup, "Relative Feature Importance," *ArXiv200708283 Cs Stat*, vol. 12667, 2021, doi: 10.1007/978-3-030-68787-8.
[37] A. Zien, N. Krämer, S. Sonnenburg, and G. Rätsch, "The Feature Importance Ranking Measure," in *Machine Learning and Knowledge Discovery in Databases*, vol. 5782, W. Buntine, M. Grobelnik, D. Mladenić, and J. Shawe-Taylor, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, pp. 694–709. doi: 10.1007/978-3-642-04174-7_45.
[38] G. Casalicchio, C. Molnar, and B. Bischl, "Visualizing the Feature Importance for Black Box Models," in *Machine Learning and Knowledge Discovery in Databases*, Cham, 2019. doi: 10.1007/978-3-030-10925-7_40.
[39] L. Breiman, "Random Forests," *Mach. Learn.*, vol. 45, no. 1, Oct. 2001, doi: 10.1023/A:1010933404324.
[40] T. Chen and C. Guestrin, "XGBoost: A Scalable Tree Boosting System," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, New York, NY, USA, Aug. 2016, pp. 785–794. doi: 10.1145/2939672.2939785.
[41] D. C. Montgomery, E. A. Peck, and G. G. Vining, *Introduction to Linear Regression Analysis*. John Wiley & Sons, 2021.
[42] R. E. Wright, "Logistic regression," in *Reading and understanding multivariate statistics*, Washington, DC, US: American Psychological Association, 1995.
[43] A. Altmann, L. Toloşi, O. Sander, and T. Lengauer, "Permutation importance: a corrected feature importance measure," *Bioinformatics*, vol. 26, no. 10, May 2010, doi: 10.1093/bioinformatics/btq134.
[44] T. Parr, K. Turgutlu, C. Csiszar, and J. Howard, "Beware Default Random Forest Importances." http://explained.ai/decision-tree-viz/index.html (accessed Dec. 09, 2021).
[45] P. Sharma *et al.*, "Evaluating Tree Explanation Methods for Anomaly Reasoning: A Case Study of SHAP TreeExplainer and TreeInterpreter," in *Advances in Conceptual Modeling*, Cham, 2020. doi: 10.1007/978-3-030-65847-2_4.
[46] D. W. Aha and R. L. Bankert, "A Comparative Evaluation of Sequential Feature Selection Algorithms," in *Learning from Data: Artificial Intelligence and Statistics V*, D. Fisher and H.-J. Lenz, Eds. New York, NY: Springer, 1996, pp. 199–206. doi: 10.1007/978-1-4612-2404-4_19.
[47] "sklearn.datasets.make_regression," *scikit-learn*. https://scikit-learn/stable/modules/generated/sklearn.datasets.make_regression.html (accessed Feb. 12, 2022).
[48] D. L. Donoho, "High-Dimensional Data Analysis: The Curses and Blessings of Dimensionality".
[49] "California Housing Dataset," *DCC*. https://www.dcc.fc.up.pt/~ltorgo/Regression/cal_housing.html (accessed Mar. 06, 2022).

[50] N. Rubocki, *Ensemble Feature Importance Ranker (EFIR) Repository*. [Online]. Available: https://github.com/NikitaRubocki/thesis
[51] "Multiple Linear Regression — Basic Analytics in Python." https://www.sfu.ca/~mjbrydon/tutorials/BAinPy/10_multiple_regression.html (accessed Mar. 06, 2022).
[52] I. Pupkin, "California housing I - Feature selection and data exploration | Ivan Pupkin's equivariance." https://pupkinivan.github.io/2019/08/19/ca-housing-01-feature-selection.html (accessed Mar. 06, 2022).
[53] "The California housing dataset — Scikit-learn course." https://inria.github.io/scikit-learn-mooc/python_scripts/datasets_california_housing.html (accessed Mar. 06, 2022).