Classification Models for Statistical Graphics

June 2023

Abstract

This paper presents an interpretable approach to detecting patterns in scatter plots, which can help automate the process of checking statistical assumptions (e.g., linearity in residual plots for linear regression). Scatter plot diagnostics (scagnostics) are used to quantify aspects of a scatter plot's characteristics, such as shape and trend, on a scale from 0 to 1. Scatter plots with varying appearances were generated and their scagnostics were used as explanatory variables to train statistical learning models—logistic regression, generalized additive models (GAMs), and random forest models—to classify plots with distinct patterns. The accuracy levels for these models were promising, ranging from 97.4% to 99.9%, with performance similar to that of a convolutional neural network (CNN). While CNNs offered a comparable level of accuracy and robustness, the interpretability of the statistical models allows for a better understanding of their limitations. Further testing on the models' sensitivity to training data reveals that the models perform exceptionally well on distributions they have been trained on, but tend to struggle significantly on additional testing distributions.

1 Introduction

In statistical analysis, graphical representations are often used to summarize and investigate statistical models' fit. These representations include residual plots, histograms, boxplots, and scatterplots. The interpretation of these graphical representations can be subjective and prone to bias, especially when dealing with complex and high-dimensional datasets. To reduce subjectivity and improve the accuracy and consistency of data interpretation, classification models can be developed to detect trends.

In recent years, there has been growing interest in using statistical measures, such as scagnostics, to quantify the characteristics of plots and use them as predictors for classification models (Kromrey and Wongsuphasawat, 2019; Doiron and Ankerst, 2020; Guo and Agrawala, 2019). *Scagnostics*, short for scatter plot diagnostics, are relatively new statistical measures for plots that capture the shape, trend, coherence, and density of scatter plots. Introduced in Tukey and Tukey (1985) and later improved upon in Wilkinson et al. (2005) and Wilkinson and Wills (2008), these measures have been shown to be effective in identifying important patterns and relationships in the bivariate data. In this paper, we present three classification models for statistical graphics based on scagnostics measures: a logistic regression model, a generalized additive model and a random forest. We aim to demonstrate the effectiveness of using scagnostics to reduce the subjectivity of investigating the fit of statistical models. Specifically, we will use scagnostics measures as predictors for our classification models and compare the performance of our model with existing computer vision classification models such as convolutional neural networks (CNNs). We will also evaluate the interpretability of our classification models, investigating whether improved interpretability is an added benefit of using these statistical models vs. more complex classification methods such as CNNs.

2 Motivation

Before describing the scagnostics and classification models that we used in our analysis, we will introduce two motivating examples. In this paper, we sought to develop a framework for identifying patterns in scatterplots created using both data and residuals.

2.1 Identifying Patterns in Scatterplots

In Matejka and Fitzmaurice (2017), researchers developed a method to produce visually dissimilar graphs with identical statistical properties. An example of a collection of these graphs is depicted in Figure 1. The plots possess the same summary statistics (mean, standard deviation, correlation) but have notably different appearances.

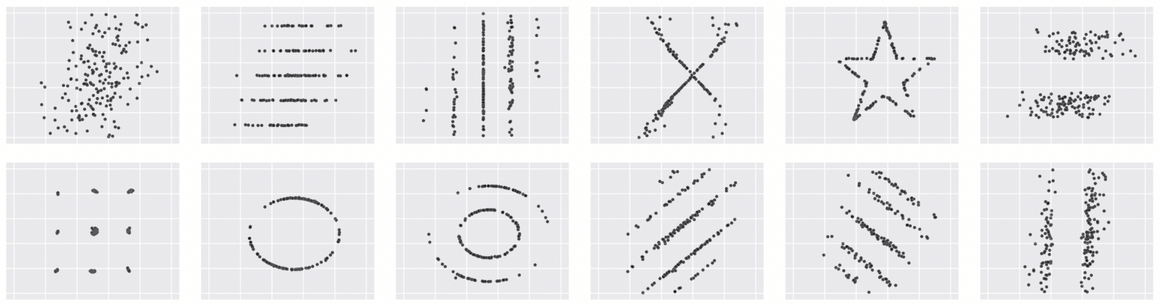


Figure 1: A collection of datasets that vary in appearance but have the same summary statistics to 2 decimal places ($\bar{x} = 54.02$, $\bar{y} = 48.09$, $sd_x = 14.52$, $sd_y = 24.79$, $r = +0.32$). Image originally published in Matejka and Fitzmaurice (2017).

This study displays the importance of interpreting graphs through visual observation. In many cases, it is not sufficient to analyze a dataset solely using summary statistics. Further, it introduces the need for additional metrics to differentiate between graphs with identical summary statistics. A series of *scagnostics* have been proposed as a way to summarize and differentiate between scatterplots with different patterns. In our paper, we propose the use of these *scagnostics* as variables in statistical learning models as a way to identify plots of a certain trends. Namely, we develop models to identify whether scatterplots possess a linear trend. These models can be used in place of summary statistics, which can be shared across visually disimilar plots, and visual observation, which can be subjective.

2.2 Identifying Patterns in Residual Plots

Residual plots are important diagnostic tools in assessing the appropriateness of a fitted model (Chihara and Hesterberg, 2018). If a model is appropriate for a data set, points in the residual plot should be scattered randomly about $e = 0$. If this is not the case, patterns in the residual plot help us identify shortcomings in our model that prevent us from fully capturing trends in the data. Additionally, residual plots can help identify outliers in the data.

Figure 2 shows the residuals of four fitted linear regression models. The points in Figure 2(a) are randomly scattered about $y = 0$, suggesting that the straight-line model is appropriate for the data. Most of the points in Figure 2(b) are below $y = 0$, suggesting that the regression line is overestimating y values. Figure 2(c) shows an outlier that hasn't been accounted for by the model, depicted as a shaded point near the top of the pannel. Figure 2(d) shows curvature in the residuals, suggesting that a straight-line model may not be appropriate for the data. In other words, there may not be a linear relationships between the two variables.

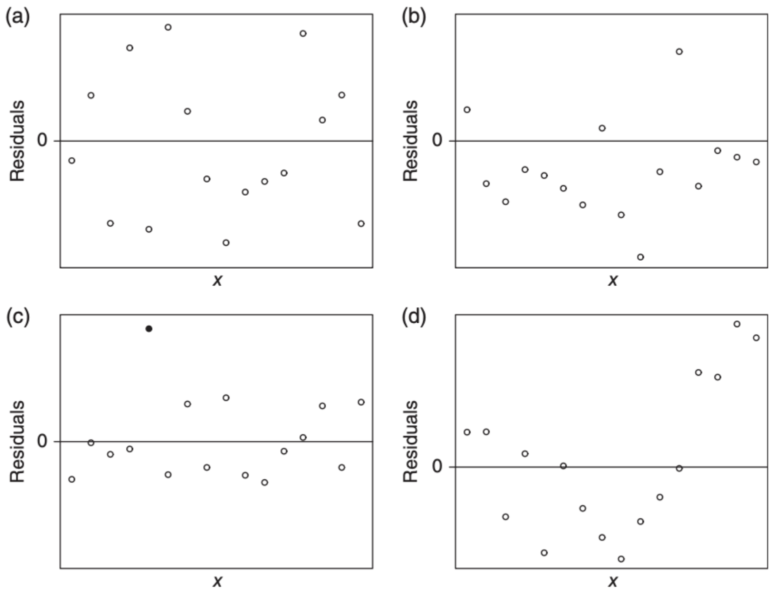


Figure 2: Examples of residual plots. (a) Residuals from a well-fitted model. (b) Residuals from a model that overestimates y values. (c) Residuals that include an outlier, depicted by the shaded point near the top of the plot. (d) Residuals from a model that is not appropriate for the relationship between the two variables in the data set. Image originally published in Chihara and Hesterberg (2018).

While some features of residual plots are easy to read, some statisticians may be unable to detect more subtle violations. The subjectivity involved with interpreting diagnostic plots introduces a potential need for an automated method to check models. An automated method using visual features may present benefits over individual tests for curvature or non-constant variance in the form of higher levels of interpretability. Features that people can observe may be more easily understood when compared to statistical tests for significance.

In the following sections, we propose a classification model trained on scagnostics as a way to differentiate between scatterplots. We show how various classification models can be used to identify problematic residual plots. These findings may be extended to additional types of diagnostic plots such as Q-Q and leverage plots.

3 Data Generation

Data is crucial for our work as it drives the learning process of our statistical models, allowing them to classify patterns accurately. Due to its high level of importance, we prioritized the data generation process to ensure robustness and high accuracy. We generated various scatter plots which included plots of both linear and non linear distributions, and computed their scagnostics. Using the cassowaryr package, we calculated the underlying structures of the scagnostic features of each plot and stored the scagnostic values as a data frame (Jean-Romain Roussel, 2018).

In this section, we introduce how we created the training and testing data sets and introduce the additional testing set for verifying the robustness of our predictions. This additional testing set contained new data not included in the training set, enabling us to identify model performance on completely new distributions. We also briefly outline the data generation process we used to train and test our CNN models.

3.1 Training and Testing sets

For each scatter plot i, we can represent all n_i points in the plot by (X_i, Y_i) where $X_i, Y_i \in \mathbb{R}^{n_i}$. That is, point j in plot i is located at $(X_i[j], Y_i[j])$. Moreover, we draw the number of points n_i uniformly from a set of all integers between 40 and 200. This procedure allows us to have both sparse and dense scatter plots.

For each scatter plot i, we generated our scatter plots of (X_i, Y_i) in two different ways: constructing a scatter plot from the distributions of X_i and Y_i either independently or jointly.

To pick the distributions of X_i and Y_i independently, we first randomly select a distribution of X_i from one of three distributions:

- **Uniform:** $X_i \overset{i.i.d.}{\sim} \text{Uniform}([-20, 0]) + S_i$.
- **Disjoint Uniform:** $X_i \overset{i.i.d.}{\sim} \text{Uniform}([-20, -12] \cup [-8, 0]) + S_i$.
- **Discrete:** $X_i \overset{i.i.d.}{\sim} \text{Uniform}(\{-10, -9, \cdots, -1, 0\}) + S_i$.

where S_i (which is called a distribution shift) is drawn from Uniform($[5, 15]$). This additional parameter S_i allows our plots to be in various shapes. In Figure 3b, for instance, S_i is equal to 11.1, resulting in an asymmetric parabolic plot. We also randomly pick a hyperparameter a_i where $a_i \sim \text{Uniform}([-4, 4])$ allowing for plots with various shapes (e.g., a scatter plot with linear trend $Y_i = a_i X_i + \epsilon_i$ is steeper when $a_i = 4$ and less steep when $a_i = 0.1$). We also randomly select a variance for the error terms, ϵ_i, from Uniform([1, 10]), and then randomly generate an n_i-dimensional vector of errors from $N(0, \sigma_i^2)$. After having sampled a_i, σ_i^2, and X_i, we constructed Y_i from these choices of distributions:

- **Linear:** $Y_i = a_i X_i + \epsilon_i$.

- **Parabolic:** $Y_i = a_i X_i^2 + \epsilon_i$.
- **Sine:** $Y_i = \sin(a_i X_i) + \epsilon_i$.
- **Piece-wise Linear:** $Y_i[1 : \frac{n_i}{2}] = a_i X_i[1 : \frac{n_i}{2}] + \epsilon_i[1 : \frac{n_i}{2}]$ and $Y_i[\frac{n_i}{2} + 1 : n_i] = a_i X_i[\frac{n_i}{2} + 1 : n_i] + \epsilon_i[\frac{n_i}{2} + 1 : n_i] + 10(|a_i| + 1)$ where X_i is sorted in an ascending order. Intuitively speaking, the first and second halves of this scatter plot are linear trends with a gap of size $10(|a| + 1)$ in the middle.
- **Exponential-Growth:** $Y = \exp(|a_i| X_i) + \epsilon_i$.
- **Exponential-Decay:** $Y = \exp(-|a_i| X_i) + \epsilon_i$.
- **Logarithmic growth:** $Y = \log(|a_i| X_i) + \epsilon_i$.

The combinations of these choices allow us to obtain data in various patterns. For instance, a large σ_i and a small a_i with $X_i's$ disjoint uniform distribution can result in a wide and flat linear trend as shown in Figure 3a.

Besides picking the distributions of X_i and Y_i independently, we can also pick them jointly as follows:

- **Funnel-shaped:** multivariate log normal with $\rho_i = 0.9$.
- **Donut-shaped:** $(x_i[j], y_i[j]) = (r_j \cos(\theta_j), r_j \sin(\theta_j))$ where r_j is from Uniform$([a_i, a_i + c])$ for some positive constant c.

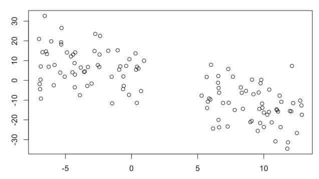
Note that some distributions for Y (e.g., Funnel) were inspired by Pickens (2019). Also, we re-scaled some of these plots to avoid computational issues. For instance, when a_i is large, the noise in the exponential growth function is too tiny, and some scagnostic variables could not be calculated.

To train and test our models, we generated a total of 33,188 scatter plots. Of these plots, nearly half were linear, while the remaining eight nonlinear patterns were evenly distributed in terms of their counts. For each scatter plot i, we then calculated its scagnostic variables from (X_i, Y_i), and we stored this information in one data frame as shown in Figure 4.

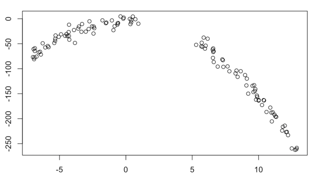
3.2 Out-of-distribution Testing sets

To test whether our models were robust to unknown data sets, we introduced additional out-of-distribution data sets, which we used to test the robustness of our models—whether they can classify patterns they had never seen in the training sets. Besides linear patterns, we introduced six new patterns that were not contained in our training sets, as illustrated in Figure 5. We generated $1,822$ samples in total.

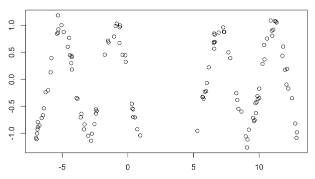
- **Cubic:** $Y_i = a_i X_i^3 + \epsilon_i$.
- **V-shaped (absolute value function):** $Y_i = |a_i X_i| + \epsilon_i$.
- **Rotated-Sine:** $Y_i = \sin(a_i X_i) + \epsilon_i$ then with θ_i degree rotated. Here, $\theta_i \sim$ Uniform$([-\pi, \pi])$.
- **Reciprocal:** $Y_i = 1/(a_i X_i) + \epsilon_i$.
- **Hyperbolic-Tan (tanh):** $Y_i = \tanh(a_i X_i) + \epsilon_i$.
- **Hyperbolic-Sec (sech):** $Y_i = \mathrm{sech}(a_i X_i) + \epsilon_i$.



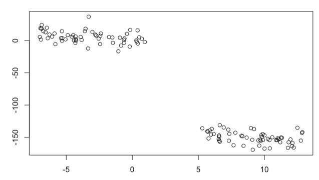
(a) Linear plot



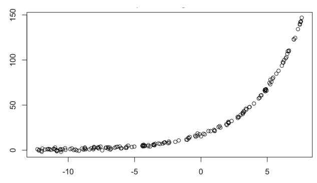
(b) Parabolic plot



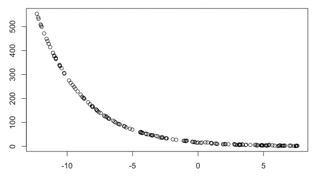
(c) Sine-shaped plot



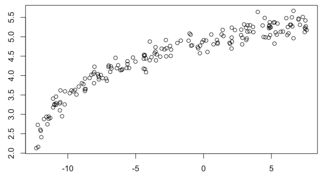
(d) Piecewise-linear



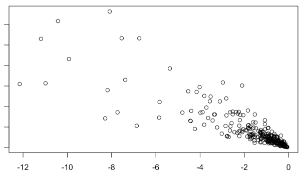
(e) Exponential-growth



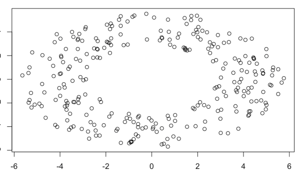
(f) Exponential-decay



(g) Logarithmic plot



(h) Funnel plot



(i) Donut plot

Figure 3: Nine distributions we used in both training and testing sets. (a)−(c) are drawn when X is from the disjoint uniform distribution. (e)−(g) are drawn when the $X's$ distributions are from a uniform distribution. (h) and (i) are drawn by sampling from a joint distribution for X and Y.

3.3 Data sets for the CNN models

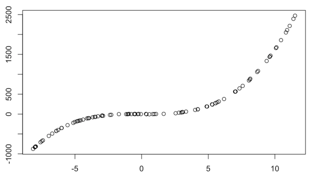
To train CNN models, we had to export the scatter plots as pictures. For the sake of computational power, we converted all data sets from the previous two subsections into 60×40 PNG format. Figure 6 shows the sample outputs of when we converted linear, parabolic, and V-shaped into PNG files.

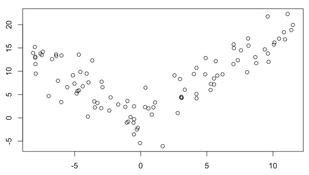
4 Graph-Theoretic Scagnostics

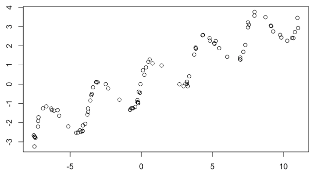
Scagnostics measure aspects of scatterplots that are often overlooked by traditional summary statistics. Specifically, scagnostics aim to measure the outliers, shape, trend, density, and coherence of scatterplots. They were originally introduced in Tukey and Tukey (1985) and later improved in Wilkinson et al. (2005). We use these scagnostics as predictors for our models and describe each

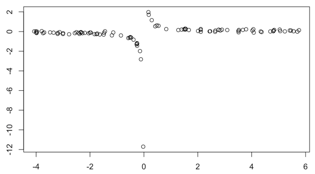
plot	outlying	stringy	striated	clumpy	sparse	skewed	convex	skinny	monotonic	splines	linear
1	0.000	0.694	0.219	0.794	0.048	0.576	0.670	0.518	0.970	0.948	TRUE
2	0.063	0.684	0.237	0.772	0.043	0.514	0.648	0.557	0.961	0.939	TRUE
3	0.037	0.740	0.259	0.817	0.043	0.545	0.682	0.540	0.965	0.941	TRUE
4	0.089	0.748	0.212	0.895	0.054	0.556	0.708	0.415	0.910	0.839	TRUE
5	0.078	1.000	0.785	0.960	0.023	0.786	0.001	0.936	0.999	1.000	FALSE
6	0.035	0.697	0.215	0.960	0.053	0.512	0.389	0.567	0.480	0.808	FALSE

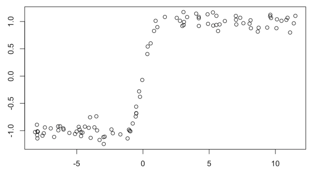
Figure 4: The training set formatted as a CSV file.


(a) Cubic


(b) V-shaped


(c) Rotated-Sine


(d) Reciprocal


(e) Tanh

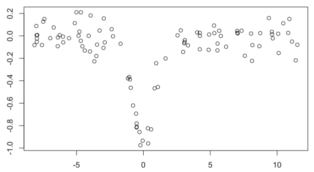
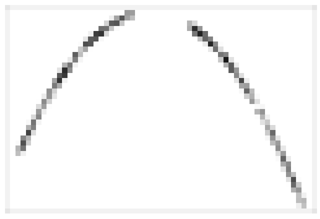

(f) Sech

Figure 5: Six patterns in the out-of-distribution testing set.

(a) Linear


(b) Parabolic

(c) V-shaped

Figure 6: 60×40-PNG data sets.

scagnostic below.

4.1 Geometric Graphs

First, it is important to introduce the idea of graphs and their corresponding notation. A graph, $G = \{V, E\}$, is a collection of a set of vertices, V, and edges, E. Scagnostics are based upon alpha hulls, convex hulls, and minimum spanning trees (MST) as seen in Figure 7. As we step through these geometric graphs, we will be using the *iris* dataset (Fisher, 1936) as an illustrative example of how these various components look on a real-world example.

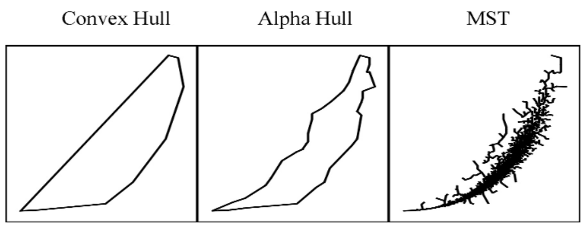


Figure 7: Depiction of the graphs that scagnostic measurements introduced in Wilkinson et al. (2005).

4.1.1 Scatter Plot

We start with a scatterplot, a traditional representation of the data. Scatterplots display data in the Cartesian plane, with points denoted in their respective x and y values. It's not necessary for this plot to be constructed from a traditional set of two-dimensional data pairs (such as in the case of a residual plot), but it's this *2D representation* that we focus on in this paper. We can see the scatter plot showing the relationship between Petal Length(x) and Petal Width(y) in Figure 8.

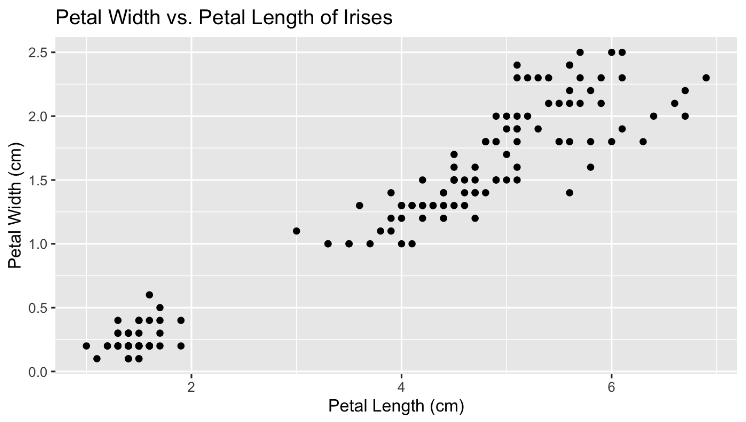


Figure 8: Scatter plot created by the Iris dataset.

From this 2D plot, we can create the three graphs mentioned previously to calculate the scagnostics.

4.1.2 Convex Hull

A set of points, X, in $\mathbb{R}^2$ is *convex* if it contains all the straight line segments connecting any pair of its points. The *convex hull* of X is the boundary of the intersection of all convex sets containing X. One can think of a convex hull as the result of "stretching a rubber band" over the set of data points. Then, we allow the rubber band to "snap" around the graph's points. This leaves us with a shell that covers only the linear paths between the furthest outward points in our dataset. If we place one of these bands around a scatterplot of points, we end up with a convex hull. A convex hull on the Iris dataset can be seen in Figure 9.

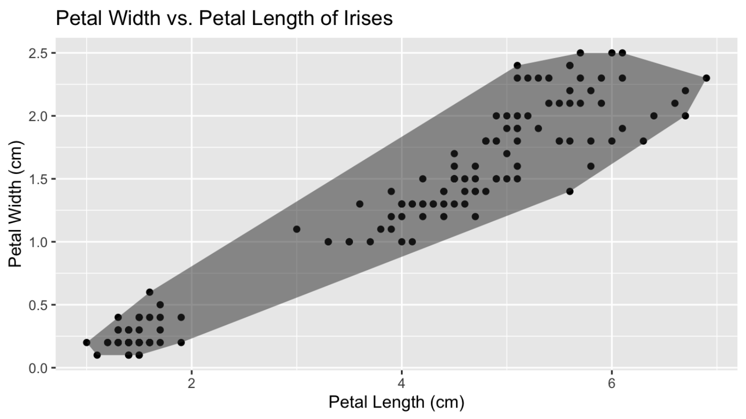
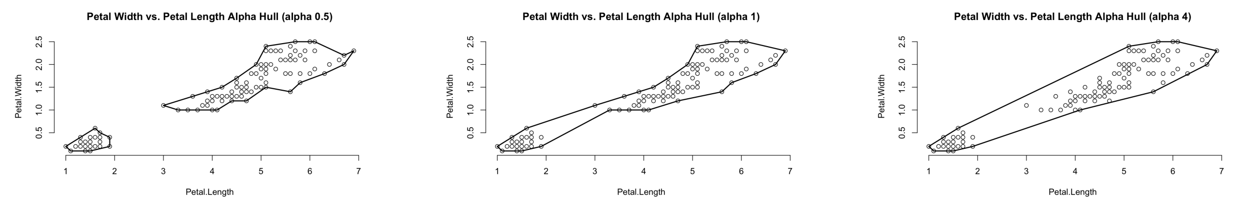


Figure 9: Convex hull over the Iris Scatter Plot from Figure 8.



(a) Alpha Hull with $\alpha = 0.5$ (b) Alpha Hull with $\alpha = 1$ (c) Alpha Hull with $\alpha = 4$

Figure 10: Alpha hulls of varying α values.

4.1.3 Alpha Hull

The alpha hull is a more precise outline than that of the convex hull. By selecting an α, we can effectively circle the outside of the points to select the points through which we need to construct our alpha hull. We can imagine this as a disk with a radius of α effectively turning around the outside of our scatter plot. As we roll the circle around the outside of the set of points, if we ever find that two points lie on the perimeter of the circle while no points lie inside the circle, the alpha hull should connect the two points for the given α level. When the alpha value is small enough for data that falls into various "clumps," there can be two discrete sections in an alpha hull. An alpha hull with various alpha cutoffs is shown in Figure 10.

An added complexity that comes with using the alpha hull is deciding on an appropriate α value. As α increases, the alpha hull begins to more closely resemble the convex hull.

4.1.4 Minimum Spanning Tree

A *tree* is an acyclic, connected, simple graph. A *spanning tree* is an undirected graph whose edges are structured as a tree. A *minimum spanning tree (MST)* is a spanning tree whose total length (sum of edge weights) is the least of all spanning trees on a given set of points.

A spanning tree is a set of edges connecting to all the graph points. For a tree to be a minimum spanning tree, it must have the shortest length of total edges possible, given the structure of the graph nodes. The "acyclic" part of the definition requires that there not exist any *cycles* in the graph. A cycle is a path in a graph in which there are some number of vertices connected in a closed chain. "Simple" means that there can only be one edge between any two vertices, and no vertex may have an edge that loops back to itself. Finally, "connected" means that all edges in the graph must be connected to all other edges.

The minimum spanning tree for the Iris dataset is shown in Figure 11.

4.2 Measures

Scagnostics are focused on assessing five characteristics of scatterplots using underlying trends: *outliers, shape, trend, density,* and *coherence* (Wilkinson et al., 2005). In the following sub sections, we discuss each of these five scatterplot characteristics along with their associated scagnostics.

4.2.1 Outliers

Outliers can be an important characteristic for determining the trend of a dataset. Outliers are observations that lie an abnormal distance from other values in a given dataset. In order to determine which values are unusual enough to be considered outliers, Wilkinson et al. (2005) introduced *peeling* the MST. We consider an outlier to be any node with degree 1 and associated edge weight greater than ω (Wilkinson et al., 2005).

Following Tukey (1977), we set ω as

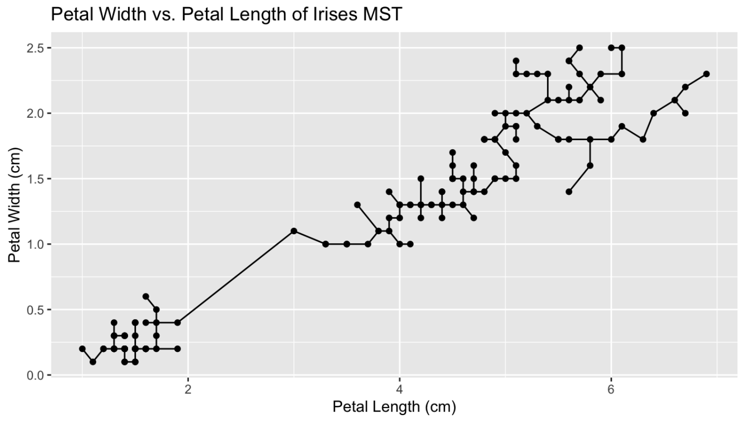


Figure 11: Minimum Spanning Tree over the Iris Scatter Plot from Figure 11.

$$\omega = q_{75} + 1.5(q_{75} - q_{25}) \tag{1}$$

where q_{75} is the 75th percentile of the MST edge lengths and q_{25} is the 75th percentile of the MST edge lengths. The expression in parentheses is the *interquartile range* of the edge lengths of the MST.

- **Outlying.** The outlying measure is the proportion of the MST excluding the outlier(s) and including the outlier(s) (Wilkinson et al., 2005). This scagnostic effectively measures the "outliers characteristic" of scatterplots (Wilkinson and Wills, 2008).

$$c_{outlying} = \frac{length(T_{outliers})}{length(T)} \tag{2}$$

4.2.2 Shape

The scagnostics described in Wilkinson et al. (2005) also aim to measure both the topological and geometric aspects of the shape of a scatterplot. In the expressions below, H denotes the convex hull, A denotes the alpha hull, and T denotes the minimum spanning trees. Outliers are ignored in our shape calculations.

- **Convex.** The convex scagnostic measures the degree to which the points in a scatterplot curve outwards (Wickham and Stryjewski, 2011). The ratio between the area of the alpha hull and the area of the convex hull ultimately provides a value that effectively describes the convexity of a scatterplot (Wilkinson and Wills, 2008). An illustration of the alpha hull overlayed onto the convex hull can be seen in Figure 12.

$$c_{convex} = \frac{area(A)}{area(H)} \tag{3}$$

- **Skinny.** The ratio of the perimeter and the area of a given polygon roughly measures how skinny it is. The skinny measurement specified in Wilkinson et al. (2005) is a corrected and normalized ratio between the perimeter and area of the alpha hull. The ratio is normalized so that a circle has a skinny value of 0, a square has a value of 0.12 and a skinny polygon has a value near 1.

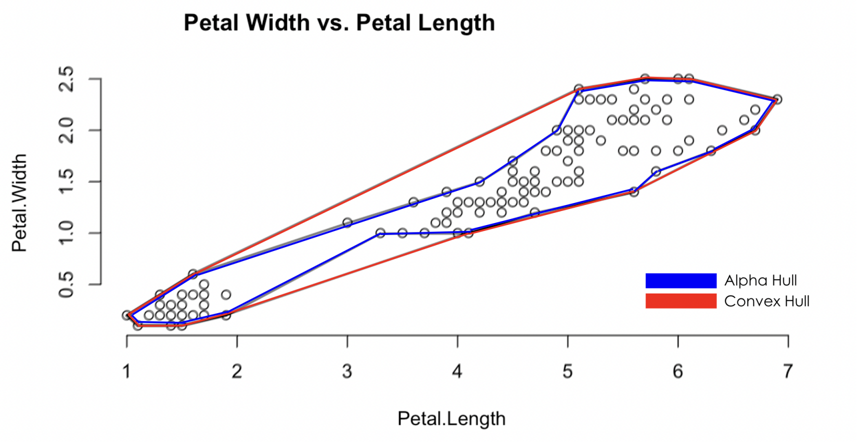


Figure 12: Depiction of the alpha hull overlayed with a convex hull on the Iris dataset. The alpha hull is highlighted in blue and the convex hull is highlighted in red. The convex scagnostic is calculated by taking the proportion of the area of the alpha hull to the area of the convex hull.

$$c_{skinny} = 1 - \frac{\sqrt{4\pi area(A)}}{perimeter(A)} \tag{4}$$

- **Stringy.** A stringy shape is a skinny shape with no branches. This is the ratio between the diameter of a minimum spanning tree and the total length of the minimum spanning tree.

$$c_{stringy} = \frac{diameter(T)}{length(T)} \tag{5}$$

- **Straight.** This measure is the Euclidean distance between the points at the ends of the diameter of the MST divided by the diameter itself. A straight graph should have a value of 1, while other graphs will have smaller values. In the expression below, t_j and t_k denote the vertices in the MST on which the diameter is defined.

$$c_{straight} = \frac{dist(t_j, t_k)}{diameter(T)} \tag{6}$$

4.2.3 Trend

We are focused on determining the monotonicity of a scatterplot. As specified in Wilkinson et al. (2005), the monotonicity of scatterplots may be indicative of whether a scatterplot has a linear distribution or not.

- **Monotonic.** The *Spearman correlation coefficient* is a Pearson correlation on the ranks of x and y and is corrected for ties. The coefficient is squared to accentuate large values and remove the distinction between negative and positive relationships. It is assumed that users are most interested in strong relationships whether they are negative or positive.

$$c_{monotonic} = r^2_{spearman} \tag{7}$$

4.2.4 Density

These indices can be used to detect different distributions of points within a scatterplot.

- **Skewed.** The distribution of edge lengths of a MST sheds light on the density of points in a scatterplot. This measure is a ratio that utilizes quantiles of edge lengths in the MST. Intuitively, the skewed scagnostic measures the relative difference between the largest edges and the most average edges of an MST where q_{90} is the 90th percentile edge based on edge length, and similarly q_{10} is the 10th percentile edge. This statistic is relatively robust to outliers (Wilkinson et al., 2005).

$$c_{skew} = (q_{90} - q_{50})/(q_{90} - q_{10}) \tag{8}$$

- **Clumpy.** To calculate the clumpiness of a dataset, we turn to the RUNT statistic specified in Hartigan and Mohanty (1992) that is based off of the MST of the dataset. This statistic is most easily understood in terms of the single-linkage hierarchical clustering tree called a *dendrogram*. The nodes in the dendrogram seen in Figure 13 are used to calculate the runt size statistic. The runt size of a dendrogram node is the smaller of the number of leaves of each of the two sub-trees joined at that node. Since there is an isomorphism between a dendrogram and the MST as specified in Gower and Ross (1969), we can associate a runt size (r_j) with each edge (e_j) in the MST, as described by Stuetzle (2003). The equation used to calculate the clumpy scagnostic is

$$c_{clumpy}(T) = \max_j [1 - \max_k [length(e_k)]/length(e_j)] \tag{9}$$

where j indexes edges in the MST and k indexes edges in each runt set derived from an edge indexed by j. An illustration of calculating the clumpy scagnostic can be seen in Figure 14. We see that the algorithm finds the edge that maximizes the overall value (the longest edge), then removes it to create two disjoint trees, and finally finds the length of the longest edge in the runt set.

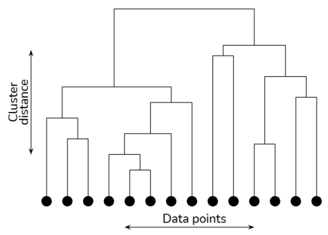


Figure 13: Example of a dendrogram.

It is worth mentioning that in Wilkinson et al. (2005) and Wilkinson and Wills (2008), the definition of the runt graph, R_j, is incorrect. They indicate that after selecting an edge, e_j, from the MST, R_j is the smaller of the two MSTs after removing e_j and all edges **smaller** than $length(e_j)$. However, this results in the clumpy scagnostic always returning a value of zero. Instead, the R_j is the smaller of the two MSTs after removing e_j and all edges **larger** than $length(e_j)$.

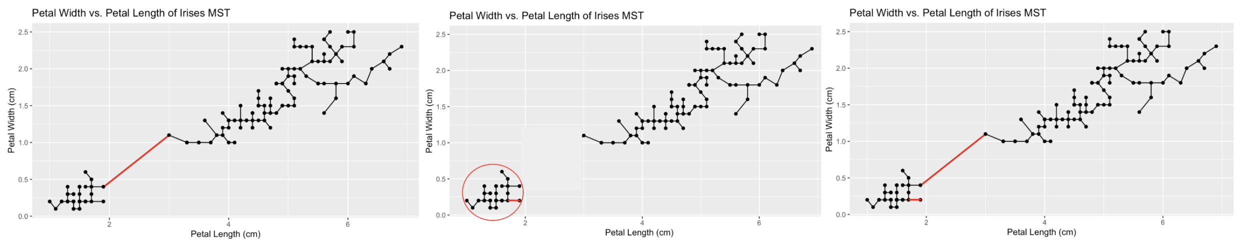


Figure 14: First find the edge that maximizes the the equation. Then remove it along with all edges greater in length, and find the longest edge in the runt set. Finally, divide the edges to get the edge proportion.

4.2.5 Coherence

We define coherence in a set of points as the presence of relatively smooth paths in the minimum spanning tree (Wilkinson et al., 2005).

- **Striated.** All of the adjacent edges in the MST on these points are collinear. The striated scagnostic sums angles over all adjacent edges with the intention of measuring the average angle between edges in an MST. Suppose $V^{(2)} \subseteq V$ is the set of all vertices of degree 2 in V. Then

$$c_{striated}(T) = \frac{1}{|V^{(2)}|} \sum_{v \in V^{(2)}} |cos\theta_{e(v,a)e(v,b)}|. \qquad (10)$$

4.3 Examples.

In Figure 15, we show examples of scatterplots accompanied by their calculated scagnostics. There are 11 different scatterplots represented as each of the columns with a picture of each one above each column. Each row represents a scagnostic. The cells are colored as blue representing a low value and red representing a high value (on a scale of 0-1).

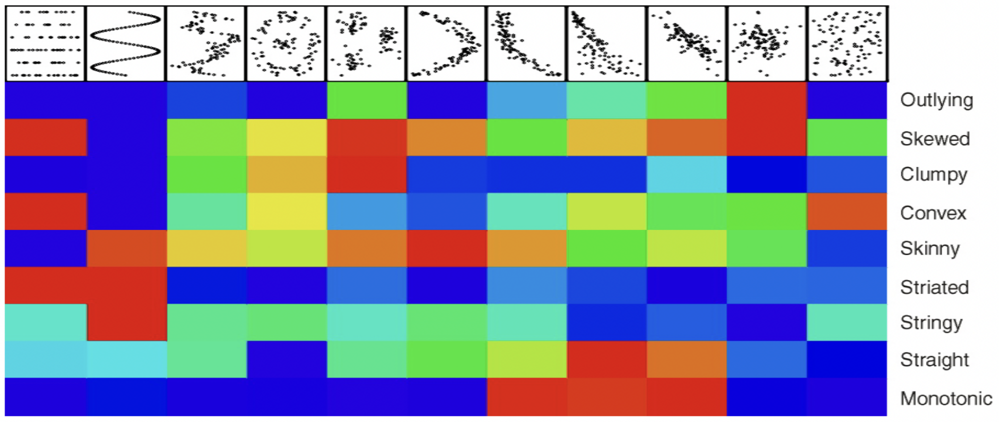


Figure 15: Examples of each scagnostic where blue represents low values and red represents high values. Image originally published in Wilkinson et al. (2005).

Each of these scatterplots correspond with different combinations of scagnostic values. We hope Figure 15 provides more intuition of what each scagnostic measures.

5 Models

After generating our training and testing data, we assessed how accurately statistical learning models could classify graphs as linear or nonlinear. We compared the performance of three statistical models using the 10 scagnostics as predictors in our models.

5.1 Logistic Regression

Each plot in our dataset was classified using a binary response variable Y_i, where $Y_i = 1$ for linear plots and $Y_i = 0$ for non-linear plots. We assume that the Y_i's come from a Bernoulli distribution where $P(Y_i = 1) = \pi_i$. Given this assumption, the first model we used was a standard logistic regression model.

Logistic regression models take the form below (James et al., 2013).

$$\log(\frac{\pi_i}{1-\pi_i}) = \beta_0 + \beta_1 x_{1i} + \beta_2 x_{2i} + \cdots + \beta_{10} x_{10i} \tag{11}$$

The left-hand side is called the *log-odds* or *logit*, where π_i represents the probability that plot i is linear. We applied logistic regression to classify scatterplots using scagnostics as predictors. In the context of the equation above, x_{1i} through x_{10i} represent each of the 10 scagnostic values for plot i, β_0 is the y-intercept, and β_1 through β_{10} are the coefficients for each of the 10 predictors. We used the "glm" engine (Fox, 2015) for our implementation due to its superior flexibility and ability to handle large data sets (Geyer and Meeden, 1993). The "glm" engine utilizes a maximum likelihood-based approach, where coefficients are selected to maximize the function below.

$$\Pr(Y_1 = y_1, ..., Y_n = y_n) = \prod_{i=1}^{n} \pi_i^{y_i}(1-\pi_i)^{1-y_i} \tag{12}$$

5.2 Generalized Additive Models

Generalized additive models (GAMs) are a flexible extension of linear regression models that allow for nonlinear relationships between response and predictor variables (James et al., 2013). GAMs may allow us to explain more variability in the data than can be explained using a line. As shown below, the β coefficients from a logistic regression model are replaced with spline functions, denoted f_1 through f_{10}.

$$\log(\frac{\pi_i}{1-\pi_i}) = \beta_0 + f_1(x_{1i}) + f_2(x_{2i}) + \cdots + f_{10}(x_{10i}) \tag{13}$$

A spline function is a piecewise polynomial function that is defined by a number of knots that are placed throughout the range of the data. Between each knot, a polynomial function is fit to the data. Splines can include polynomials of any degree, under the constraint that a degree-d spline must be continuous in derivatives up to degree $d - 1$. An example of a cubic smoothing spline is shown in Figure 16.

For our final implementation of the GAM, we fit cubic smoothing splines to each of our 10 predictors. We utilized the "mgcv" package, which selected the number of knots that were used for each spline (Feng and Sang, 2021). Using cubic splines required our splines to be continuous at their first and second derivatives. An example of a cubic smoothing spline fit to logistic data is shown in Figure 17.

5.3 Random Forests

A decision tree is a method used for both regression and classification (James et al., 2013). A classification tree relies on quantitative and/or qualitative explanatory variables to predict a binary response. Classification tree nodes are chosen such that the classification error rate is minimized.

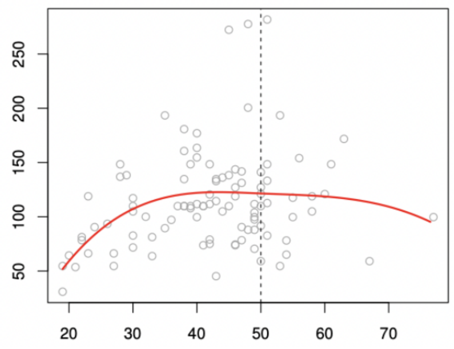


Figure 16: An example of a spline function (one knot) fit to a scatterplot created from two quantitative variables. A cubic polynomial is fit to the data on either side of the knot (dashed line) (James et al., 2013).

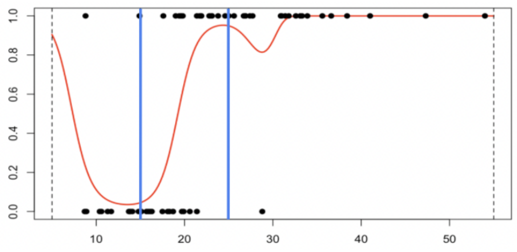


Figure 17: An example of a spline function (two knots) fit to data with a binary response variable and quantitative predictor variable. A cubic polynomial is fit to the data on either side of the two knots (blue lines) (Dowle, 2018).

An example of a classification tree is shown in Figure 18. Two of our predictor variables are used to split nodes. Plots with convexity values less than 0.5 will continue to the left side of the tree, while plots with convexity values greater than 0.5 will be classified as linear.

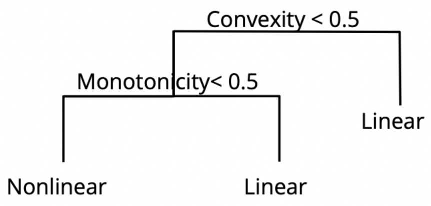


Figure 18: Example of a decision tree that uses scagnostics to classify a graph as linear or nonlinear.

Random forest classifiers consist of an ensemble of decision trees. Due to the high variance of decision trees, random forests rely on a concept called bagging. In the classification setting, we construct a large number of decision trees using bootstrapped training sets (sampling with replacement). For each decision tree, we use a randomly selected subset of the predictor variables available in our dataset. Usually, the size of the subset of predictor variables that we use is the square root of the total number of predictors in the dataset. Then, once we have created these decision trees, we conduct a majority vote. In other words, for a given observation, the class that is predicted by the majority of the bootstrapped trees is the class that we predict for that observation. This method reduces the variance introduced by decision trees. A visual of this process is shown in Figure 19.

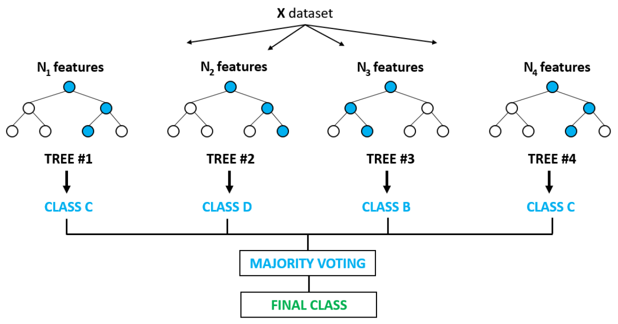


Figure 19: Depiction of a random forest algorithm. Image originally published in Chauhan (2021).

To optimize the performance of our random forest model, we tuned the model hyperparameters by testing several values. These included the number of trees (900, 1000, 1100), the maximum depth of the trees (2, 3, 4), and the minimum number of samples required to split a node (10, 15, 20). We used 5-fold cross validation to select the combination of hyperparameters that gave the best performance.

The most accurate model had a maximum depth of 4, 900 trees, and a minimum number of samples to split of 15. Thus, we implemented our final model using these hyperparameter values.

5.4 Convolutional Neural Networks

A convolutional neural network (CNN) is a Deep Learning algorithm that falls under the class of computer vision (Krizhevsky et al., 2012a,b). It uses information about the graphical constitution of various pixels and then passes that information through a number of layers to come to some supervised conclusion about the class of image that is being input. The process involves placing "weights" to different parts of the image to determine what is important when determining what class an image falls into. A diagram of a generic CNN can be seen in Figure 20.

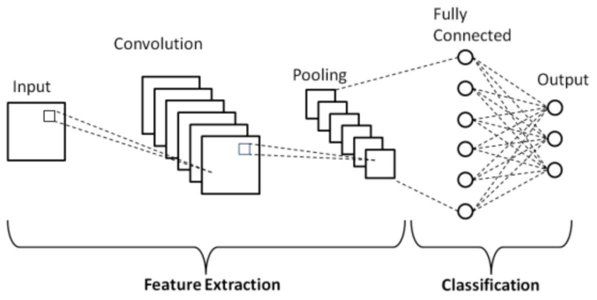


Figure 20: Diagram of a Generic Convolutional Neural Network (Phung and Rhee, 2019).

A neural network starts by looking at a grid of pixels on an image that comes into a first "hidden layer", or the convolution layer in Figure 20. From there, a series of calculations are run placing weights on the various parts of the image to determine their importance to the overall classification. At each layer, an equation is used to modify the variables. There are several equations to choose from, but in our model we used the rectified linear activation function. This brings negative predictions from previous layers. Various options can lead to different results, although the ReLu function is generally considered a safe default (He et al., 2015).

At a high level, a convolutional neural net is looking for "features" in the images in the same way that our scagnostics describe some sort of visual feature. The CNN is creating these from scratch by pooling the pixels into groups as the model passes through successive layers. This makes the model potentially more robust but also impacts our ability to interpret its results.

The exact construction of our model is three layers deep before being sent to a fully connected layer whose job is to decide on a final classification for our graph. Each layer in our CNN uses the ReLu activation function, which brings any weights of negative value up to a floor of 0.

Our model is trained on images from our datasets scaled down to 60x100 pixels. This removes much of our graphs' graphical clarity but is required to make the model computationally feasible. It's possible this reduction in resolution could impact our model's ability to classify certain types of graphs accurately, but the graph was still visually recognizable in the reduced form.

Thus far in the paper, we have been discussing use cases for scagnostics as a tool to predict the classification of plots. The intent of using a CNN for the same purpose is primarily an exercise of comparison. The primary goal of the research and modeling was to create an interpretable model that allows us to use quantitative visual metrics to describe whether or not particular graphs had specific characteristics.

6 Model Results

On the distributions that the models were trained on, logistic regression achieved an accuracy of 97.4%, GAMs achieved an accuracy of 98.3%, random forest achieved an accuracy of 99.9%, and SVMs achieved an accuracy of 99.5%. We can see the results in Table 1.

Distribution	**Generalized Additive Model**	**Random Forest**	**Logistic Regression**
Accuracy	99.8%	99.5%	97.3%
Specificity	99.7%	99.4%	97.2%
Sensitivity	99.9%	99.5%	97.5%

Table 1: Metrics for all models on Original Testing Distributions.

In addition to the training distributions, the accuracy of the models was evaluated on the additional testing distributions for robustness (as described in the Datasets section). The logistic regression model achieved an accuracy of 84.3% on the additional testing distributions, while GAMs achieved an accuracy of 88.5% and random forest achieved an accuracy of 83.4%. We can see the results in Table 2.

Distribution	**Generalized Additive Model**	**Random Forest**	**Logistic Regression**
Accuracy	88.5%	83.6%	84.3%
Specificity	86.3%	82.2%	82.7%
Sensitivity	94.2%	91.8%	94.1%

Table 2: Metrics for all models on Additional Testing Distributions.

Accuracy measures for each training distribution were also calculated for each model as shown in Table 3. Notice the cell highlighted in red (logistic regression result for the logarithmic growth distribution). The accuracy value in this cell is significantly lower than all other accuracies indicating that the logistic regression model does struggles to correctly classify logarithmic growth distributions as non linear.

The accuracy results for each additional testing distribution was also recorded. See Table 4. Notice the three distribution rows highlighted in red (absolute value, linear near 0 slope, rotated

Distribution	Logistic Regression	Generalized Additive Model	Random Forest
Circle	98.3%	99.8%	100%
Disconnected Linear	99.9%	99.8%	100%
Donut	100%	100%	100%
Exponential Decay	98.7%	99.4%	99.0%
Exponential Growth	100%	100%	100%
Funnel	98.2%	100%	100%
Linear	97.8%	99.8%	99.6%
Logarithmic Growth	78.3%	98.0%	95.5%
Parabola	98.5%	99.5%	98.5%
Sine	93.8%	98.6%	99.1%

Table 3: Accuracy for all models on Trained Distributions. Cells highlighted in red represent values that are significantly lower than the other accuracy values.

sine). Each of these distributions have lower accuracies across each of the three models. This indicates that all of our models struggle to correctly classify these distributions.

Distribution	Logistic Regression	Generalized Additive Model	Random Forest
Absolute Value	80.1%	76.8%	56.9%
Cubic	100%	100%	100%
Linear	97.2%	99.2%	99.3%
Linear near 0 slope	61.0%	76.0%	72.0%
Reciprocal	93.3%	95.3%	91.9%
Sech	99.1%	96.4%	94.5%
Rotated Sine	29.0%	41.0%	27.0%
Tanh	100%	100%	100%

Table 4: Accuracy for all models on Additional Testing Distributions. Cells highlighted in red represent values that are significantly lower than the other accuracy values.

Also, the convolutional neural networks showed accuracy on a validation set for linear graphs was 98%. We observed a greater decrease in accuracy when predicting graphs derived from distributions outside our original training set. These accuracies can be observed in Table 5.

Distribution	GAM	Random Forest	Logistic Regression	CNN
Original Train Accuracy	99.8%	99.5%	97.3%	98.4%
Additional Test Accuracy	88.5%	83.6%	84.3%	77.6%

Table 5: Accuracy for all models including Convolutional Neural Network.

7 Discussion

7.1 Statistical Models

The accuracies for each of the models were extremely high on scatterplot distributions that the models were trained on. But when exposed to additional testing distributions, the models' accuracies decreased significantly. In particular, by looking at Table 4, we see that the absolute value, linear near 0 slope, and rotated sine distributions have significantly lower accuracies than the other distributions. These three distibutions are displayed in Figure 21.

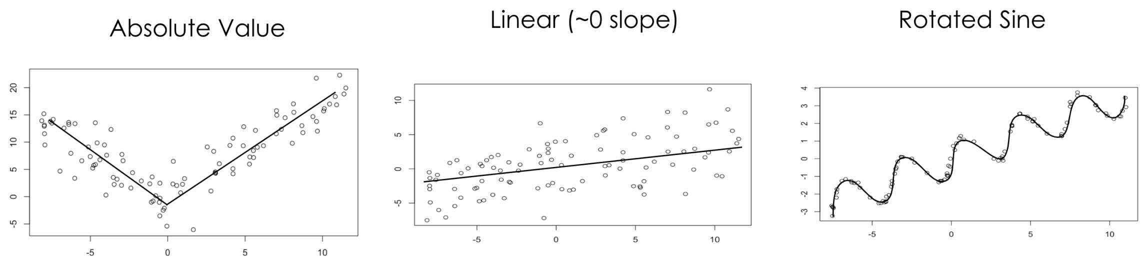


Figure 21: Examples of the three problematic distributions our models were tested on.

One method to improve our models and investigate their potential to be robust is to include the additional testing distributions in the training set. After including these in the training set, we see from Table 6 slight improvements in the logistic regression accuracy and significant improvements of the random forest and GAM accuracies. The original accuracies for these three distributions are highlighted in red and the accuracies after training on the additional testing distributions is highlighted in blue.

Distribution	**Logistic Regression**	**Generalized Additive Model**	**Random Forest**
Absolute Value	80.1% → **78.2%**	76.8% → **80.1%**	56.9% → **77.8%**
Cubic	100%	100%	99.8%
Linear	97.8%	99.3%	99.5%
Linear near 0 slope	61.0% → **69.9**%	76.0% → **91.2%**	72.0% → **90.2%**
Reciprocal	91.4%	98.0%	98.0%
Sech	96.9%	97.5%	98.8%
Rotated Sine	29.0% → **53.5%**	41.0% → **82.1%**	27.0% → **83.1%**
Tanh	100%	100%	100%
Circle	98.4%	99.8%	99.8%
Disconnected Linear	100%	100%	100%
Donut	99.3%	99.8%	99.8%
Exponential Decay	100%	100%	100%
Exponential Growth	99.6%	100%	100%
Funnel	95.3%	99.8%	100%
Logarithmic Growth	76.7%	98.2%	97.8%
Parabola	98.4%	99.1%	98.1%
Sine	91.7%	97.8%	97.8%

Table 6: Accuracies for all models on all distributions. The change in accuracies for each of the three problematic distributions are displayed with the original accuracy in red and the new accuracy in blue.

To ensure accuracy is not lost on distributions the models were originally trained on, we tested the accuracies on distributions that were originally included in the training set. The logistic regression model and GAM see larger decreases in overall accuracy than the random forest, but the decreases for all three are negligible. This is likely due to the random forest's ability to learn and adapt to new training distributions.

The models' improvements after including the additional testing sets in the training process leads to an insightful inference. These models are highly sensitive to the training data and can only be used on plots that follow distributions as seen in the training set. If this is the case, the models perform well.

Along with the models' sensitivity to the training data, another limitation of scagnostics is that it may be difficult for the models to identify differences between similar plots that have different classifications. For instance, a sine distribution that has a very high frequency may look similar to a linear distribution with a very high variance. This comparison can be seen in Figure 22.

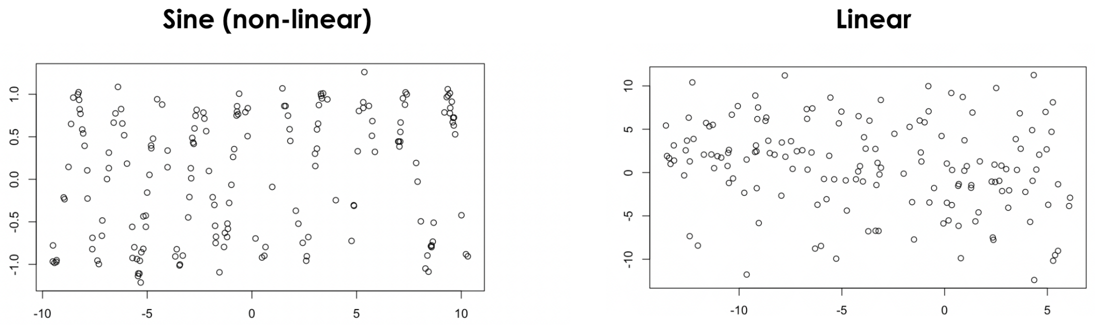


Figure 22: The left plot follows a sine distribution while the right plot follows a linear distribution. The scagnostics for these plots, however, are very similar. While it is understandable that models would struggle to differentiate these distributions, it is relatively clear that there is a wavy trend in the sine distribution.

There exist other scagnostics that have the potential to discern differences in similar distributions, like the two plots in Figure 22. The idea of adding more scagnostics to our models roughly resembles adding more features to a model, which is similar to the training process of a neural network. We decided to implement a CNN and compare it to our statistical models' performance to see if making the models more complex would result in accuracy improvements. While comparing our models to a CNN presents an interesting comparison, future work should be done to explore new types of scagnostics that may lead to higher accuracy measures.

7.2 Convolution Neural Networks

The offer of CNN's on the type of task we've been working on is promising. Classifying visual images is the exact type of task these neural networks are built for, and the hope is that with enough data they could be the most optimal choice of any method we've discussed. But, the nature of such a model comes with several drawbacks, most notably a decrease in our ability to interpret the results.

In this paper, we've been discussing how to use individual scagnostics to assess the linearity of a particular plot. As laid out in the section dedicated to describing those scagnostics, it's possible to see precisely where they come from and exactly where they appear in a plot. If we were to build a CNN that could perfectly or near-perfectly assess the linearity of a pair of plots (something our introductory research indicated was more difficult than one might expect), we would have a good tool to achieve a high level of accuracy. However, we will have no intuition about what our model considers when coming to conclusions unlike the statistical models that rely on scagnostics. CNNs are notoriously difficult to interpret, and one of our primary goals throughout this research was to create an interpretable model, like the random forest (Yosinski et al., 2015). Therefore, while CNNs have great potential in improving accuracies of classifying patterns in scatterplots, there is reason to favor statistical models that provide users with high levels of interpretability.

7.3 Residuals

This paper has primarily focused on our efforts to classify graphs in the "linear" setting - or in the original scatterplot setting. The reason for that is it's a reasonably clearly defined problem that might have some applications in the real world. Once we figured out this was a possible task, however, we focused on another potentially more "productive" application. Specifically, we investigated whether or not residual plots from linear regression could be classified using these same techniques. Since univariate and multivariate regression both result in 2D plots where a "good" residual plot looks relatively similar, this technique could create a universal residual classifier. We ran all four models discussed in this paper (the three statistical models and a CNN) and found that the same techniques can correctly classify these residuals. The more uniform nature of residual plots indicating a model's fit means we observed better accuracy when we classified them. The exact accuracies we achieved when predicting those in our original training distributions are listed in Table 7.

Distribution	**GAM**	**Random Forest**	**Logistic Regression**	**CNN**
Original Train Accuracy	98.1%	98.1%	95.6%	98.2%
Additional Test Accuracy	93.7%	95.6%	92.4%	89.5%

Table 7: Accuracy for all models in residual context.

It's worth noting that the drop in accuracy for the CNN as compared to the other models is reduced, indicating that with further data and tuning, the residual context might be an even stronger application of our CNN model. Our hypothesis for why our models did better on the residual plots from our additional testing distributions when compared to their performance in the linear setting is that while linear plots have a wide array of possible graphs: positive slope, negative slope, near zero slope, etc., the residual context is much more uniform. A well-fit residual plot in linear regression will either be a set of points roughly spread around the $y = 0$ line, while a bad residual plot will have some sort of trend or deviation away from that single "optimal" graph type. The information that scagnostics have trouble picking up seems to be at least partially removed in the transformation from a 2D scatter plot to the linear residuals of the same plot.

8 Conclusion

Scagnostics look to be a promising method in classifying the distributions of plots given their visual characteristics. As described in this paper, there are many characteristics of scatterplots that can be quantified with scagnostics. These scagnostic values showed themselves to be relatively effective predictors in standard statistical models to classify linear and non-linear distributions. We also found that these techniques could be applied to the linear regression residual setting. The results in the residual space are quite promising and deserve more research in the future.

In the analysis of our models, we found that the random forest and generalized additive model are the most effective methods for classifying scatterplot distributions. Further exploration of more statistical methods might lead to even better results, and warrants more research going forward.

We also determined that convolutional neural networks are an effective tool to classify these graphs, but that using such technology would sacrifice the ability to interpret the resulting classifications. The research we've conducted was focused on being able to reflect what humans see visually within our models, and in the CNN setting that reflection isn't possible post-classification.

References

Chauhan, A. (2021). Random forest classifier and its hyperparameters. https://medium.com/analytics-vidhya/random-forest-classifier-and-its-hyperparameters-8467bec755f6. Accessed on April 20, 2023.

Chihara, L. M. and Hesterberg, T. C. (2018). *Mathematical Statistics and Resampling and R*. Wiley.

Doiron, N. and Ankerst, M. (2020). Graphical perception assessment and multidimensional scaling of scatterplot matrices. *IEEE Transactions on Visualization and Computer Graphics*, 26(1):107–116.

Dowle, M. (2018). Classification from scratch: Logistic with splines. https://www.r-bloggers.com/2018/05/classification-from-scratch-logistic-with-splines-2-8/. Accessed: May 9, 2023.

Feng, D. and Sang, H. (2021). *mcgv: Multivariate Covariance Generalized Linear Models*. R package version 0.1.0.

Fisher, R. A. (1936). The use of multiple measurements in taxonomic problems. *Annals of Eugenics*, 7(2):179–188.

Fox, J. (2015). Generalized linear models. *Journal of the American Statistical Association*, 113(522):617–624.

Geyer, C. J. and Meeden, G. (1993). The fitting of generalized linear models using the iteratively reweighted least squares algorithm. *Communications in Statistics - Theory and Methods*, 22(1):23–44.

Gower, J. C. and Ross, G. J. S. (1969). Minimal spanning trees and single linkage cluster analysis. *Applied Statistics*, 18(1):54–64.

Guo, Z. and Agrawala, M. (2019). Machine learning for visualization: Discriminative embeddings of graphical perception. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5535–5544.

Hartigan, J. A. and Mohanty, S. (1992). The runt test for multimodality. *Journal of Classification*, 9(1):63–70.

He, K., Zhang, X., Ren, S., and Sun, J. (2015). Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE International Conference on Computer Vision*.

James, G., Witten, D., Hastie, T., and Tibshirani, R. (2013). *An Introduction to Statistical Learning: with Applications in R*. Springer.

Jean-Romain Roussel (2018). CassowaryR: An R Interface to the Cassowary Constraint Solver.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012a). Convolutional neural networks for visual recognition. In *Advances in Neural Information Processing Systems*.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012b). Imagenet classification with deep convolutional neural networks. *Communications of the ACM*.

Kromrey, J. and Wongsuphasawat, K. (2019). Scalable multivariate visual analytics using the scatterplot matrix. *IEEE Transactions on Visualization and Computer Graphics*, 25(1):814–824.

Matejka, J. and Fitzmaurice, G. (2017). Same stats, different graphs: generating datasets with varied appearance and identical statistics through simulated annealing. *CHI'17 Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, pages 1290–1294.

Phung and Rhee (2019). A high-accuracy model average ensemble of convolutional neural networks for classification of cloud image patches on small datasets. *Applied Sciences*, 9:4500.

Pickens, E. (2019). Graphical inference with convolutional neural networks. *arXiv preprint arXiv:1902.02168*.

Stuetzle, W. (2003). Estimating the cluster tree of a density by analyzing the minimal spanning tree of a sample. *Journal of Classification*, 20(1):25–47.

Tukey, J. and Tukey, P. (1985). Computer graphics and exploratory data analysis: An introduction. In *Proceedings of the Sixth Annual Conference and Exposition: Computer Graphics*, Fairfax, VA, United States. National Computer Graphics Association.

Tukey, J. W. (1977). *Exploratory Data Analysis.* Addison-Wesley Publishing Company, Reading, MA.

Wickham, H. and Stryjewski, L. (2011). Understanding scatterplots: A convex hull approach. *Journal of Computational and Graphical Statistics*, 20(1):1–19.

Wilkinson, L., Anand, A., and Grossman, R. (2005). Graph-theoretic scagnostics. In *IEEE Symposium on Information Visualization, 2005. INFOVIS 2005.*, pages 157–164. IEEE.

Wilkinson, L. and Wills, G. (2008). Scagnostics distributions. *Journal of Computational and Graphical Statistics*, 17(2):473–491.

Yosinski, J., Clune, J., Nguyen, A., Fuchs, T., and Lipson, H. (2015). Understanding neural networks through deep visualization. *arXiv preprint arXiv:1506.06579.*