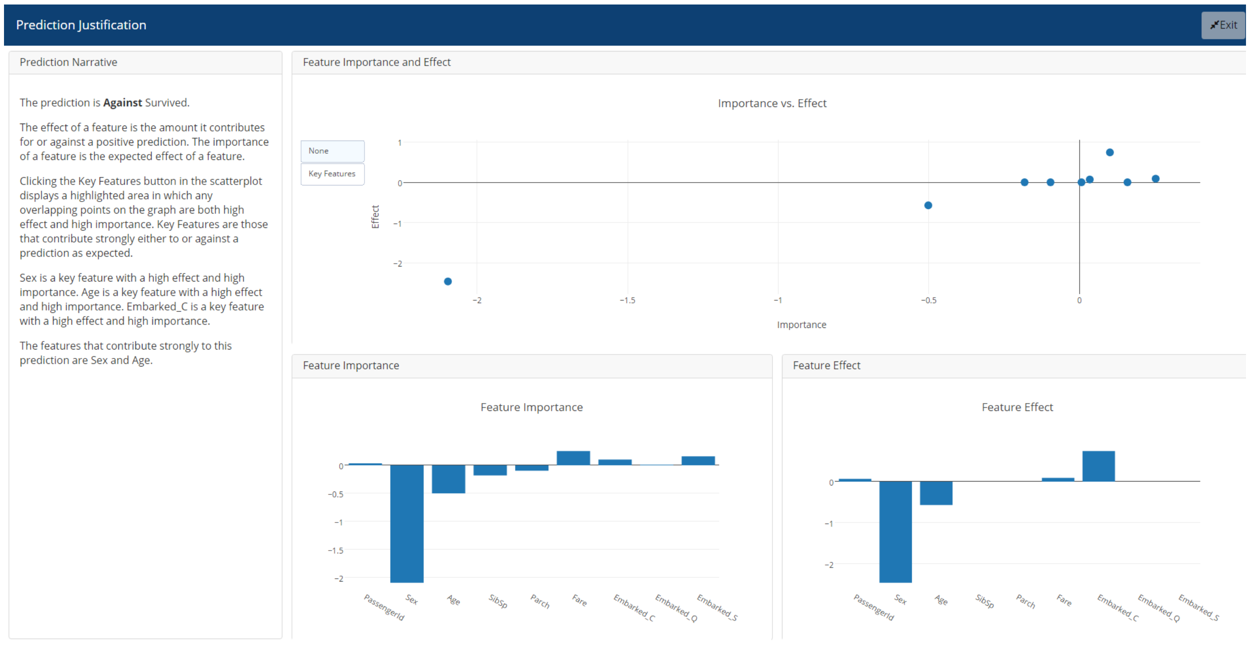
Visualizations for Justifying Machine Learning Predictions

David Johnson



The prediction is **Against** Survived.

The effect of a feature is the amount it contributes for or against a positive prediction. The importance of a feature is the expected effect of a feature.

Clicking the Key Features button in the scatterplot displays a highlighted area in which any overlapping points on the graph are both high effect and high importance. Key Features are those that contribute strongly either to or against a prediction as expected.

Sex is a key feature with a high effect and high importance. Age is a key feature with a high effect and high importance. Embarked_C is a key feature with a high effect and high importance.

The features that contribute strongly to this prediction are Sex and Age.

Fig. 1: A system for justifying machine learning predictions.

Abstract—Machine learning has seen its uses continually grow into fields far removed from its computer science and statistical roots. Users with no training in machine learning and now confronted with the task of deciding whether to accept or reject a prediction from a machine learning algorithm that they do not have the training to understand. This system addresses this issue by providing visualization idioms that justify to the user why a prediction was made given important or outlying features.

Index Terms—Information Visualization, Machine Learning

1 Introduction

As research has grown in the field of machine learning (ML) more strengths are discovered, and potential uses unearthed. This progress has seen use of ML expand into many new domains, confronting users with no training in ML with the task of understanding the reasons for why an ML algorithm has made its prediction. To someone with no domain knowledge, an ML algorithm is a black box.

Previous work by Biran et al. [2] indicates that in many fields being able to understand the justification for why a prediction was made by an ML system is the most important reason in whether to adopt an ML system or not. This is particularly important in fields like health care, in which there are very real and important consequences to accepting an ML prediction as accurate. Given that its not a trivial task for a user to learn the inner workings of some ML algorithm without previous training, a system able to justify the reasons for an ML prediction to a user without requiring the user to understand the underlying model could be of great use. This system is intended to generate such a justification.

In order to create these justifications, this system visualizes both feature effect and feature importance. Feature effect and importance measure a features contribution to a prediction and a features expected contribution to a prediction respectively. These measures are chosen because they are more interpretable than a raw fit model or raw features; effect and importance let a user see clearly which features are contributing negatively or positively as well as which are contributing the most to the prediction. Effect and importance also allow a user to easily spot outliers for instance, a feature that is high importance (high expected effect) but low effect might be notable for some reason.

Through utilizing scatter and bar chart visualizations of effect and importance the system presents a clear and simple way for users to understand the justification of an outcome from an ML predictor. To further increase system efficacy, I implemented an interaction that allows users to also see features identified as key features - those that are high effect and high importance.

Additionally, a text view is added which acts as a simulated result of a natural language generation implementation. Although the natural language generation implementation was outside the scope of this project, the textual justification contributes notably to the overall usefulness of the tool for the target user, and was therefore important to simulate. The text presents a narrative which identifies the outcome of the prediction on the test set instance, an explanation of how to use the interactive effects of the chart, and identifies key features.

By combining the visualizations and the textual justifications the system has a simple and concise way of providing justifications. The system is viewable at https://plot.ly/dashboard/davewj:9/view

• *David is with the University of British Columbia. Email: davewj@cs.ubc.ca*

2 Related Work

Biran et al [2] have written about difficulties involved in generating justifications – difficulties particularly relevant to my system. They suggest generating justifications is a task composed of two problems: selecting relevant features, and discussing both the state of the model and the real world implications of the features. These issues will be the main two issues I expect to encounter during development. Selecting relevant features is an implementation issue, one that will require experimentation during development to determine the best approach for selecting features. The efficacy of showing the user the state of the model and implications of features will depend on the visualizations I decide to use.

Krause et al [4] developed a system called Prospector which allows for interaction with their system in such a way as to allow users to see how a model performs after changing feature values. By allowing this interaction, Prospector improves model interpretability and user understanding of features and their impact on a prediction. Although Krause et al indicate that Prospector is primarily intended to be used by data scientists, I believe similarly styled interaction could be used to also improve model interpretability for non-domain experts. The interaction in Prospector is too complex for me to implement given the time constraints, but a more simplistic implementation of model interaction should be realistic and effective.

One notable decision for my system is that it focuses exclusively on explaining logistic regression. Although this is primarily due to the heavy time constraint, there is also reason to believe logistic regression is a good choice due to its interpretability. Lipton [5] discussed model interpretability, suggesting that interpretable models are transparent and have strong post-hoc interpretability. Logistic regression fits these requirements well, suggesting it is a good choice for this type of system.

In contrast to the idea of focusing on one interpretable model, Ribeiro et al [6] argue that explanations of predictions should be model agnostic, i.e. that explanations should be separate from the models themselves. Rather than focusing on beginning with interpretable models, they treat the original models as black boxes and instead learn interpretable models on the predictions of the black boxes. Using this approach, they developed LIME (Local Interpretable Model-agnostic Explanations) [7] which generates both text and visualization explanations of their learned models. Although this is quite an interesting approach, its substantially more challenging and unrealistic for my project given the time constraints.

Among previous works in developing actual systems, the most relevant to my proposed project is PreJu (Prediction Justifier) by Biran et al. PreJu [1] determines the effect of each feature in a logistic regression classifier and the importance (expected effect) of a feature to determine key features. Key features are then used to generate a justification narrative with both text and visualization in an attempt to justify to the user the importance of features to the model.

Although my current system focuses on justifying predictions from a single learned logistic regression model, a possible future addition to my system could be allowing comparisons of learned models. Kahng et al [3] developed a system known as MLCube Explorer that allows users to visually compare different learned machine learning models for a particular dataset on measures such as accuracy and score distribution. MLCube Explorer also shows that some learned model performs better on an instance of a feature than some other learned model (i.e. model A outperforms model B when A only when it has certain feature values). The intention is to show that strictly comparing the accuracies of two models is not necessarily effective enough, instead users could compare model performance on subsets of features. A similar addition to my system could possibly allow justification beyond the current concept of looking at the state of one fit model. The finely grained comparison allowing visualization of performance of multiple models against each other on feature subsets could be considered a means of justifying predictions to the user – particularly the idea of justifying why one fit model outperforms the other (and on what subset of features).

3 Solution

Ensuring to keep the user in mind, I chose to stick to simplistic visualizations that the user would likely be familiar with even without training in ML or a related statistics field. A faceted view was used with scatterplots and bar charts to visualize effect and importance. The scatterplot was chosen to show effect against importance as a means of allowing the user to visually see features that are key features (high effect high importance) as well as those that are outliers. The bar plots were used as a means of allowing direct comparison between feature effect and importance and to cover one weakness of the scatterplot.

The most prominent and expressive visualization used in my system is the scatterplot of effect and importance. Effect and importance are two quantitative values for which we want a clear overview and the ability to find outliers, so the scatterplot is a very effective idiom. Scatter plots use of both vertical and horizontal spatial position channels make them intuitive to understand for users without much formal statistical training. Theyre also quite likely to be familiar even for users without any formal statistical training simply due to their wide usage in many scenarios.

The scatterplot also allows for interaction. While hovering over each point on the scatterplot a window appears showing the value and feature name. Additionally, the plot has a Key Features button which when clicked brings up a highlighted area on the plot. The highlight is representative of key features, so that any points which overlap with this highlight are identified as key features. These key features are features which are both high effect and high importance.

Two separate bar charts, one for effect and one for importance, are also used. The decision to implement two bar charts was chosen to add an even simpler means of finding effect and importance. Bar charts are very effective at allowing looking up of individual values. The ability to quickly find effect and importance of a single feature ties in with the textual justification, users that read that a feature is a key feature in the text narrative can then quickly find the actual values of effect or importance by glancing at the bar charts. This is easier for the user than having to look at the scatterplot and mouse over points to see the feature name.

The system implements some faceting for the purpose of showing the multiple views of the scatter and bar charts together. As mentioned, a particularly helpful aspect of this faceting is that the weakness of the scatterplots in not immediately showing feature names on points is covered by the bar charts which have feature names on their axis particularly helpful when the textual narrative exists in which features are referred to by name. In addition to this, faceting in such a way as to juxtapose bar charts of effect and importance allows users to compare both charts back and forth quickly so that differences in effect and importance can be compared between features.

Although I used a scatterplot and bar charts, they were not the only design idioms considered. The first other design considered was a parallel coordinates idiom. The parallel coordinates idiom wouldve been effective for allowing users to quite easily see all the features at once, as well as seeing the range of the attributes. This wouldve worked well at allowing users to quickly see which features are high effect or high importance. The problems with this decision are that the parallel coordinates couldve only shown one of effect or importance at a time, so there would have to be two parallel coordinates views juxtaposed. Additionally, there can be issues with occlusion on parallel coordinates plots in instances in which the feature set is quite large. After considering the parallel coordinates plot it did seem that scatterplot was more effective at displaying data.

I also considered using a pie chart as a means of showing effect and importance. The strength here wouldve been to allow the user to pick out key features quite intuitively by noticing one feature taking up a large amount of the effect or importance pie. The strength was not enough to make up for the multiple weaknesses present in pie charts, however. Firstly, the pie chart wouldve required splitting the chart into two, one for effect and importance. Second, pie charts have issues with angle judgments. It can be quite hard to compare two similarly sized slices of pie and determine which one is greater. It was decided that pie charts were not going to be nearly as effective at displaying data of this type as a scatterplot.

Lastly its also worth noting that I did consider adding the actual logistic regression plot as well. The thought was it might be interesting to the user, even those not familiar with logistic regression, to see the generated s-curve shape of the logistic plot. I decided against it however, thinking that users might try to find the connection between the plot and effect and importance which are derived values and dont necessarily have an impact on the s-curve plot. It was thought that the small bonus of possibly being interesting was not worth the confusion and particularly the screen real estate that it would require to add the regression plot.

After considering multiple possible designs, the scatterplot with supplemental bar charts seemed clearly the optimum choice. The one weakness that the scatterplot might have is that the user doesnt immediately see the names of each feature, instead they must mouse over each point looking at feature names. This weakness is offset using the bar charts however, which include the feature names as part of their axis. These design choices together are concise and easy to understand, while clearly displaying positive and negative effect and importance while additionally making it easy for the user to spot outliers and unusual events.

4 Implementation

The system was written in Python. Scikit-learn was also used to run the logistic regression machine learning. Pandas was used for the data structures. Plotly was used to draw the plots, as well as combine them into a dashboard which allows fitting multiple plots into a single view. Numpy was used for interacting with data structures and at times doing matrix calculations.

A substantial amount of development time was devoted to calculating effect and importance, as well as calculating key features. Biran et al. [1] propose a method for calculating these 3 values. Effect is a measure of the contribution of a feature in the instance being predicted.

$$\text{Effect}_{ji} = \theta_{ji} x_i \tag{1}$$

Essentially, we iterate through our test instance and find all the feature values. Next, multiply the feature values by the matching model coefficient.

Next, we calculate importance. Importance is a measure of the expected contribution of a feature.

$$\text{Importance}_{ji} = \theta_{ji} \frac{\sum_{x \in X^j} x_i}{|X^j|} \tag{2}$$

To find this value I find all instance in the training set that have the same class as the predicted output class from the model. Then I find the mean feature value for all these instances with the correct matching class. Lastly, multiply the mean feature values by the matching model coefficients.

Key features are those that are high effect and high importance. This is measured by first quantizing features into those that are high and low importance. High importance is defined as being the smallest subset of features that contribute r of the total importance of all features combined where r is a tunable parameter. I chose to use r = 75% as this is also number recommended by Biran. Low importance is the set of remaining features which do not contribute r of the total importance.

After quantizing features into high and low importance we quantize features into high and low effect. To find features which are high effect I find the point between the lowest high importance feature and the highest low importance feature. In other words, I aim to find a margin that maximizes the distance between high importance and low importance features. Once I find the margin, I classify negatively contributing features as high effect if they are further negative than the margin. Similarly, I classify positively contributing features as high effect if they are further positive than the margin. After this process I now have a set of high and low importance features (both negatively and positively) , a set of high and low effect features (again negatively and positively), and a set of key features (also negatively and positively). I am now able to plot effect and importance in a scatterplot as well as individually in bar charts. I'm also able to implement the highlighting aspect of the design that shows key features.

Since the scale of this project was not intended to be a full web application implementation, I also had to manually do some feature engineering for the simulated user. This involved changing some features from names to categorical numerical values (0 or 1). Certain features also had to be dropped in cases where they would not contribute to a prediction. I also had to manually fit the logistic regression model which again is a simulation of what the user would theoretically do.

5 Data and Task Abstraction

5.1 Domain-specific data

The data used here is an arbitrary dataset chosen by a user. Since a user can choose an arbitrary data set to visualize, understanding the semantics of the data is a task that is up to the user herself. The dataset will be a dataset fit for machine learning i.e. the dataset will have a training set and test set. In addition to the training and test sets, there will be a logistic regression model that is fit to the training set. Effect and importance values will be calculated from this fit model, training, and test set instances.

5.2 Domain-specific task

The task is to have the system first make a single prediction on one instance in the test set. A justification should be created based on effect and importance which allow a user to understand why a prediction was made, and allow the user to utilize the justification to either agree and accept the prediction or disagree with it.

5.3 Abstract data

In abstracting the data it seems it is composed of items (training and test set instances, attributes (features) and values (effect and importance) in a table. The attribute are all quantitative.

5.4 Abstract task

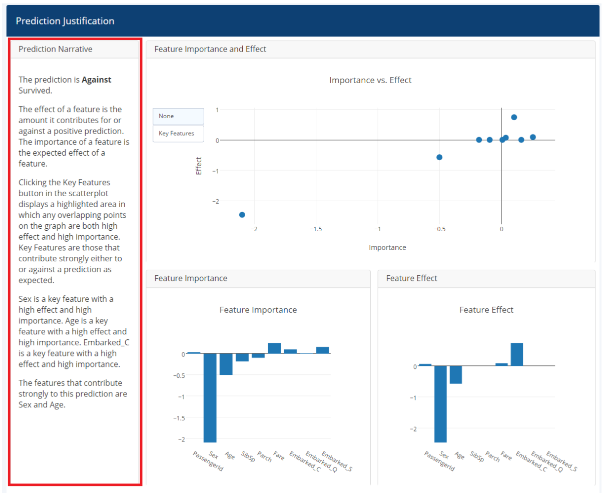
The general overview here is that I have quantitative attributes (feature values) and want to generate a scatterplot and two bar charts. This is done for the purpose of analysis, discovery, exploration and enjoyment. I want to accomplish this by arranging tabular data. The system has a producing task since its required to derive new data from original raw data - notably, effect and importance must be derived from the original feature values. The system additionally derives an attribute for key feature classification. I also want to add manipulation by allowing highlighting as a means of demonstrating on the scatterplot where key features are located. The system also needs to facet features into multiple views, the scatterplot and juxtaposed bar charts.

6 Results

Prior to Figure 1 the user has already obtained their own data set and has fit a logistic regression model. The user has brought the fit model, the training set used to train the model, and the test set and loaded them into the model (in the current implementation this must be done in the actual code but in future implementations could be accomplished through a UI in a full web application deployment).

The user first starts by inspecting the justification narrative which includes text that states the actual prediction outcome in Fig. 2. The user proceeds to read about which features are identified as key features - features that contribute strongly either negatively or positively to the prediction.

After reviewing the text the user focuses on the scatterplot of importance and effect in Fig. 3. Using the visualized scatterplot, the user can gain a better sense of how strongly certain features contribute to the predictions beyond the simpler textual explanation they looked at first. The user interacts with the chart by mousing over the points on the graph to discover that its the Sex feature that seems to be contributing very negatively compared to the other features in the dataset in Fig. 4. This is a strong indication that the justification for this prediction is due to the Sex feature. The user presses the Key Features button which brings up a highlight upon which any overlapping points are



The prediction is **Against** Survived.

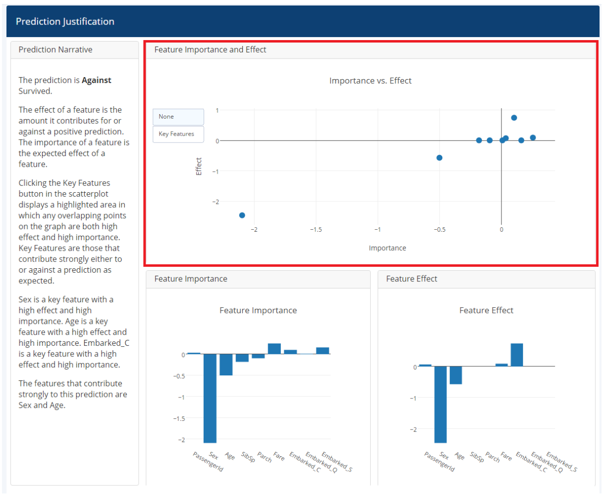
The effect of a feature is the amount it contributes for or against a positive prediction. The importance of a feature is the expected effect of a feature.

Clicking the Key Features button in the scatterplot displays a highlighted area in which any overlapping points on the graph are both high effect and high importance. Key Features are those that contribute strongly either to or against a prediction as expected.

Sex is a key feature with a high effect and high importance. Age is a key feature with a high effect and high importance. Embarked_C is a key feature with a high effect and high importance.

The features that contribute strongly to this prediction are Sex and Age.

Fig. 2: User inspects narrative (highlighted in red).


The prediction is **Against** Survived.

The effect of a feature is the amount it contributes for or against a positive prediction. The importance of a feature is the expected effect of a feature.

Clicking the Key Features button in the scatterplot displays a highlighted area in which any overlapping points on the graph are both high effect and high importance. Key Features are those that contribute strongly either to or against a prediction as expected.

Sex is a key feature with a high effect and high importance. Age is a key feature with a high effect and high importance. Embarked_C is a key feature with a high effect and high importance.

The features that contribute strongly to this prediction are Sex and Age.

Fig. 3: User inspects scatterplot (highlighted in red).

identified as key features in Fig 5. This allows the user to see that indeed Sex is an important feature here, but so is Age and EmbarkedC which also contribute strongly negatively and positively respectively in comparison to other features.

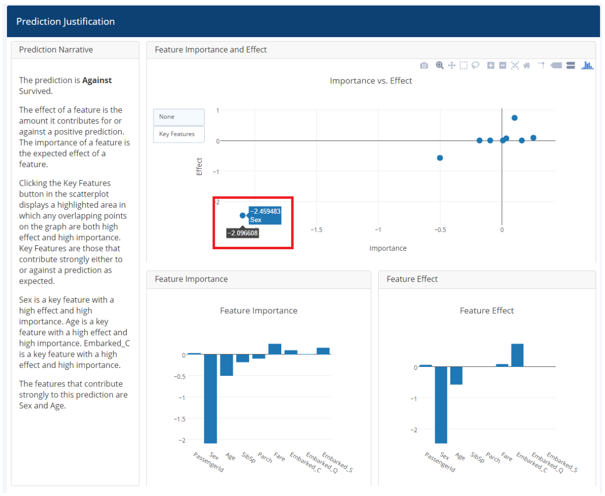
Noting that EmbarkedC is a key feature but is somewhat lacking in importance, the user turns to the importance bar plots to gain a clearer sense of just where EmbarkedC ranks in importance in comparison to the other features in Fig. 6.

The user, now understanding why this prediction against Survived was made, is able to decide to accept the prediction as correct, or decide that the prediction is inaccurate due to its reliance on certain features for this particular prediction.

7 Discussion and Future Works

7.1 Strengths

The main goal of this system was to provide a justification for why a ML prediction is what it is by visualizing the effect and importance of features. It was also important to keep the users of this system in mind - meaning its important to keep in mind these are untrained users, cognitive load needs to be kept low. It was key to the design to maintain simplicity, as adding design complexity on a system that is designed to remove complexity from a ML model would just make for a contradicting design. The system's simplicity also adds to how intuitive



The prediction is **Against** Survived.

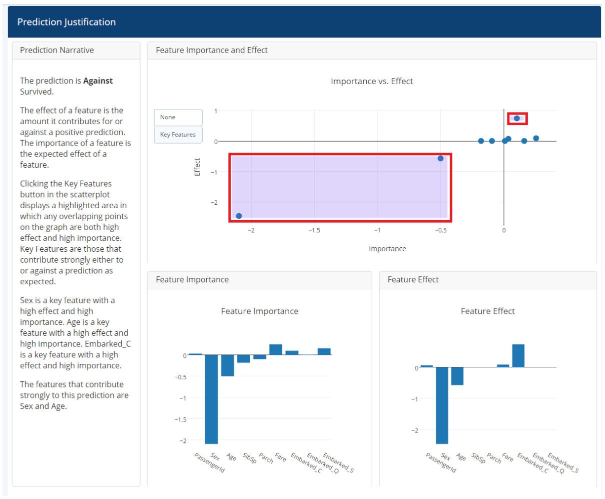
The effect of a feature is the amount it contributes for or against a positive prediction. The importance of a feature is the expected effect of a feature.

Clicking the Key Features button in the scatterplot displays a highlighted area in which any overlapping points on the graph are both high effect and high importance. Key Features are those that contribute strongly either to or against a prediction as expected.

Sex is a key feature with a high effect and high importance. Age is a key feature with a high effect and high importance. Embarked_C is a key feature with a high effect and high importance.

The features that contribute strongly to this prediction are Sex and Age.

Fig. 4: User interacts with chart to see feature names (highlighted in red).


The prediction is **Against** Survived.

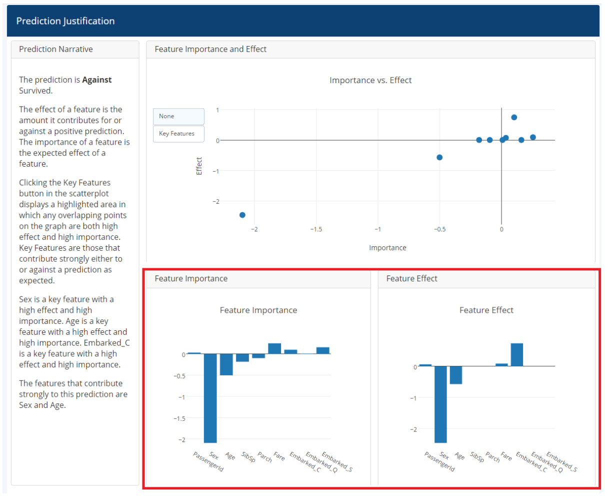
The effect of a feature is the amount it contributes for or against a positive prediction. The importance of a feature is the expected effect of a feature.

Clicking the Key Features button in the scatterplot displays a highlighted area in which any overlapping points on the graph are both high effect and high importance. Key Features are those that contribute strongly either to or against a prediction as expected.

Sex is a key feature with a high effect and high importance. Age is a key feature with a high effect and high importance. Embarked_C is a key feature with a high effect and high importance.

The features that contribute strongly to this prediction are Sex and Age.

Fig. 5: User presses Key Features to see points classified as key features (highlighted in red).


The prediction is **Against** Survived.

The effect of a feature is the amount it contributes for or against a positive prediction. The importance of a feature is the expected effect of a feature.

Clicking the Key Features button in the scatterplot displays a highlighted area in which any overlapping points on the graph are both high effect and high importance. Key Features are those that contribute strongly either to or against a prediction as expected.

Sex is a key feature with a high effect and high importance. Age is a key feature with a high effect and high importance. Embarked_C is a key feature with a high effect and high importance.

The features that contribute strongly to this prediction are Sex and Age.

Fig. 6: User turns to bar charts (highlighted in red).

it is for the user. It's quickly intuitive what the prediction would be without even looking at the text that states the predicted outcome simply by seeing which features are stronger, negative or positive. The system does not have any unnecessary design additions that would risk simply getting in the way of understanding for a user. I believe that this system accomplishes all its original goals. The main strength of this system then, is that it accomplishes its stated goals. The design makes it clear what are key features, where there are outliers (where effect is much higher or lower than expected), and the overall contributions either negatively or positively by the feature set.

Although not specifically stated as a goal, one advantage to the simplicity of the design is also in the speed with which the system works. There is nothing that takes any significant amount of time to render or compute with the system, so it also runs very quickly.

The implementation also presents another strength, which is that the system would be quite easy to embed in a web application. Use of Plotly makes embedding quite straight forward as the entire dashboard structure can be embedded in an iframe. This would allow a web application to simply present a front facing UI that allows a user to upload a dataset with training and test examples and have the backend compute the feature engineering automatically for the user and feed the features into this system. In this sense, the advantage is the ease with which a user could access this system. They would not need to worry about downloading any extra libraries or packages and trying to install them. They would simply need to open a web browser and go to the address with their dataset in hand.

7.2 Weaknesses

A big limitation is that the system is not fully implemented. To realistically be useful the system needs to have a full web application deployment. Currently, there is a contradiction in that in order to be used the user would need to do their own feature engineering. To do this requires enough of a knowledge that the user likely has at least a general sense of the functioning of a logistic regression model anyway, and therefore the system is a bit less effective for them (though the system certainly could still be helpful and interesting). With a full web application deployment a user could simply be prompted to just upload some data set and a backend could handle automatically doing basic feature engineering. Another aspect of the system that could be implemented is the natural language generation aspect for the justification narratives. The narratives could add quite a bit of helpfulness to the justification, but without the natural language generation implemented the text must be manually written.

An additional limitation of this system is that it focuses only on logistic regression. Logistic regression was chosen mainly for its interpretability, not its overall effectiveness. Although some problems certainly can be predicted well by logistic regression, there are lots of types of problems which one would not see very accurate predictions.

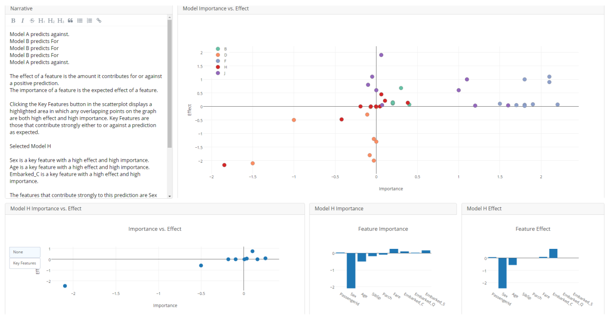
7.3 Limitations

Work on this project was limited by the amount of time available, as well as my unfamiliarity with Plotly. With more time I believe I could've implemented at least a partial start to the natural language generation aspect of the design to have the justification narratives be automatically generated.

If I was more familiar with Plotly to begin with I think I might have been able to implement some linked highlighting effects with the bar charts and scatterplot so that by mousing over a feature on the bar chart the corresponding feature point would be highlighted on the scatterplot or vice versa. Without familiarity however, I ended up spending too much time implementing the rest of the visualizations and did not have time to complete this type of linked highlighting. Additionally, the Plotly documentation is somewhat lacking. The documentation covers the basics quite well but I found in numerous places the documentation was actually wrong or completely missing for more difficult tasks.

7.4 Future Works

As future work I would like to fully implement the natural language generation system. The natural language generation could be a very



Model A predicts against.
Model B predicts For
Model B predicts For
Model B predicts For
Model A predicts against.

The effect of a feature is the amount it contributes for or against a positive prediction.
The importance of a feature is the expected effect of a feature.

Clicking the Key Features button in the scatterplot displays a highlighted area in which any overlapping points on the graph are both high effect and high importance. Key Features are those that contribute strongly either to or against a prediction as expected.

Selected Model H

Sex is a key feature with a high effect and high importance.
Age is a key feature with a high effect and high importance.
Embarked_C is a key feature with a high effect and high importance.

The features that contribute strongly to this prediction are Sex

Fig. 7: A potential redesign that incorporates multiple ML models into the visualization justifications.

effective aspect of the system to have. By implementing this the system could textually explain the outcome of the prediction, provide general instructions such as what effect is and what importance is, point out to the user what the key features are, and identify to the user some outliers or unexpected feature results (such as those with high importance but low effect). It's possible that just having the text alone would be quite helpful in justifying to a user why a particular prediction was made, so being able to combine the text with the visualization should add a effective way of understanding the prediction.

Assuming that I had implemented the natural language generation aspect of the system, I would also add some linked highlighting. With linked highlighting the user could click or mouse over a feature in the textual justification narrative and see the corresponding feature point be highlighted on the scatterplot, as well as on both bar charts. This would help the user logically mesh together all aspects of the design at once and likely contribute strongly to their understanding the prediction. It would also speed up the process of finding features that they read about in the narrative that they find interesting.

I would also definitely want to do a full web application development. Id want to have a UI which allowed users to just select the data set to upload and have the system automatically handle difficult tasks like feature engineering.

A future addition to the system that might be quite effective is an ability to use other ML algorithms besides just logistic regression, or even a way to visualize the effect and importance of features from multiple fit ML algorithms at the same time such as in 7. This could be visualized by plotting the feature effect and importance from multiple ML models into the scatterplot and adding in color hue to identify the different models. The advantage to doing this would be that if one model has a particularly unusual prediction compared to other models, the user might be more inclined to disregard the predictions of that model. The opposite is also of course true, if the user sees 6 models all justifying a prediction in similar ways (similar effect and importance for features) they might feel more comfortable accepting the predictions as accurate.

7.5 Lessons Learned

I certainly feel that I learned to take a much longer look at what sort of tools are out there before deciding on using one for this type of project. I had originally decided to use Matplotlib prior to Plotly, but after beginning the coding I felt it wasn't quite as flexible as I had hoped. I also thought that doing certain tasks in Matplotlib required some 'hacked together' solutions rather than being things that were support natively. I eventually decided to scrap my initial work and move to a new tool which meant that I had less time for implementation. I decided on Plotly thinking it would good looking plots and that the dashboard would make a good UI frame for the system. However after using the tool for a while I found that there were certain tasks that seemed to be unimplemented for Python but were implemented for other languages like Javascript. Not knowing Javascript this didn't help me. I think in the future I would definitely focus a bit more on really closely looking at potential tools before deciding on one.

8 Conclusion

This is a system which intends to justify logistic regression predictions to a non-expert user without requiring the user to understand the underlying workings of the logistic regression model. It accomplishes this by faceting a scatterplot with juxtaposed bar charts and includes a textual justification narrative. The system focuses on simplicity of design and conciseness of idiom expression. Although I believe the system does accomplish the broad view of its goals, it does also allow room for improvement through further implementation. I believe that I have laid the groundwork for an effective full web application deployment in the future of a system that could be effective and widely used as ML continues to see its use expand.

References

[1] O. Biran and K. McKeown. Justification narratives for individual classifications. In *Proceedings of the AutoML workshop at ICML*, vol. 2014, 2014.

[2] O. Biran and K. McKeown. Generating justifications of machine learning predictions., 2015. Retrieved from http://www.cs.unc.edu. doi: techreports/89-022.pdf

[3] M. Kahng, D. Fang, and D. H. P. Chau. Visual exploration of machine learning results using data cube analysis. In *Proceedings of the Workshop on Human-In-the-Loop Data Analytics*, p. 1. ACM, 2016.

[4] J. Krause, A. Perer, and K. Ng. Interacting with predictions: Visual inspection of black-box machine learning models. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, pp. 5686–5697. ACM, 2016.

[5] Z. C. Lipton. The mythos of model interpretability. *arXiv preprint arXiv:1606.03490*, 2016.

[6] M. T. Ribeiro, S. Singh, and C. Guestrin. Model-agnostic interpretability of machine learning. *arXiv preprint arXiv:1606.05386*, 2016.

[7] M. T. Ribeiro, S. Singh, and C. Guestrin. Why should i trust you?: Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 1135–1144. ACM, 2016.