ARTICLE TEMPLATE

Exploring Local Explanations of Nonlinear Models Using Animated Linear Projections

Nicholas Spyrison, Dianne Cook, Przemyslaw Biecek

[a]Monash University, Melbourne, Australia; [b]Warsaw University of Technology and University, Warsaw, Poland

ARTICLE HISTORY
Compiled March 26, 2023

ABSTRACT
The increased predictive power of nonlinear models comes at the cost of interpretability of its terms. This trade-off has led to the emergence of eXplainable AI (XAI). XAI attempts to shed light on how models use predictors to arrive at a prediction with *local explanations*, a point estimate of the linear feature importance in the vicinity of one instance. These can be considered linear projections and can be further explored to understand better the interactions between features used to make predictions across the predictive model surface. Here we describe interactive linear interpolation used for exploration at any instance and illustrate with examples with categorical (penguin species, chocolate types) and quantitative (soccer/football salaries, house prices) output. The methods are implemented in the **R** package **cheem**, available on CRAN.

KEYWORDS
explainable artificial intelligence; nonlinear model interpretability; visual analytics; local explanations; grand tour; radial tour

1. Introduction

There are different reasons and emphases when considering to fit a model. **?** and **?** taxonomize models based on their purpose. *Explanatory* modeling is done for some inferential purpose, while *predictive* modeling focuses more narrowly on the performance of some objective function. The intended use has important implications for model selection and development. In explanatory modeling, interpretability is vital for drawing inferential conclusions. Nonlinear models range from additive models with at least one polynomial term to more complex machine learning models such as random forests, support-vector machines, or neural networks, to name a few (**???**, receptively).

Nonlinear models have many or complexly interacting terms, which cause an opaqueness to the interpretation of the variables. This difficulty in interpreting the terms of complex nonlinear models sometimes lead to them being referred to as black box models. Despite the potentially better performance of nonlinear models, their use is not without controversy (**??**). And the loss of interpretation presents a challenge.

Interpretability is vital for exploring and protecting against potential biases in any

CONTACT Nicholas Spyrison. Email: spyrison@gmail.com, Dianne Cook. Email: dicook@monash.edu, Przemyslaw Biecek. Email: przemyslaw.biecek@gmail.com

model, e.g., sex---?; ?, race---?, and age---?. For instance, models regularly pick up on biases in the training data where such classes correlate with changes in the response variable. This bias is then built into the model. Variable-level (feature-level) interpretability of models is essential in evaluating and addressing such biases.

Another concern is data drift, where a shift in the range of the explanatory variables (features or predictors) between training and test sets. Some nonlinear models are sensitive to this and do not extrapolate well outside the support of the training data (?). Maintaining variable interpretability is also essential to address issues arising from data drift.

Explainable Artificial Intelligence (XAI) is an emerging field of research that provides methods for the interpreting of black box models (??). A common approach is to use *local explanations*, which attempt to approximate linear feature importance at the location of each instance (observation) or the predictions at a specific point in the data domain. Because these are point-specific, it is challenging to comprehensively visualize them to understand a model. There are common approaches for visualising high-dimensional data as a whole, but what is needed are new approaches for viewing these individual local explanations, in relation to the whole.

For multivariate data visualization, a *tour* (???) of linear data projections onto a lower-dimensional space, can help. Tours are viewed as an animation over minor changes to the projection basis. Structure in a projection can then be explored visually to see which features contribute to the formation of that structure. The intuition is similar to watching the shadow of a hidden 3D object change as the object is rotated; watching the shape of the shadow change conveys information about the structure and features of the object.

Applying tours to models has been done in a couple of contexts. Specifically for exploring various statistical model fits and classification boundaries (?) and using tree-based approaches as a projection pursuit index to generate a tour basis paths (??).

There are various types of tours distinguished by method of generating the sequence of projection bases. In a *manual* tour (??), the path is defined by changing the contribution of a selected feature. We propose a radial manual tour can be used to scrutinize a local explanation. Additional interactive elements in a graphical user interface should allow the user to identify an instance of interest and then explore its local explanation by changing feature contribution with the radial tour. The methods are implemented in the **R** package **cheem**. Example data sets are provided to illustrate usage for classification and regression tasks.

Using a radial tour can be compared with counterfactual, what-if analysis, such as *ceteris paribus* profiles (?). Ceteris paribus, is Latin for "other things held constant" or "all else unchanged". These profiles show how an instance's prediction would change from a marginal change in one explanatory feature, given that other features are held constant. It ignores correlations of the features and imagines a case that was not observed. In contrast, our approach is a geometric explanation of the factual; it varies contributions of the features by rotating the basis, a reorientation of the data object. A constraint in our approach is that the basis must remain orthonormal. When the contribution of one feature decreases, the contributions of others necessarily increase such that there is a complete component in that direction. This also ensures that what is seen is strictly a low-dimensional projection from high-dimensional space and is thus an interpretable visualization.

The remainder of this paper is organized as follows. Section @ref(sec:explanations) covers the background of the local explanation and the traditional visuals produced.

Section @ref(sec:tour) explains the tours and particularly the radial manual tour. Section @ref(sec:cheemviewer) discusses the visual layout in the graphical user interface and how it facilitates analysis, data preprocessing, and package infrastructure. Illustrations are provided in Section @ref(sec:casestudies) for a range of supervised learning tasks with categorical and quantitative outputs. Section @ref(sec:cheemdiscussion) concludes with a summary of the insights gained.

2. Local Explanations

Consider a highly nonlinear model. It can be hard to determine whether small changes in a feature's value will make a class prediction change group or identify which features contribute to an extreme residual. Local explanations shed light on these situations by approximating linear feature importance in the vicinity of a single instance.

A comprehensive summary of the taxonomy and literature on explanation techniques is provided in Figure 6 by **?**. It includes a large number of model-specific explanations such as deepLIFT (**??**), a popular recursive method for estimating importance in neural networks. There are fewer model-agnostic explanations, of which LIME, (**?**) SHAP, (**?**), and their variants are popular.

These instance-level explanations are used in various ways depending on the data. In image classification, where pixels correspond to predictors, saliency maps overlay or offset a heatmap indicating important pixels (**?**). For instance, pixels corresponding to snow may be highlighted when distinguishing if a picture contains a wolf or husky. In text analysis, word-level contextual sentiment analysis highlights the sentiment and magnitude of influential words (**?**). In the case of numeric regression, they are used to explain feature additive contributions from the model intercept to the instance's prediction (**?**).

SHaply Additive exPlanations (SHAP) quantifies the feature contributions of one instance by examining the effect of other features on the predictions. The SHAP explanation refers to **?**'s method to evaluate an individual's contribution in cooperative games by assessing this player's performance in the presence or absence of other players. **?** introduced SHAP for local explanations in ML models. The attribution of feature importance depends on the sequence of the included features. The SHAP values are the mean contributions over different feature sequences. The approach is related to partial dependence plots (**?**), used to explain the effect of a feature by predicting the response for a range of values on this feature after fixing the value of all other features to their mean. Though partial dependence plots are a global approximation of the feature importance, while SHAP is specific to one instance. It could also be considered similar to examining the coefficients from all subsets regression, as described in **?** help to understand the relative importance of each feature in the context of all other candidate features.

Following the use case *Explanatory Model Analysis* (**?**), FIFA data is used to illustrate SHAP. Consider soccer data from the FIFA 2020 season (**?**). There are 5000 instances of 9 skill measures (after aggregating highly correlated features). A random forest model is fit regressing players' wages [2020 Euros] from their skill measurements. The SHAP values are compared for a star offensive player (L. Messi) and defensive player (V. van Dijk). The results are displayed in Figure @ref(fig:shapdistrbd). A difference in the attribution of the feature importance across the two positions of the players can be expected. This would be interpreted as how a player's salary depends on this combination of skills. Panel (a) is a modified breakdown plot (**?**) where three

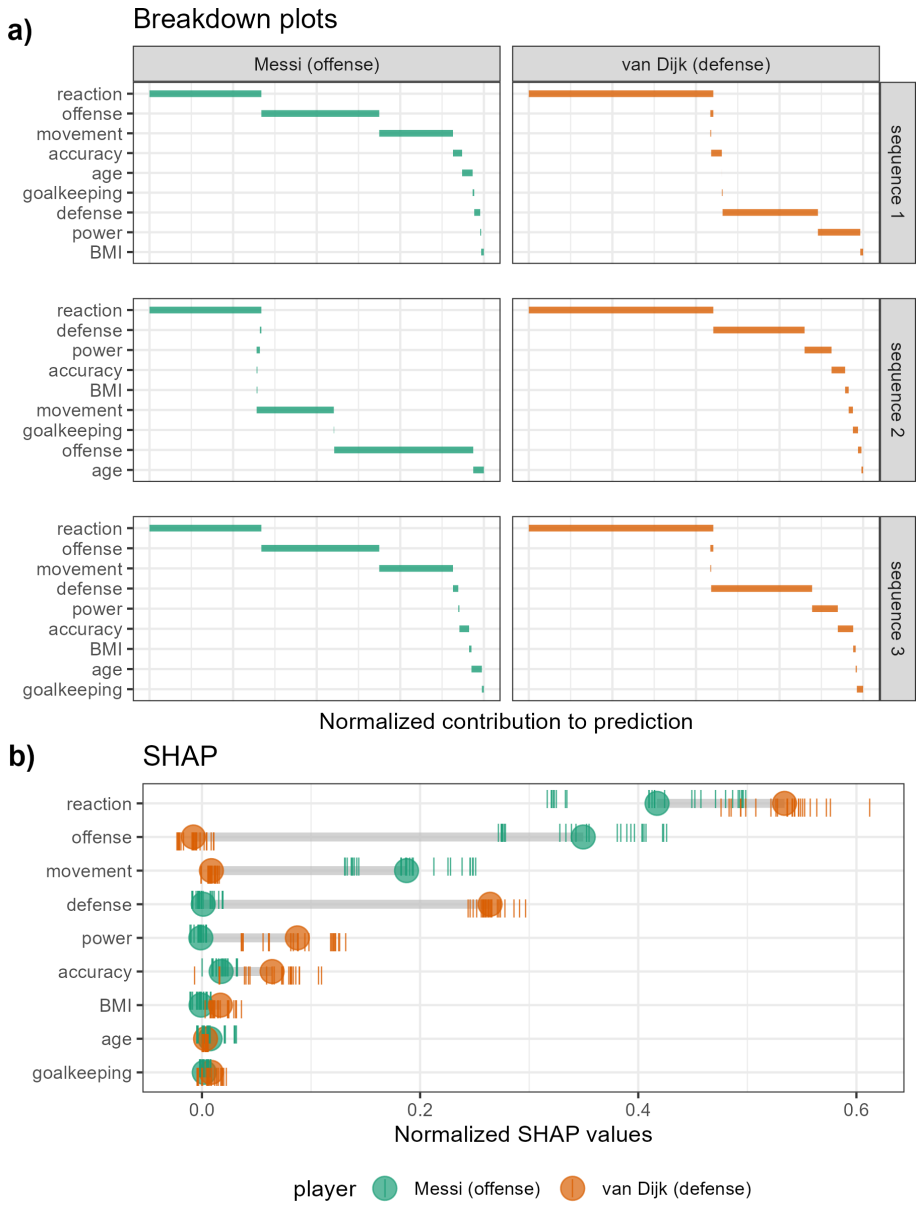

Figure 1. Illustration of SHAP values for a random forest model FIFA 2020 player wages from nine skill predictors. A star offensive and defensive player are compared, L. Messi and V. van Dijk, respectively. Panel (a) shows breakdown plots of three sequences of the features. The sequence of the variables impacts the magnitude of their attribution. Panel (b) shows the distribution of attribution for each feature across 25 sequences of predictors, with the mean displayed as a dot for each player. Reaction skills are important for both players. Offense and movement are important for Messi but not van Dijk, and conversely, defense and power are important for van Dijk but not Messi.

sequences of features are presented, so the two instances can be more easily compared. The magnitude of the contributions depends on the sequence in which they appear. Panel (b) shows the differences in the player's median values of over 25 such sequences. In summary, these plots highlight how local explanations bring interpretability to a model, at least in the vicinity of their instances. In this instance, two players with different positions receive different profiles of feature importance to explain the prediction of their wages.

For the application, we use *tree SHAP*, a variant of SHAP that enjoys a lower computational complexity (**?**). Instead of aggregating over sequences of the features, tree SHAP calculates instance-level feature importance by exploring the structure of the decision trees. Tree SHAP is only compatible with tree-based models; random forests are used for illustration. The following section will use normalized SHAP values as a projection basis (call this the *attribution projection*) that will be used to explore the sensitivity of the feature contributions.

3. Tours and the Radial Tour

A *tour* enables the viewing of high-dimensional data by animating many linear projections with small incremental changes. It is achieved by following a path of linear projections (bases) of high-dimensional space. One key feature of the tour is the object permanence of the data points; one can track the relative change of instances in time and gain information about the relationships between points across multiple features. There are various types of tours that are distinguished by how the paths are generated (**??**).

The manual tour (**?**) defines its path by changing a selected feature's contribution to a basis to allow the feature to contribute more or less to the projection. The requirement constrains the contribution of all other features that a basis needs to be orthonormal (column correspond to vectors, with unit length, and orthogonal to each other). The manual tour is primarily used to assess the importance of a feature to structure visible in a projection. It also lends itself to pre-computation queued in advance or computed on-the-fly for human-in-the-loop analysis (**?**).

A version of the manual tour called a *radial tour* is implemented in **?** and forms the basis of the new work. In a radial tour, the selected feature can change its magnitude of contribution but not its angle; it must move along the direction of its original contribution. The implementation allows for pre-computation and interactive re-calculation to focus on a different feature.

4. Shapley's Lenses

Shapley values are a useful technique for understanding the nature of the relationship between the input variables and the target variable. The data exploration process is navigated by a trained predictive model.

Thus, local attributions can be viewed as a new representation of data (commonly referred as data embeddings by machine learning community) with certain desired properties, e.g., the unit is standardized between variables because it reflects the relation with target variable, so it is expressed on the scale of the target variable.

But different models can produce different local attributions. This phenomenon, known as the multiplicity of good models or as the Rashomon set of equally good

models, means that models that fit the data equally well can describe the relationships present in the data in different ways.

The response to the challenge in the characterization of Rashomon sets may be Shapley's Lenses - the opportunity to compare these models from the perspective of exploring the local attributions generated by the Shapley values for different models.

5. The Cheem Viewer

To explore the local explanations, coordinated views (**?**) (also known as ensemble graphics, **?**) are provided in the *cheem viewer* application. There are two primary plots: the **global view** to give the context of all of the SHAP values and the **radial tour view** to explore the local explanations with user-controlled rotation. There are numerous user inputs, including feature selection for the radial tour and instance selection for making comparisons. There are different plots used for the categorical and quantitative responses. Figures @ref(fig:classificationcase) and @ref(fig:regressioncase) are screenshots showing the cheem viewer for the two primary tasks: classification (categorical response) and regression (quantitative response).

5.1. *Global View*

The global view provides context for all instances and facilitates the exploration of the separability of the data- and attribution-spaces. These spaces both have dimensionality $n \times p$, where n is the number of instances and p is the number of features. The attribution space corresponds to the local explanations for each instance; feature importance in the vicinity of the instance.

The visualization is composed of the first two principal components of the data (left) and the attribution (middle) spaces. These single 2D projections will not reveal all of the structure of higher-dimensional space, but they are helpful visual summaries. In addition, a plot of the observed against predicted response values is also provided (Figures @ref(fig:classificationcase)b, @ref(fig:regressioncase)a) to help identify instances poorly predicted by the model. For classification tasks, misclassified instances are circled in red. Linked brushing between the plots is provided, and a tabular display of selected points helps to facilitate exploration of the spaces and the model (shown in Figures @ref(fig:regressioncase)d).

While the comparison of these spaces is interesting, the primary purpose of the global view is to enable the selection of particular instances to explore in detail. The projection attribution of the primary instance (PI) is examined and typically viewed with an optional comparison instance (CI). These instances are highlighted as asterisk and $\times$, respectively.

5.2. *Radial Tour*

The local explanations for all observations are normalized (sum of squares equals 1), and thus, the relative importance of features can be compared across all instances. These are depicted as vertical parallel coordinate plots (**?**) on the basis biplot (**?**). 1D biplot displays the values of the current basis as bars. The parallel coordinate overlays lines connecting one instance's feature attribution (Figures @ref(fig:classificationcase)e and @ref(fig:regressioncase)e). The attribution projections of the PI and CI are shown

as dashed and dotted lines. From this plot, the range and density of the importance across all instances can be interpreted. For classification, one would look at differences between groups on any feature. For example, Figure @ref(fig:classificationcase)e suggests that `bl` is important for distinguishing the green class from the other two. For regression, one might generally observe which features have low values for all instances (not important). For example, `BMI` and `pwr` in Figure @ref(fig:regressioncase)e, have a range of high and low values (e.g., `off`, `def`), suggesting they are important for some instances and not important for others.

The overlaid bars on the parallel coordinate plot represent the attribution projection of the PI. (Remember that the PI is interactively selected from the global view). The attribution projection approximates the feature importance for predicting this instance. The combination of features best explains the difference between the mean response and an instance's predicted value. It is not an indication of the local shape of the model surface. That is, it is not some indication of the tangent to the curve at this point.

The attribution projection of the PI is the initial 1D basis in a radial tour, displayed as a density plot for a categorical response (Figure @ref(fig:classificationcase)f) and as scatterplots for a quantitative response (Figure @ref(fig:regressioncase)f). The PI and CI are indicated by vertical dashed and dotted lines. The radial tour varies the contribution of the selected feature between 0 and 1. This is viewed as an animation of the projections from many intermediate bases. Doing so tests the sensitivity of structure (class separation or strength of relationship) to the feature's contribution. For classification, if the separation between classes diminishes when the feature contribution is reduced, this suggests that the feature is important for class separation. For regression, if the relationship scatterplot weakens when the feature contribution is reduced, indicating that the feature is important for accurately predicting the response.

5.3. *Classification Task*

Selecting a misclassified instance as PI and a correctly classified point nearby in data space as CI makes it easier to examine the features most responsible for the error. The global view (Figure @ref(fig:classificationcase)c) displays the model confusion matrix. The radial tour is 1D and displays as density where color indicates class. An animation slider enables users to vary the contribution of features to explore the sensitivity of the separation to that feature.

5.4. *Regression Task*

Selecting an inaccurately predicted instance as PI and an accurately predicted instance with similar feature values as CI is a helpful way to understand how the model is failing or not. The global view (Figure @ref(fig:regressioncase)a) shows a scatterplot of the observed vs predicted values, which should exhibit a strong relationship if the model is a good fit. The points can be colored by a statistic, residual, a measure of outlyingness (log Mahalanobis distance), or correlation to aid in understanding the structure identified in these spaces.

In the radial tour view, the observed response and the residuals (vertical) are plotted against the attribution projection of the PI (horizontal). The attribution projection can be interpreted similarly to the predicted value from the global view plot. It represents a linear combination of the features, and a good fit would be indicated when there is

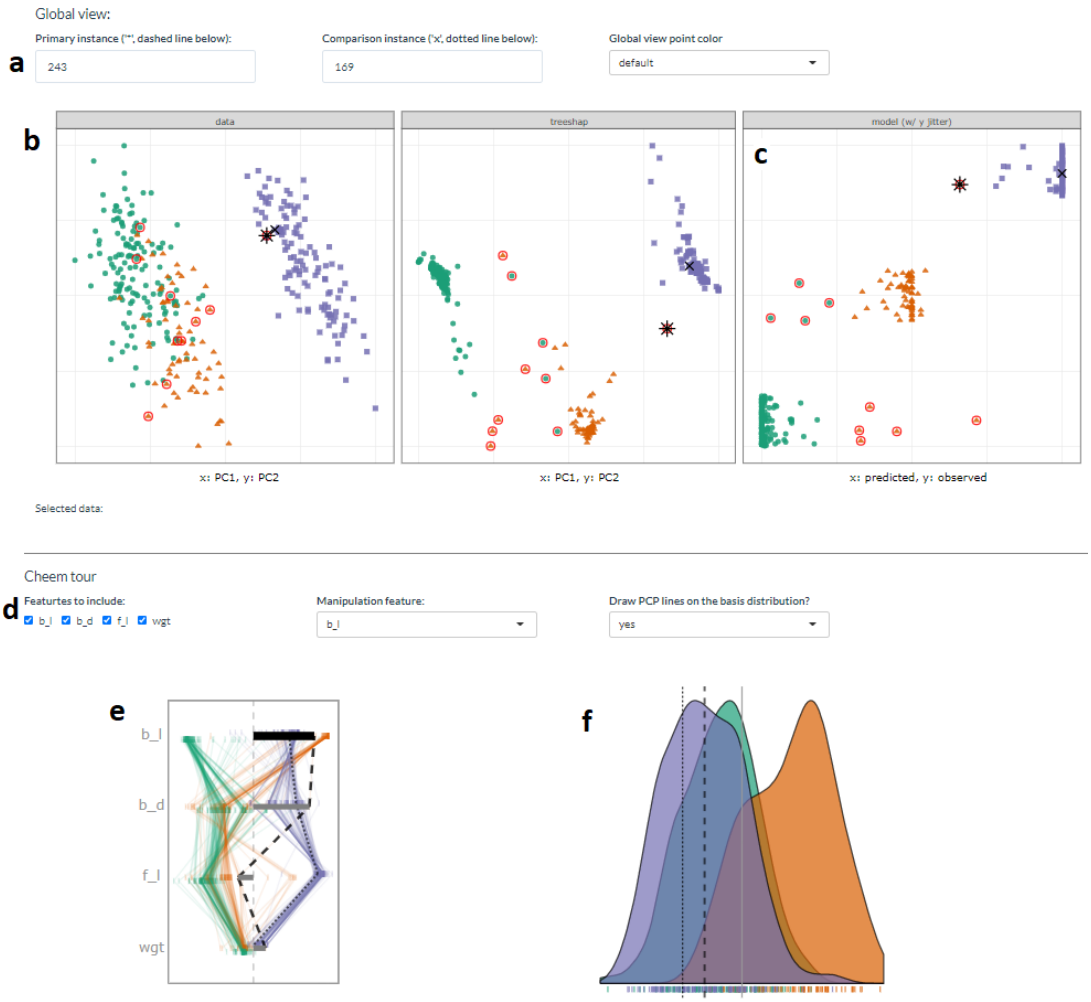

Figure 2. Overview of the cheem viewer for classification tasks (categorical outputs). Global view inputs, (a), set the PI, CI, and color statistic. Global view, (b) PC1 by PC2 approximations of the data- and attribution-space. (c) prediction by observed y (visual of the confusion matrix for classification tasks). Points are colored by predicted class, and red circles indicate misclassified instances. Radial tour inputs (d) select features to include and which feature is changed in the tour. (e) shows a parallel coordinate display of the distribution of the feature attributions while bars depict contribution for the current basis. The black bar is the variable being changed in the radial tour. Panel (f) is the resulting data projection indicated as density in the classification case.

a strong relationship with the observed values. This can be viewed as a local linear approximation if the fitted model is nonlinear. As the contribution of a feature is varied, if the value of the PI does not change much, it would indicate that the prediction for this instance is NOT sensitive to that feature. Conversely, if the predicted value varies substantially, the prediction is very sensitive to that feature, suggesting that the feature is very important for the PI's prediction.

5.5. *Interactive Features*

The application has several reactive inputs that affect the data used, aesthetic display, and tour manipulation. These reactive inputs make the software flexible and extensible (Figure @ref(fig:classificationcase)a & d). The application also has more exploratory interactions to help link points across displays, reveal structures found in different spaces, and access the original data.

A tooltip displays instance number/name and classification information while the cursor hovers over a point. Linked brushing allows the selection of points (left click and drag) where those points will be highlighted across plots (Figure @ref(fig:classificationcase)a & b). The information corresponding to the selected points is populated on a dynamic table (Figure @ref(fig:classificationcase)d). These interactions aid exploration of the spaces and, finally, the identification of primary and comparison instances.

5.6. *Preprocessing*

It is vital to mitigate the render time of visuals, especially when users may want to iterate many explorations. All computational operations should be prepared before runtime. The work remaining when an application is run solely reacts to inputs and rendering visuals and tables. Below discusses the steps and details of the reprocessing.

- **Data:** predictors and response are unscaled complete numerical matrix. Most models and local explanations are scale-invariant. Keep normality assumptions of the model in mind.
- **Model:** any model and compatible explanation could be explored with this method. Currently, random forest models are applied via the package **randomForest** (**?**), compatibility tree SHAP. Modest hyperparameters are used, namely: 125 trees, the number of variables at each split, mtry = $\sqrt{p}$ or $p/3$ for classification and regression, and minimum size of terminal nodes $max(1, n/500)$ or $max(5, n/500)$ for classification and regression.
- **Local explanation:** Tree SHAP is calculated for *each* observation using the package **treeshap** (**?**). We opt to find the attribution of each instance in the training data and not fit to fit feature interactions.
- **Cheem viewer:** after the model and full explanation space are calculated, each variable is scaled by standard deviations away from the mean to achieve common support for visuals. Statistics for mapping to color are computed on the scaled spaces.

The time to preprocess the data will vary significantly with the complexity of the model and local explanation. For reference, the FIFA data contained 5000 instances of nine explanatory features took 2.5 seconds to fit a random forest model of modest hyperparameters. Extracting the tree SHAP values of each instance took 270 seconds

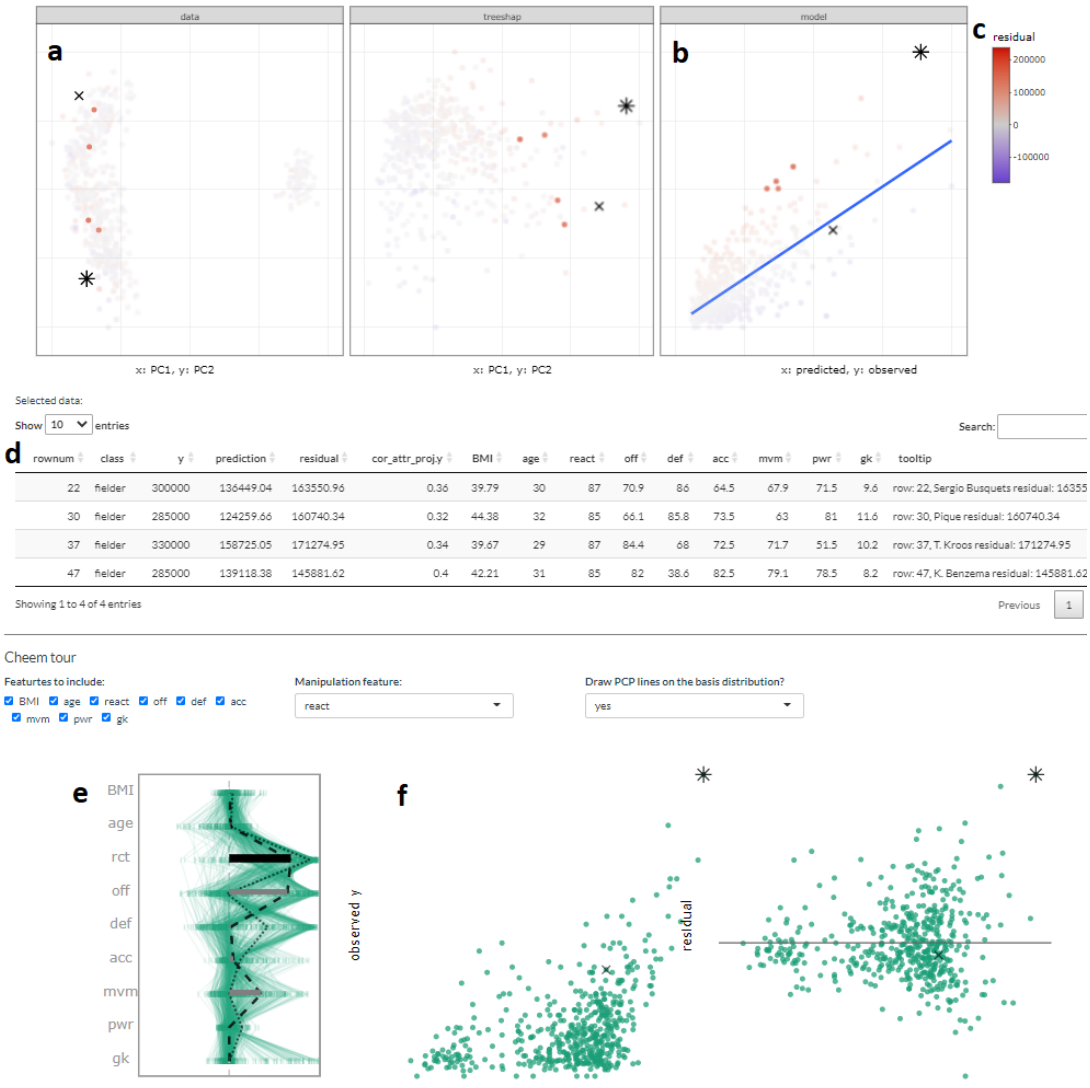

Figure 3. Overview of the cheem viewer for regression tasks (quantitative outputs) and illustration of interactive features. Panel (a) PCA of the data- and attributions- spaces and the (b) residual plot, predictions by observed values. Four selected points are highlighted in the PC spaces and tabularly displayed. Coloring on a statistic (c) highlights structure organized in the attribution space. Interactive tabular display (d) populates when instances are selected. Contribution of the 1D basis affecting the horizontal position (e) parallel coordinate display of the feature attribution from all observations, and horizontal bars show the contribution to the current basis. Regression projection (f) uses the same horizontal projection and fixes the vertical positions to the observed y and residuals (middle and right).

total. PCA and statistics of the features and attributions took 2.8 seconds. These runtimes were from a non-parallelized session on a modern laptop, but suffice to say that most of the time will be spent on the local attribution. An increase in model complexity or data dimensionality will quickly become an obstacle. Its reduced computational complexity makes tree SHAP an excellent candidate to start. (Alternatively, the package **fastshap** claims extremely low runtimes, attributed to fewer calls to the prediction function, partial implementation in C++, and efficient use of logical subsetting, **?**)

5.7. *Package Infrastructure*

The above-described method and application are implemented as an open-source **R** package, **cheem** available on CRAN at `https://CRAN.R-project.org/package=cheem`. Preprocessing was facilitated with models created via **randomForest** (**?**) and explanations calculated with **treeshap** (**?**). The application was made with **shiny** (**?**). The tour visual is built with **spinifex** (**?**). Both views are created first with **ggplot2** (**?**) and then rendered as interactive `html` widgets with **plotly** (**?**). **DALEX** (**?**) and the free ebook, *Explanatory Model Analysis* (**?**) were a boon to understanding local explanations and how to apply them.

5.8. *Installation and Getting Started*

The package can be installed from CRAN, and the application can be run using the following **R** code:

```
install.packages("cheem", dependencies = TRUE)
library("cheem")
run_app()
```

Alternatively,

- A version of the cheem viewer shiny app can be directly accessed at `https://ebsmonash.shinyapps.io/cheem_initial/`.
- The development version of the package is available at `https://github.com/nspyrison/cheem`, and
- Documentation of the package can be found at `https://nspyrison.github.io/cheem/`.

Follow the examples provided with the package to compute the local explanation (using `?cheem_ls`). The application expects the output returned by `cheem_ls()`, saved to an `rds` file with `saveRDS()` to be uploaded.

6. Case Studies

To illustrate the cheem method it is applied to modern data sets, two classification examples and then two of regression.

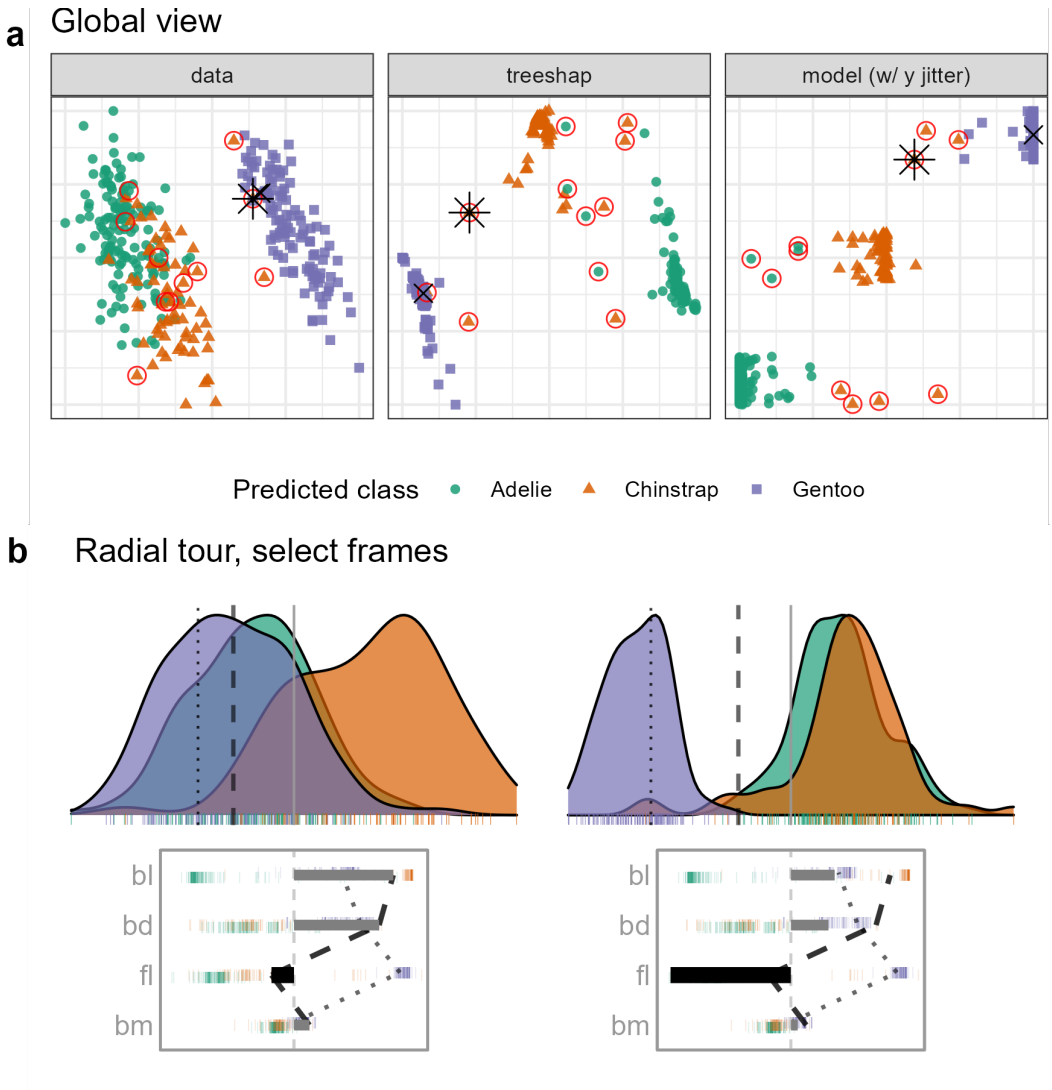

Figure 4. (ref:casepenguins-cap)

6.1. *Palmer Penguin, Species Classification*

The Palmer penguins data (**??**) was collected on three species of penguins foraging near Palmer Station, Antarctica. The data is publicly available to substitute for the overly-used iris data and is quite similar in form. After removing incomplete instances, there are 333 instances of four physical measurements, bill length (`bl`), bill depth (`bd`), flipper length (`fl`), and body mass (`bm`) for this illustration. A random forest model was fit with species as the response feature.

(ref:casepenguins-cap) Examining the SHAP values for a random forest model classifying Palmer penguin species. The PI is a Gentoo (purple) penguin that is misclassified as a Chinstrap (orange), marked as an asterisk in (a) and the dashed vertical line in (b). The radial view shows varying the contribution of `fl` from the initial attribution projection (b, left), which produces a linear combination where the PI is more probably (higher density value) a Chinstrap than a Gentoo (b, right). (The animation of the radial tour is at `https://vimeo.com/666431172`.)

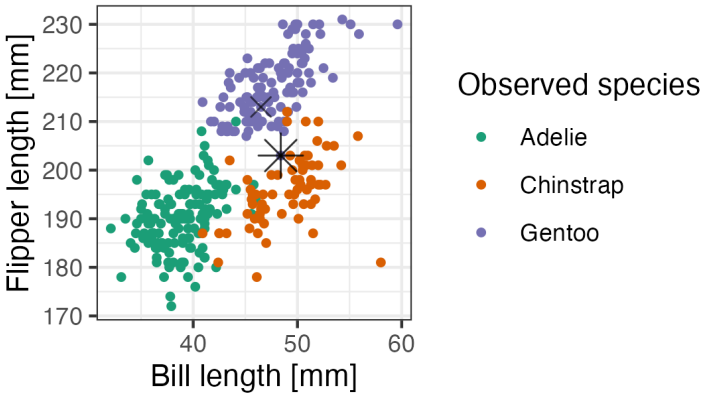

Figure 5. (ref:casepenguinsblfl-cap)

Figure @ref(fig:casepenguins) shows plots from the cheem viewer for exploring the random forest model on the penguins data. Panel (a) shows the global view, and panel (b) shows several 1D projections generated with the radial tour. Penguin 243, a Gentoo (purple), is the PI because it has been misclassified as a Chinstrap (orange).

(ref:casepenguinsblfl-cap) Checking what is learned from the cheem viewer. This is a plot of flipper length (`fl`) and bill length (`bl`), where an asterisk highlights the PI. A Gentoo (purple) misclassified as a Chinstrap (orange). The PI has an unusually small `fl` length which is why it is confused with a Chinstrap.

There is more separation visible in the attribution space than in the data space, as would be expected. The predicted vs observed plot reveals a handful of misclassified instances. A Gentoo that has been wrongly labeled as a Chinstrap is selected for illustration. The PI is a misclassified point (represented by the asterisk in the global view and a dashed vertical line in the tour view). The CI is a correctly classified point (represented by an $\times$ and a vertical dotted line).

The radial tour starts from the attribution projection of the misclassified instance (b, left). The important features identified by SHAP in the (wrong) prediction for this instance are mostly `bl` and `bd` with small contributions of `fl` and `bm`. This projection is a view where the Gentoo (purple) looks much more likely for this instance than Chinstrap. That is, this combination of features is not particularly useful because the PI looks very much like other Gentoo penguins. The radial tour is used to vary the contribution of flipper length (`fl`) to explore this. (In our exploration, this was the third feature explored. It is typically helpful to explore the features with more significant contributions, here `bl` and `bd`. Still, when doing this, nothing was revealed about how the PI differed from other Gentoos). On varying `fl` as it contributes increasingly to the projection (b, right), more and more, this penguin looks like a Chinstrap. This suggests that `fl` should be considered an important feature for explaining the (wrong) prediction.

Figure @ref(fig:casepenguinsblfl) confirms that flipper length (`fl`) is vital for the confusion of the PI as a Chinstrap. Here, flipper length and body length are plotted, and the PI can be seen to be closer to the Chinstrap group in these two features, mainly because it has an unusually low value of flipper length relative to other Gentoos. From this view, it makes sense that it is a hard instance to account for, as decision trees can only partition only vertical and horizontal lines.

6.2. *Chocolates, Milk/Dark Classification*

The chocolates data set consists of 88 instances of ten nutritional measurements determined from their labels and labeled as either milk or dark. Dark chocolate is considered healthier than milk. Students collected the data during the Iowa State University class STAT503 from nutritional information on the manufacturer's websites and were normalized to 100g equivalents. The data is available in the **cheem** package. A random forest model is used for the classification of chocolate types.

It could be interesting to examine the nutritional properties of any dark chocolates that have been misclassified as milk. A reason to do this is that a dark chocolate, nutritionally more like milk should not be considered a healthy alternative. It is interesting to explore which nutritional features contribute most to misclassification.

(ref:casechocolates-cap) Examining the local explanation for a PI which is dark (orange) chocolate incorrectly predicted to be milk (green). From the attribution projection, this chocolate correctly looks more like dark than milk, which suggests that the local explanation does not help understand the prediction for this instance. So, the contribution of Sugar is varied—reducing it corresponds primarily with an increased magnitude from Fiber. When Sugar is zero, Fiber contributes strongly towards the left. In this view, the PI is closer to the bulk of the milk chocolates, suggesting that the prediction put a lot of importance on Fiber. This chocolate is a rare dark chocolate without any Fiber leading to it being mistaken for a milk chocolate. (A video of the tour animation can be found at `https://vimeo.com/666431143`.)

This type of exploration is shown in Figure @ref(fig:casechocolates), where a chocolate labeled dark but predicted to be milk is chosen as the PI (instance 22). It is compared with a CI that is a correctly classified dark chocolate (instance 7). The PCA plot and the tree SHAP PCA plots (a) show a big difference between the two chocolate types but with confusion for a handful of instances. The misclassifications are more apparent in the observed vs predicted plot and can be seen to be mistaken in both ways: milk to dark and dark to milk.

The attribution projection for chocolate 22 suggests that Fiber, Sugars, and Calories are most responsible for its incorrect prediction. The way to read this plot is to see that Fiber has a large negative value while Sugars and Calories have reasonably large positive values. In the density plot, instances on the very left of the display would have high values of Fiber (matching the negative projection coefficient) and low values of Sugars and Calories. The opposite would be interpreting a point with high values in this plot. The dark chocolates (orange) are primarily on the left, and this is a reason why they are considered to be healthier: high fiber and low sugar. The density of milk chocolates is further to the right, indicating that they generally have low fiber and high sugar.

The PI (dashed line) can be viewed against the CI (dotted line). Now, one needs to pay attention to the parallel plot of the SHAP values, which are local to a particular instance, and the density plot, which is the same projection of all instances as specified by the SHAP values of the PI. The feature contribution of the two different predictions can be quickly compared in the parallel coordinate plot. The PI differs from the comparison primarily on the Fiber feature, which suggests that this is the reason for the incorrect prediction.

From the density plot, which is the attribution projection corresponding to the PI, both instances are more like dark chocolates. Varying the contribution of Sugars and altogether removing it from the projection is where the difference becomes apparent. When a frame with contribution primarily from Fiber is examined instance 22 looks

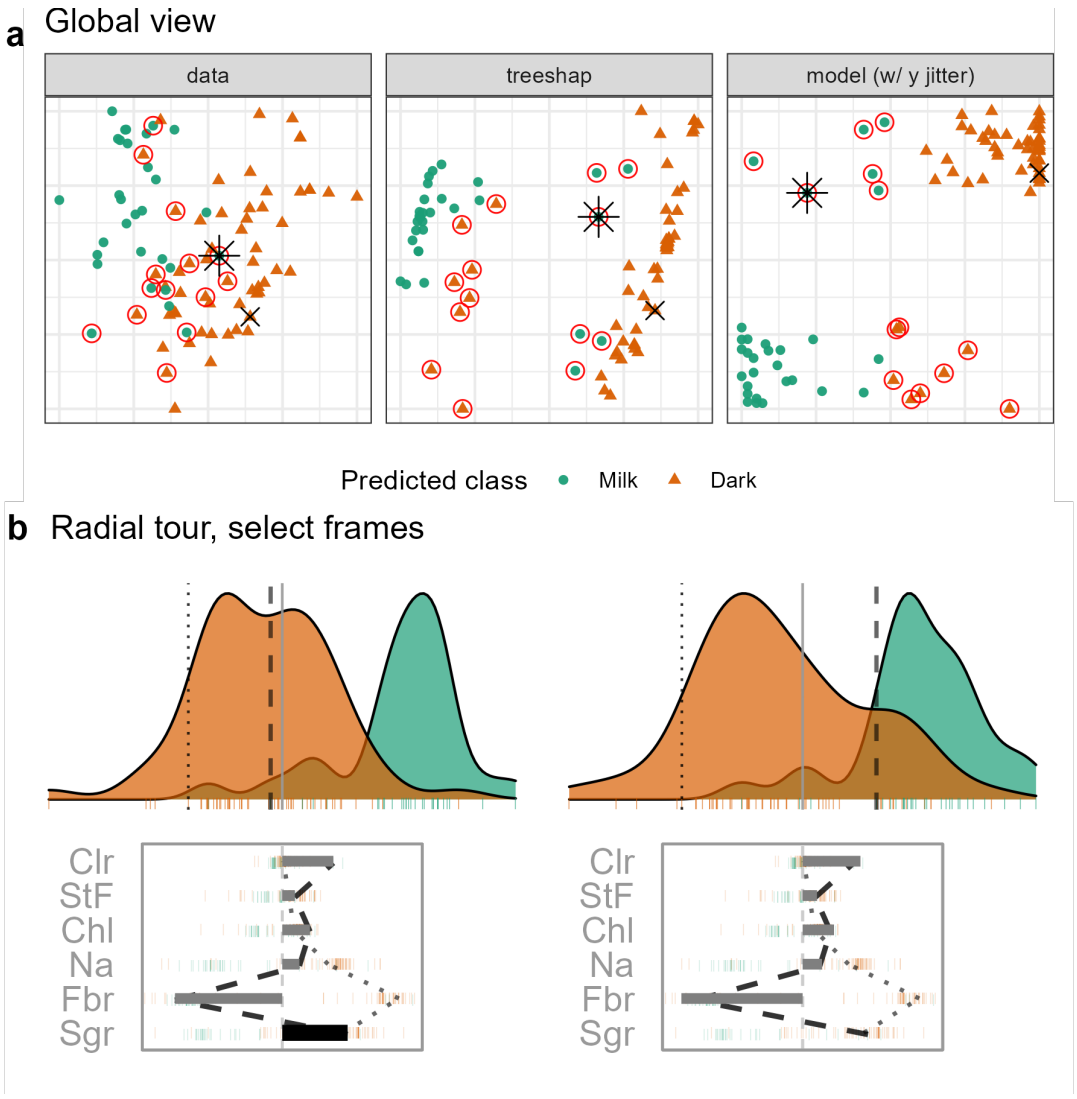

Figure 6. (ref:casechocolates-cap)

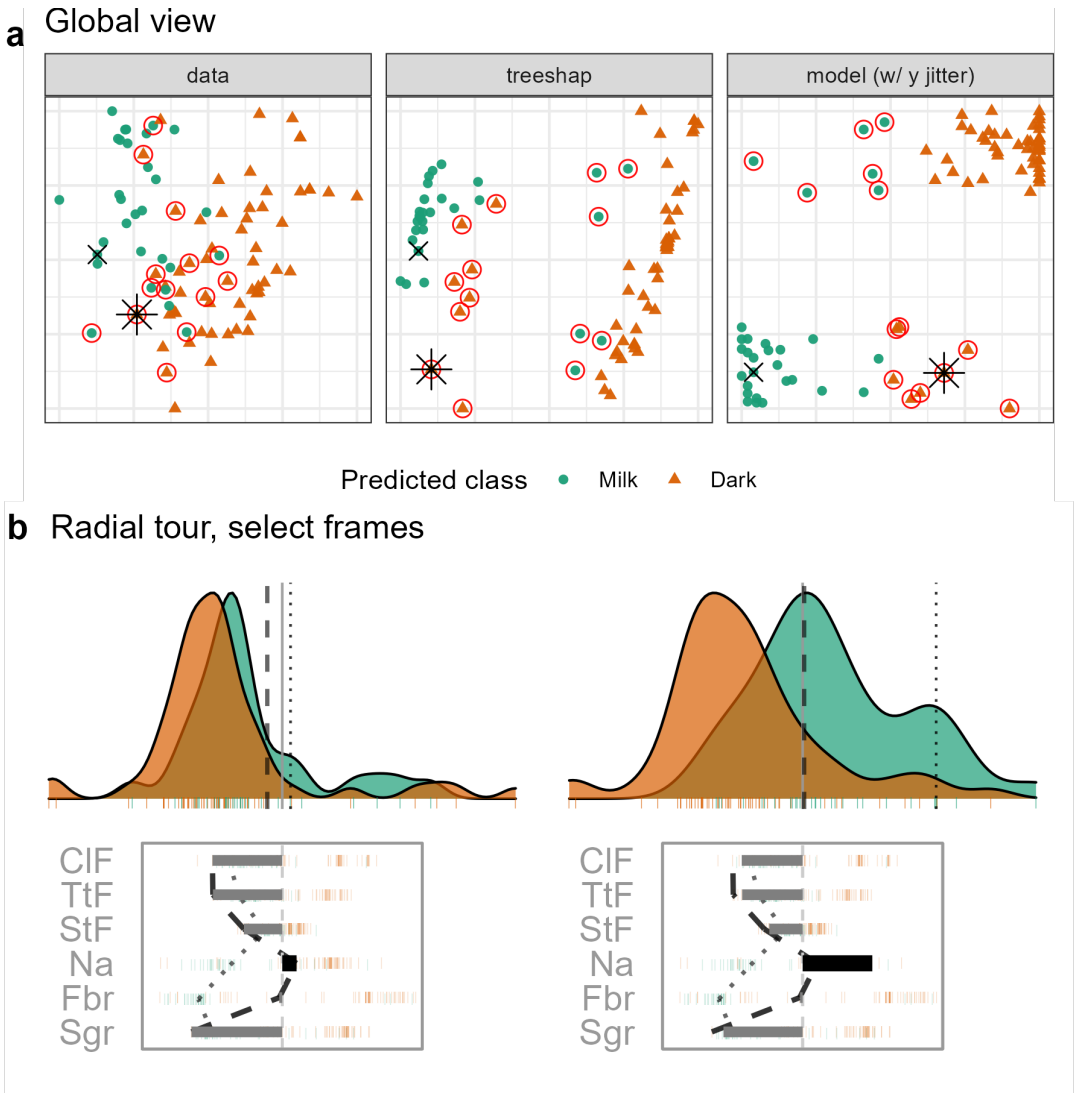

Figure 7. (ref:casechocolatesinverse-cap)

more like a milk chocolate.

It would also be interesting to explore an inverse misclassification. In this case, a milk chocolate is selected while it was misclassified as a dark chocolate. Chocolate 84 is just this case and is compared with a correctly predicted milk chocolate (instance 71). The corresponding global view and radial tour frames are shown in Figure @ref(fig:casechocolatesinverse).

(ref:casechocolatesinverse-cap) Examining the local explanation for a PI which is milk (green) chocolate incorrectly predicted to be dark (orange). In the attribution projection, the PI could be either milk or dark. Sodium and Fiber have the largest differences in attributed feature importance, with low values relative to other milk chocolates. The lack of importance attributed to these variables is suspected of contributing to the mistake, so the contribution of Sodium is varied. If Sodium had a larger contribution to the prediction (like in this view). the PI would look more like other milk chocolates. (A video of the tour animation can be found at `https://vimeo.com/666431148`.)

The difference of position in the tree SHAP PCA with the previous case is quite significant; this gives a higher-level sense that the attributions should be quite different. Looking at the attribution projection, this is found to be the case. Previously, Fiber was essential while it is absent from the attribution in this case. Conversely, Calories from Fat and Total Fat have high attributions here, while they were unimportant in the preceding case.

Comparing the attribution with the CI (dotted line), large discrepancies in Sodium and Fiber are identified. The contribution of Sodium is selected to be varied. Even in the initial projection, the instance looks slightly more like its observed milk than predicted dark chocolate. The misclassification appears least supported when the basis reaches sodium attribution of typical dark chocolate.

6.3. *FIFA, Wage Regression*

The 2020 season FIFA data (**??**) contains many skill measurements of soccer/football players and wage information. Nine higher-level skill groupings were identified and aggregated from highly correlated features. A random forest model is fit from these predictors, regressing player wages [2020 euros]. The model was fit from 5000 instances before being thinned to 500 players to mitigate occlusion and render time. Continuing from the exploration in Section @ref(sec:explanations), we are interested to see the difference in attribution based on the exogenous player position. That is, the model should be able to use multiple linear profiles to better predict the wages from different field positions of players despite not having this information. A leading offensive fielder (L. Messi) is compared with a top defensive fielder (V. van Dijk). The same instances were used in Figure @ref(fig:shapdistrbd).

(ref:casefifa-cap) Exploring the wages relative to skill measurements in the FIFA 2020 data. Star offensive player (L. Messi) is the PI, and he is compared with a top defensive player (V. van Dijk). The attribution projection is shown on the left, and it can be seen that this combination of features produces a view where Messi has very high predicted (and observed) wages. Defense (`def`) is the chosen feature to vary. It starts very low, and Messi's predicted wages decrease dramatically as its contribution increases (right plot). The increased contribution in defense comes at the expense of offensive and reaction skills. The interpretation is that Messi's high wages are most attributable to his offensive and reaction skills, as initially provided by the local explanation. (A video of the animated radial tour can be found at `https://vimeo.com/666431163`.)

Figure @ref(fig:casefifa) tests the support of the local explanation. Offensive and reaction skills (`off` and `rct`) are both crucial to explaining a star offensive player. If either of them were rotated out, the other would be rotated into the frame, maintaining a far-right position. However, increasing the contribution of a variable with low importance would rotate both features out of the frame.

The contribution from `def` will be varied to contrast with offensive skills. As the contribution of defensive skills increases, Messi's is no longer separated from the group. Players with high values in defensive skills are now the rightmost points. In terms of what-if analysis, the difference between the data mean and his predicted wages would be halved if Messi's tree SHAP attributions were at these levels.

(ref:caseames-cap) Exploring an instance with an extreme residual as the PI in relation to an instance with an accurate prediction for a similarly priced house in a random forest fit to the Ames housing data. The local explanation indicates a sizable

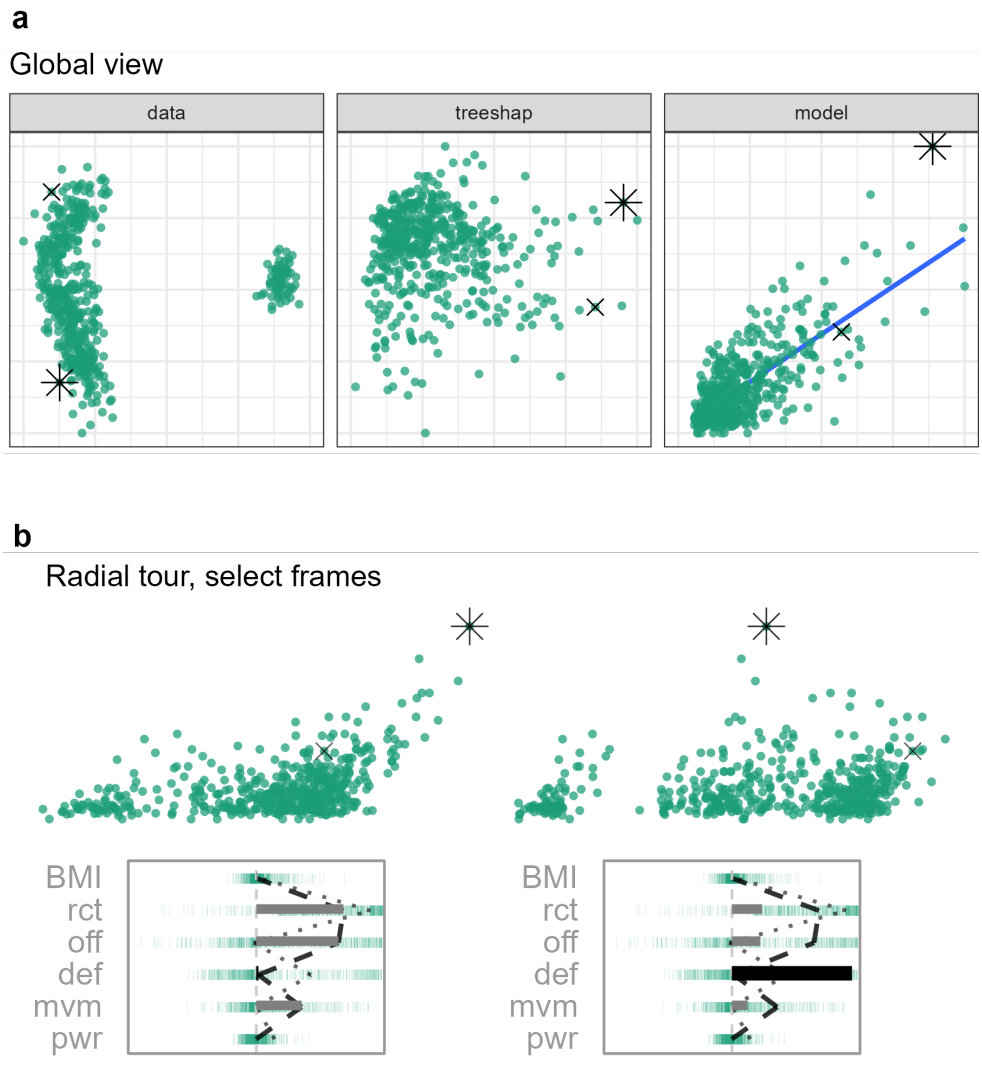

Figure 8. (ref:casefifa-cap)

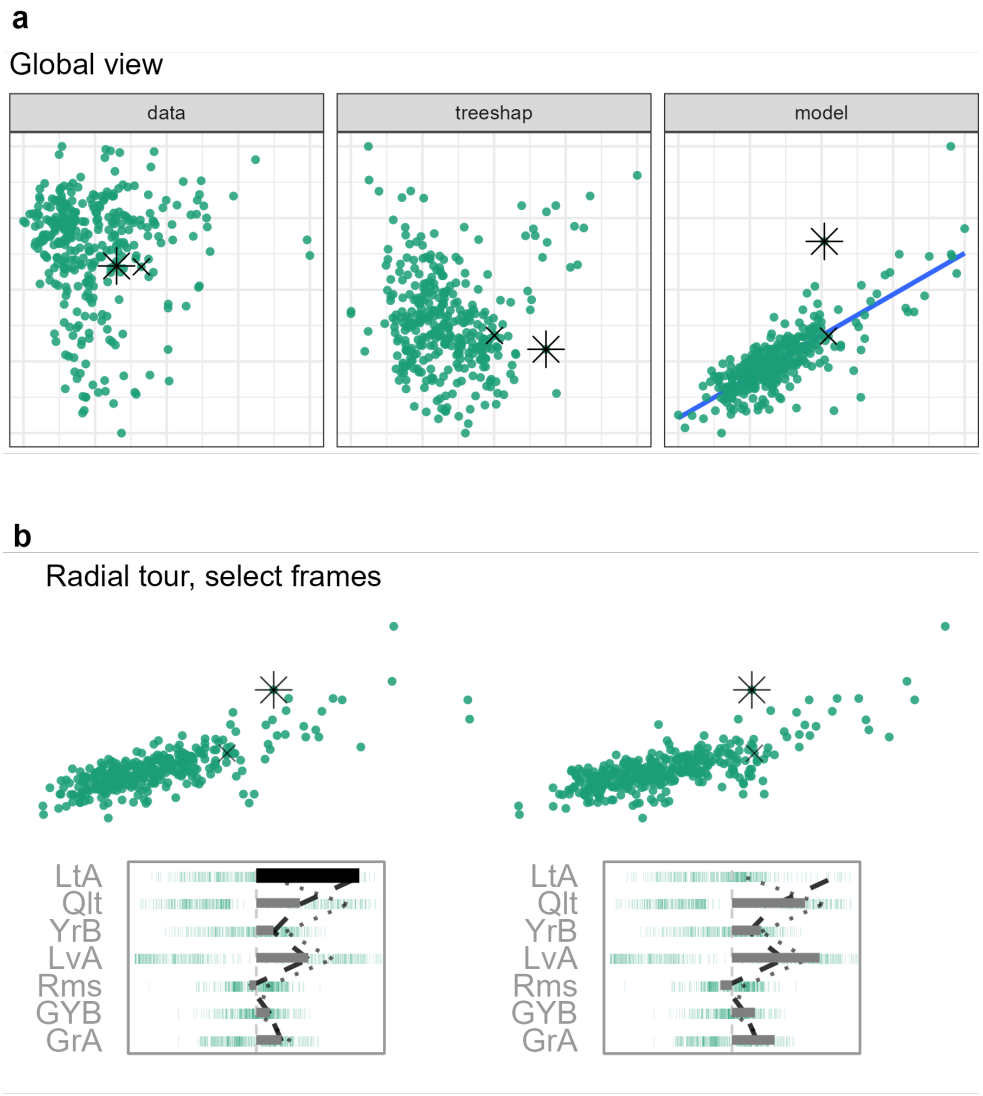

Figure 9. (ref:caseames-cap)

attribution to Lot Area (`LtA`), while the CI has minimal attribution to this feature. The PI has a higher predicted value than the CI in the attribution projection. Reducing the contribution of Lot Area brings these two prices in line. This suggests that if the model did not value Lot Area so highly for this observation, then the observed sales price would be quite similar. That is, the large residual is due to a lack of factoring in the Lot Area for the prediction of PI's sales price. (A video showing the animation is at `https://vimeo.com/666431134`.)

6.4. *Ames Housing 2018, Sales Price Regression*

Ames housing data 2018 (**??**) was subset to North Ames (the neighborhood with the most house sales). The remaining are 338 house sales. A random forest model was fit, predicting the sale price [USD] from the property features: Lot Area (`LtA`),

Overall Quality (`Qlt`), Year the house was Built (`YrB`), Living Area (`LvA`), number of Bathrooms (`Bth`), number of Bedrooms (`Bdr`), the total number of Rooms (`Rms`), Year the Garage was Built (`GYB`), and Garage Area (`GrA`). Using interactions with the global view, a house with an extreme negative residual and an accurate instance with a similar prediction is selected.

Figure @ref(fig:caseames) selects the house sale 74, a sizable under prediction with an enormous Lot Area contribution. The CI has a similar predicted price though the prediction was accurate and gives almost no attribution to lot size. The attribution projection places instances with high Living Areas to the right. The contribution of Living Area contrasts the contribution of this feature. As the contribution of Lot Area decreases, the predictive power decreases for the PI, while the CI remains stationary. This large importance in the Living Area is relatively uncommon. Boosting tree models may be more resilient to such an under-prediction as they would up-weighting this residual and force its inclusion in the final model.

7. Discussion

There is a clear need to extend the interpretability of black box models. This paper provides a technique that builds on local explanations to explore the feature importance local to an instance. The local explanations of an attribution projection from which feature contributions are varied using a radial tour. Several diagnostic plots are provided to assist with understanding the sensitivity of the prediction to particular features. A global view shows the data space, explanation space, and residual plot. The user can interactively select instances to compare, contrast, and study further. Then the radial tour is used to explore the feature sensitivity identified by the attribution projection.

This approach has been illustrated using four data examples of random forest models with the tree SHAP local explanation. Local explanations focus on the model fit, and help to dissect which variables are most responsible for the fitted value. They can also form the basis of learning how the model has got it wrong, with when the instance is misclassified or has a large residual.

In the penguins example, we showed how the misclassification of a penguin arose due to it having an unusually small flipper size compared to others of its species. This was verified by making a follow-up plot of the data. The chocolates example shows how a dark chocolate was misclassified primarily due to its attribution to Fiber, and a milk chocolate was misclassified as dark due to its lowish Sodium value. In the FIFA example, we show how low Messi's salary would be if it depended on defensive skills. In the Ames housing data, an inaccurate prediction for a house was likely due to the Lot Area is not being effectively used by the random forest model.

This analysis is manually intensive and thus only feasible for investigating a few instances. The recommended approach is to investigate an instance where the model has not predicted accurately and compare it with an instance with similar predictor values where model fitted well. The radial tour launches from the attribution projection to enable exploration of the sensitivity of the prediction to any feature. It can be helpful to make additional plots of the features and responses to cross-check interpretations made from the cheem viewer. This methodology provides an additional tool in the box for studying model fitting.

An implementation is provided in the open-source **R** package **cheem**, available on CRAN at `https://CRAN.R-project.org/package=cheem`. Example data sets are

provided, and you can upload your data after model fitting and computing the local explanations. In theory, this approach would work with any black box model, but the implementation currently only calculates tree SHAP for tree-based models supported by **treeshap** (tree based models from **gbm**, **lightgbm**, **randomForest**, **ranger**, or **xgboost ?????**, respectively). Tree SHAP was selected because of its computational efficiency. The SHAP and oscillation explanations could be added with the use of the `DALEX::explain()` and would be an excellent direction to extend the work (**??**).

Acknowledgments

Kim Marriott provided advice on many aspects of this work, especially on the explanations in the applications section. We would like to thank Professor Przemyslaw Biecek for his input early in the project and to the broader MI 2 lab group for the DALEX ecosystem of **R** and Python packages. This research was supported by the Australian Government Research Training Program (RTP) scholarships. Thanks to Jieyang Chong for helping proofread this article. The namesake, Cheem, refers to a fictional race of humanoid trees from Doctor Who lore. **DALEX** pulls on from that universe, and we initially apply tree SHAP explanations specific to tree-based models.