Data Visualization

Contents

- **Part I**
  - What is data visualization ?
  - Why is data visualization so important ?
  - Common visualization libraries in Python
  - Data types, relationships and visualization types
  - How to select a graphic ?
- **Part II**
  - Data visualization and model explainability
  - Case study

- **Part I**

What is data visualization?

- Data visualization is the **graphical representation** that contains **information** and **data**.
- Data visualization tools provide an accessible way to see and understand trends, patterns, correlations and outliers in data.
- General types of data visualization are **Charts, Tables, Graphs, Maps, Dashboards……**

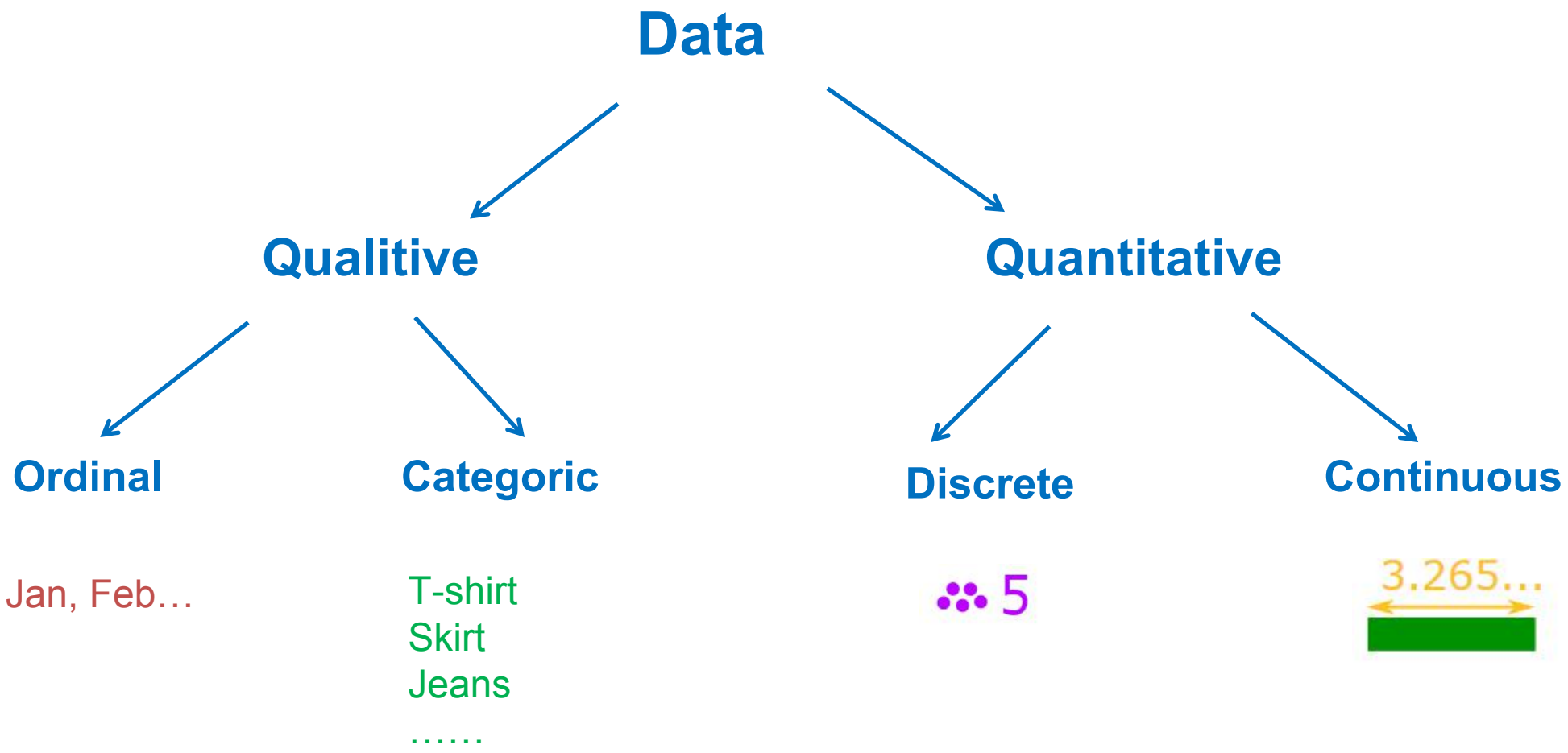
Benefits of good data visualization

- It is a powerful technique to **explore** the data with **presentable** and **interpretable** results.
- In the **data mining process**, it acts as a primary step in the pre-processing portion.
- It supports the **data cleaning process** by finding incorrect data and corrupted or missing values.
- It also helps to **construct and select variables**, which means we have to determine which variable to include and discard in the analysis.
- In the process of **data reduction**, it also plays a crucial role while combining the categories.

Common data visualization libraries in Python

	Matplotlib	Seaborn	Plotly	Pandas
Characteristics	• The **first Python data visualization library**. • It offers numerous rendering backends and uses a verbose syntax • Give plots a high degree of **flexibility and customizability**	• Built on top of Matplotlib that uses short lines of code to create and style statistical plots from Pandas dataframes. • It allows for **a concise but limited approach to quickly visualize** data sets with **better-looking** style defaults than Matplotlib	• A mostly open-source data analytics and visualization tool (with **some closed-source** products and services). • It creates **interactive charts** for web browsers. • It supports **multiple languages**, e.g. python, Julia, R and Matlab	• Mainly for **tabular data manipulation and analysis**, with built-in plotting functions that rely on matplotlib
Use cases	to create highly customized plots or look to learn the plotting tool behind seaborn	to write concise code and create plots (especially statistical plots) with more attractive default styles in less time	to display interactive data visualizations on the web, or use other programming languages	to organize and rearrange the data to create proof-of-concept visualizations without using other libraries explicitly

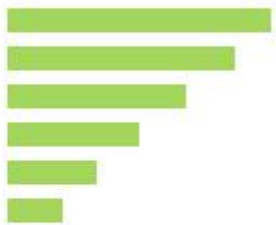
2 kinds of data



7 data relationships

Data relationships can be simple, like the progress of a single metric over time (such as visits to a blog over the course of 30 days or the number of users on a social network), or they can be complex, precisely comparing relationships, revealing structure, and extracting patterns from data. There are **seven data relationships** to consider:

Ranking: A visualization that relates two or more values with respect to a relative magnitude. For example: a company's most sold products.



Nominal comparisons: Visualizations that compare quantitative values from different subcategories. For example: product prices in various supermarkets.

Series over time: Here we can trace the changes in the values of a constant metric over the course of time. For example: monthly sales of a product over the course of two years.

Correlation: Data with two or more variables that can demonstrate a positive or negative correlation with one another. For example: salaries based on level of education.

Deviation: Examines how each data point relates to the others and, particularly, to what point its value differs from the average. For example: the line of deviation for tickets to an amusement park sold on a rainy versus a normal day.

Partial and total relationships: Show a subset of data as compared with a larger total. For example: the percentage of clients that buy specific products.

Distribution: Visualization that shows the distribution of data spatially, often around a central value. For example: the heights of players on a basketball team.

Data visualization types

1D/Linear: List of data items, organized by a single feature

Name in alphabetic order:
Bert, Carina, Daniel, Elijah, Errol

2D/Polar (incl. Geospatial): dot distribution map, contour

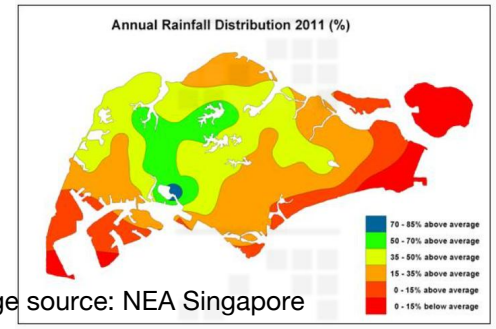


Image source: NEA Singapore

3D/Volumetric: 3D models, surface / volume rendering

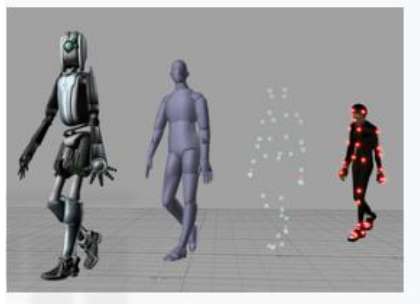
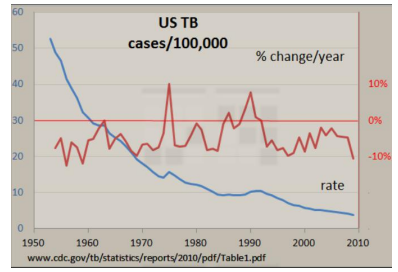
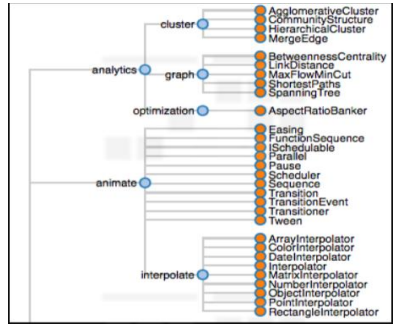


Image source: Wikipedia

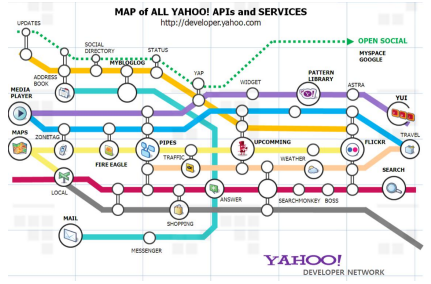
Temporal: timeline, time series



Tree / Hierarchical: general tree visualization, dendrogram, radial tree, tree map



Network: matrix, dependency graph, subway/tube map



nD / Multi-dimensional: pie chart, histogram, word cloud, bubble chart/bubble cloud, bar chart, tree map, scatter plot, line chart, area chart, heat map, radar/spider chart, box and whisker plot, waterfall

Bar plots

- One of the most popular ways for visualizing data. Can be used for numeric and categorical data.
- They are versatile, typically used to compare discrete categories, to analyze changes over time, or to compare parts of a whole
- Variations: vertical column, horizontal column, full stacked column

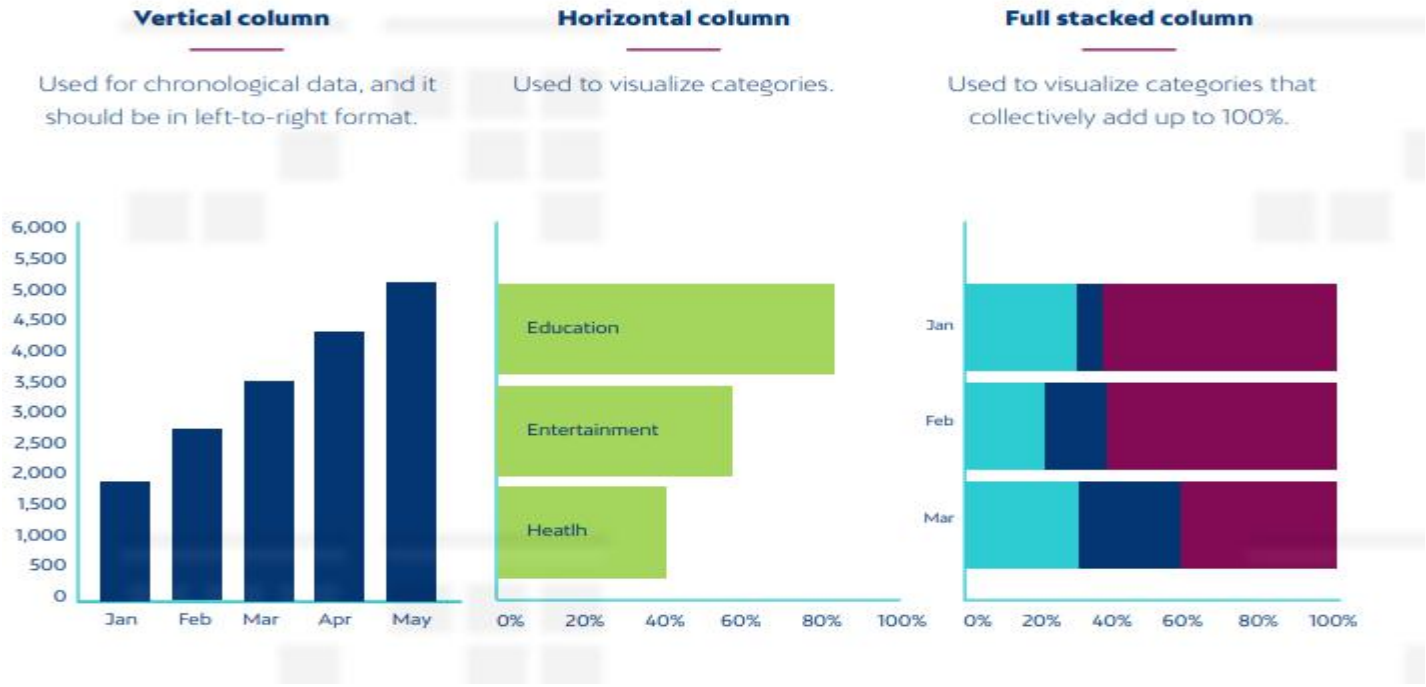
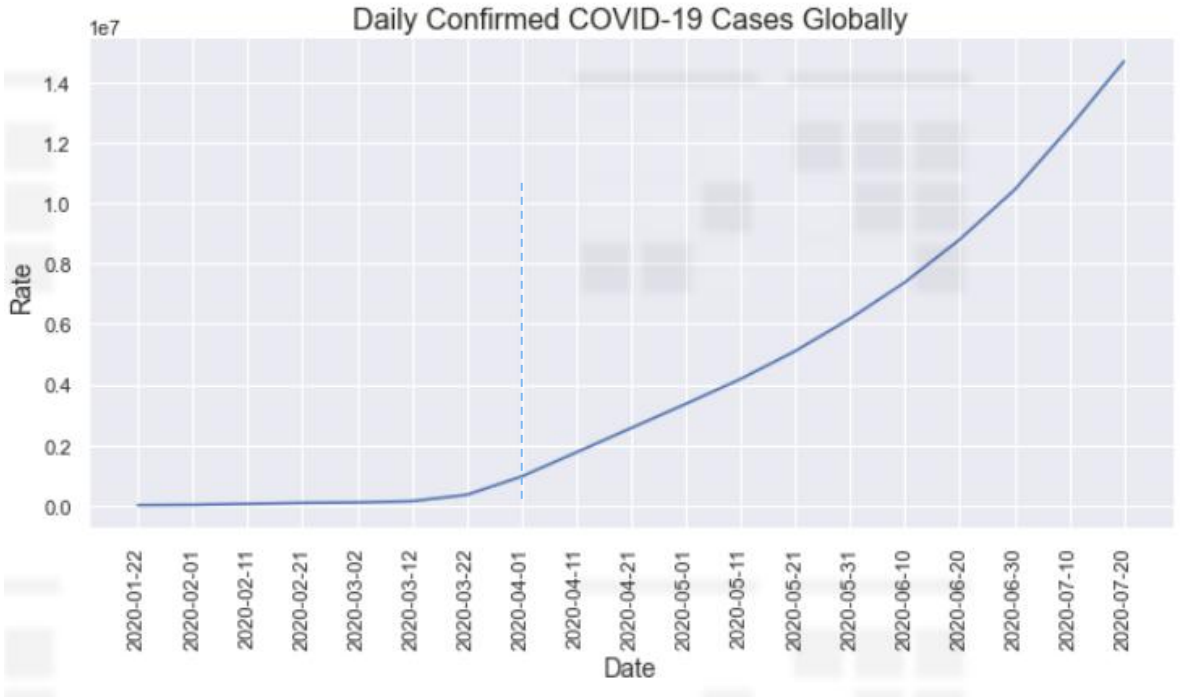


Image Source: Google Images

Line charts

- These are used to display changes or trends in data over a period of time. They are especially useful for showcasing relationships, acceleration, deceleration, and volatility in a dataset.
- Use cases: stock prices over time, visitors over a period of time……

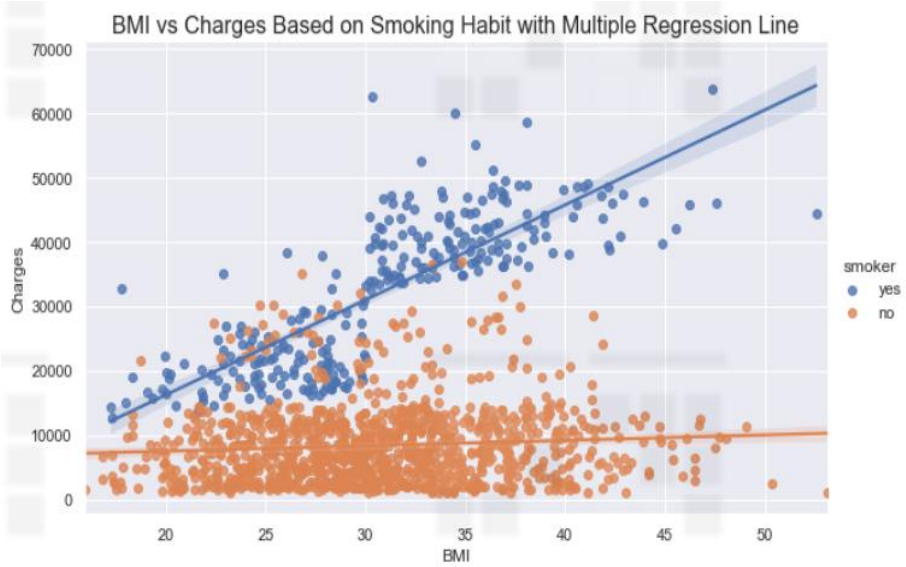


Inference:

- Covid cases start rising exponentially from April 2020

Scatter plots

- They use dots to represent values for two different numeric variables. They also help us to determine whether or not different groups of data are correlated.
- Use cases: weight and height distributions, visitors based on climate ……
- Extension: Regression plots (Mainly used to fit the linear regression line of the scatter plots)

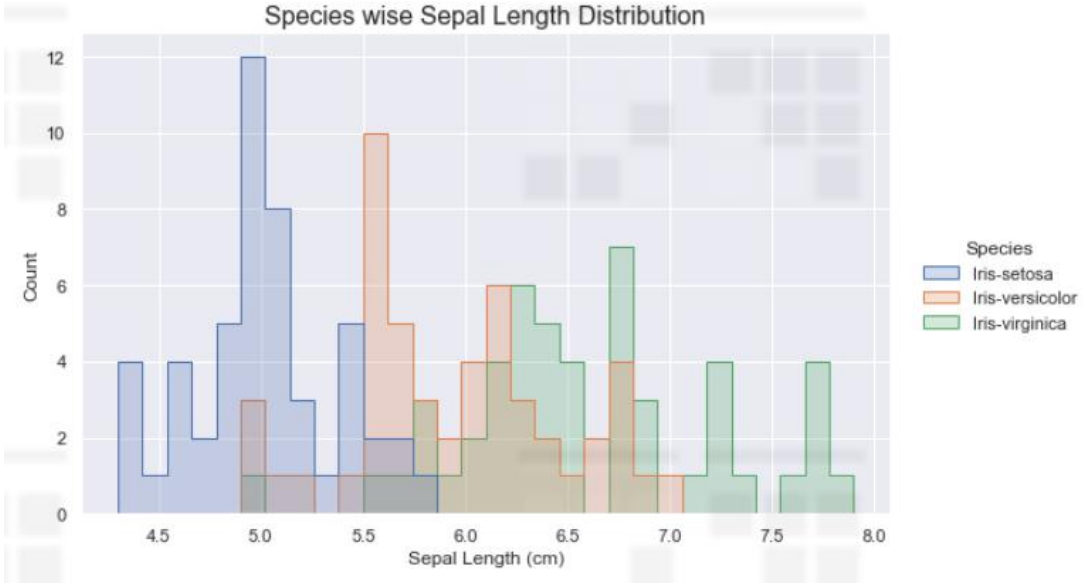


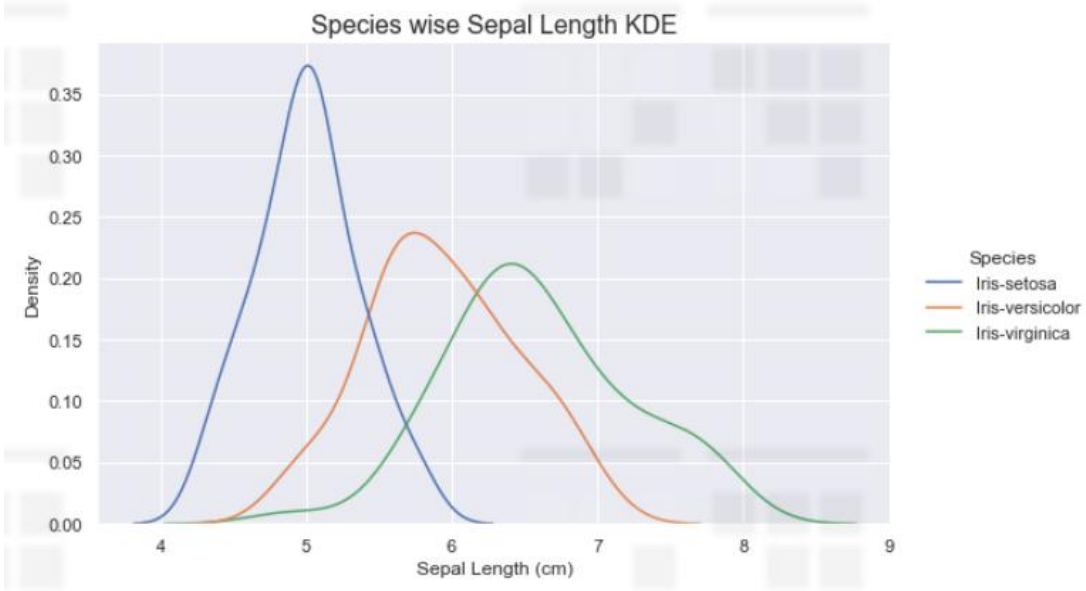
Inference:
- Persons with higher BMI tend to pay more charges compared to lower BMI.
- Smokers tend to pay more charges compared to non-smokers

Inference:
- Linear relationship between BMI and Charges based on smoking habits.

Histograms

- Histograms represent a variable in the form of bars, where the surface of each bar is proportional to the frequency of the values represented.
- The most common approach to visualize a distribution, which helps to perform univariate analysis to identify ranges, skewness, outliers, etc. of the dataset
- Extensions: KDE, Multi-plot





Inference:

- Most setosa flowers are lying in the range of 4.9 to 5.1
- Only setosa flowers have a sepal length less than 5.
- Only virginica flowers have a sepal length > 7

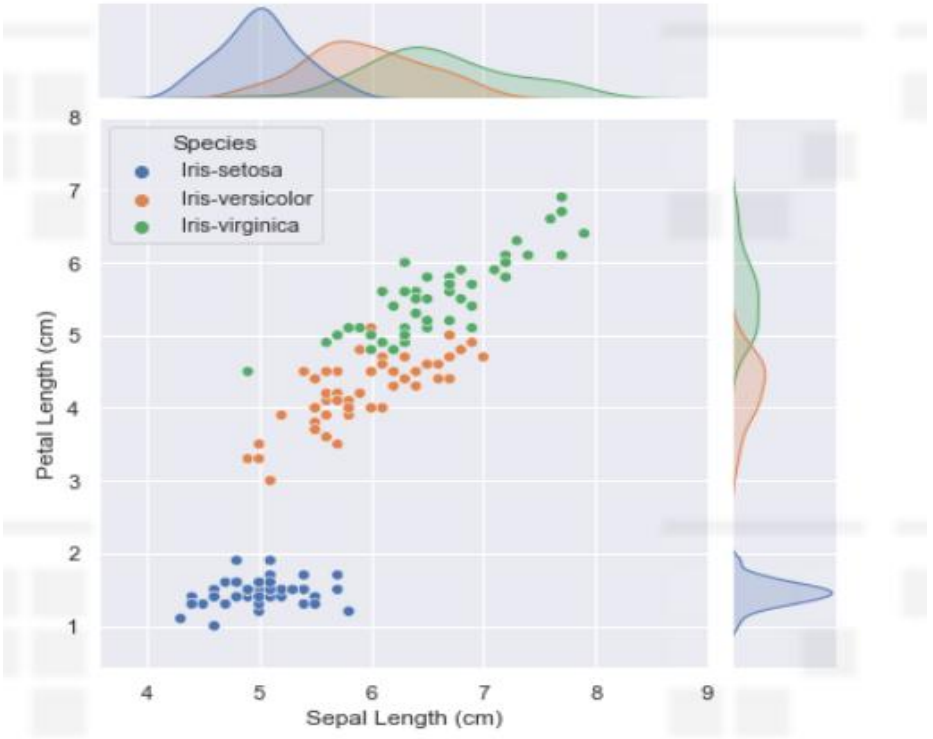
Inference:

- Setosa flowers are having a highest density and less skewed
- Virginica flowers are having a lower density and highly skewed

Histograms

- Extensions: Multi-plot

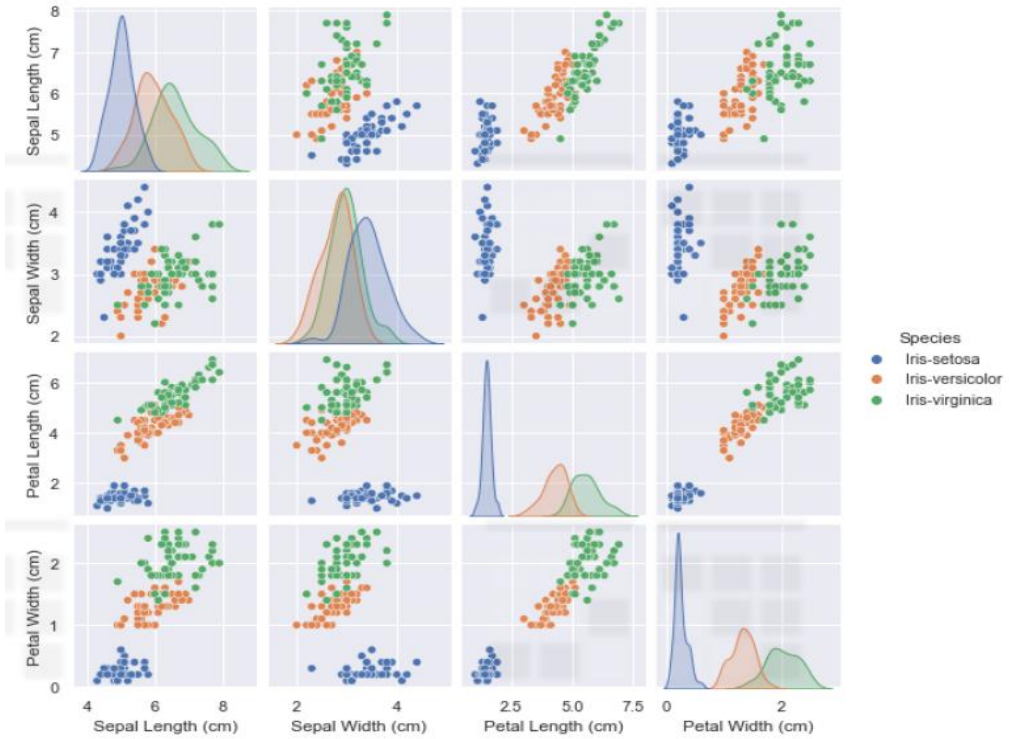
Joint plot



Inference:

- The sepal length of setosa flowers lies in the range of 4-6
- The petal length of setosa flowers lies I the range of 1-2, and it is easily distinguishable from other species
- Petal length distribution for setosa is less skewed, which it is highly skewed for versicolor and virginica

Pair plot

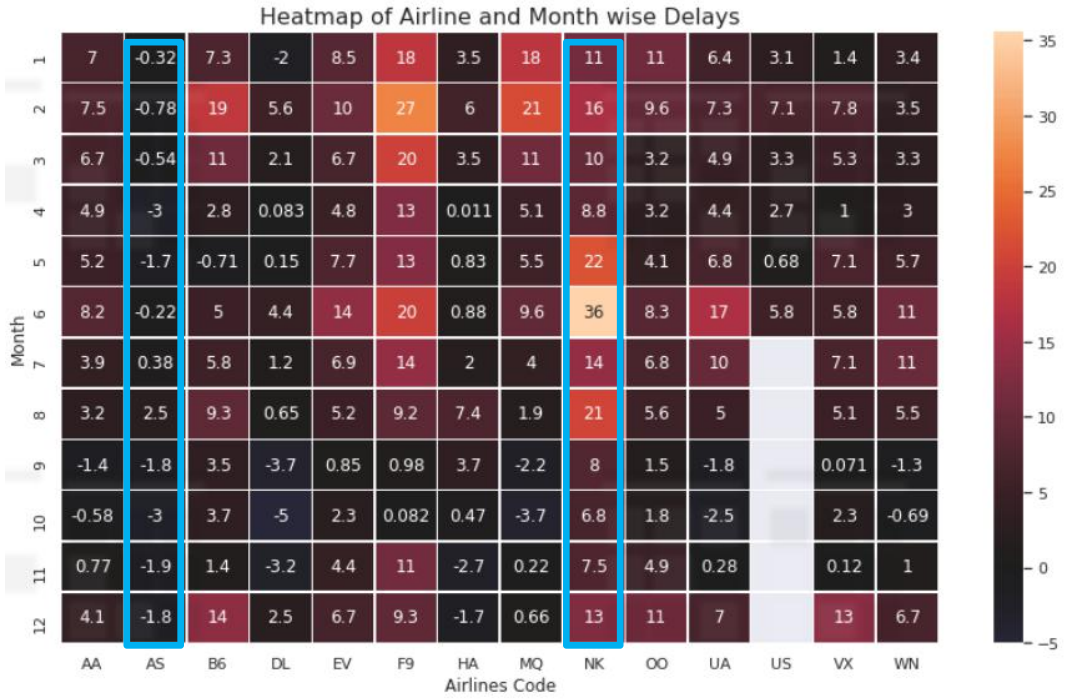


Inference:

- Same as left
- Setosa distribution is less skewed for sepal length, petal length and petal width, while it is highly skewed for sepal width

Heat maps

- Represent individual values from a dataset on a matrix using variations in color or color intensity
- They can be used to visualize the relationship between variables, or plot a correlation matrix



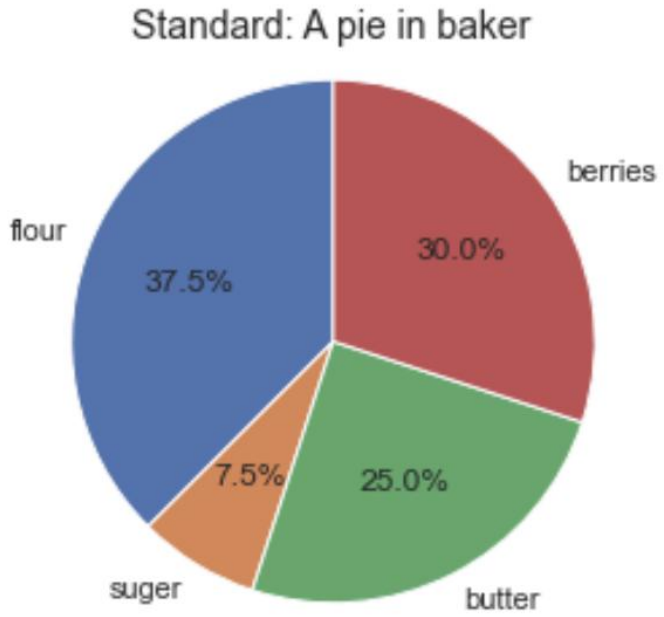
Note: White cells denotes missing data, and negative delay denotes that the flights come earlier than the mentioned time

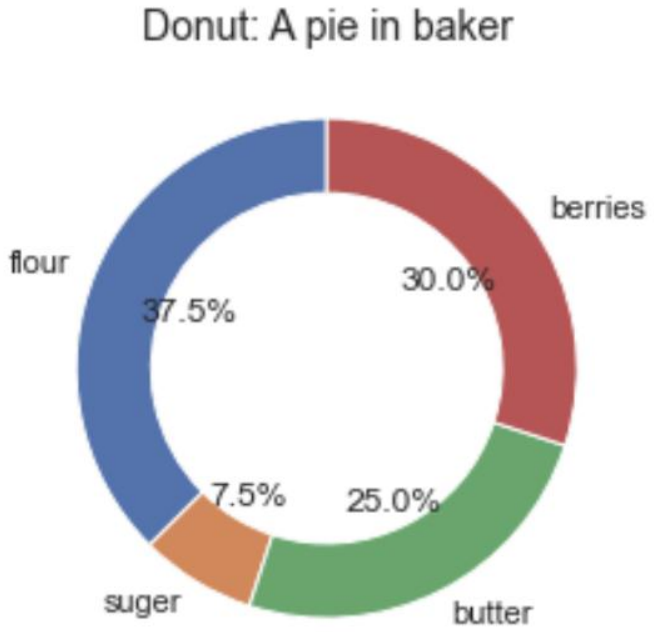
Inference:

- NK airline is having the highest delay compared to other airlines, with a highest delay in June
- AS airline is having a negative delay.
- The first six months noted higher delays compared to the last six months
- ……

Pie charts

- Pie charts consist of a circle divided into sections, each of which represents a portion of the total.
- They can be useful for comparing discrete or continuous data.
- Easy plot, user friendly, readability, but may get cluttered for more categories
- Variations: standard vs donut

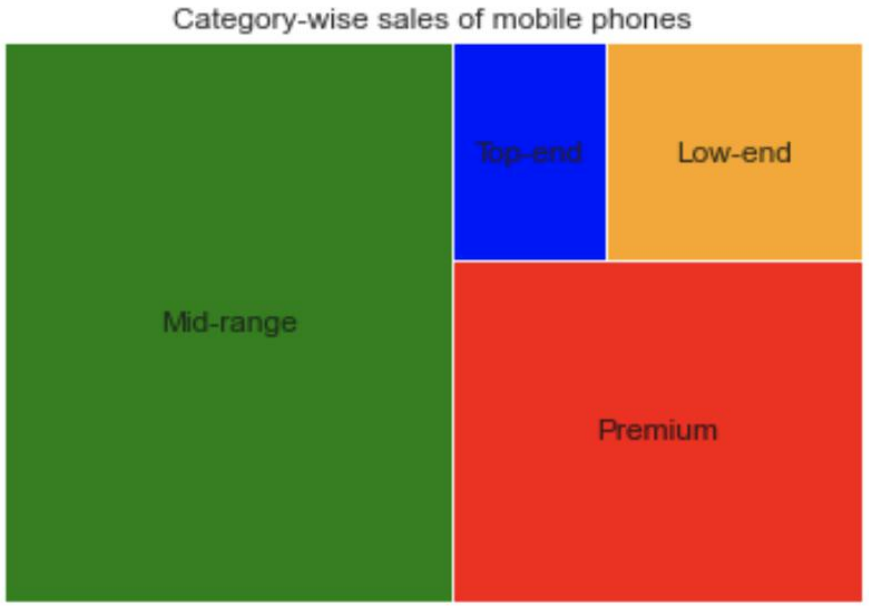




- Donut Charts: The blank circle at the center can be used to show additional statistical or general information about the data.

Tree maps

- Treemaps work well when there is a clear ‘Part-to-whole’ relationship amongst multiple categories present in the data.
- Hierarchical Data is needed. The branches and sub-branches can be visualized using rectangles of different dimensions and using more than one color.
- The focus is not on precise comparisons between categories but rather on spotting the key factors/trends or patterns.
- Use cases: category-wise product availability, customer segmentation for a product
- Benefits: space constraint, easier to read, quickly spot patterns
- Limitations: not preferrable for large data points, ineffective for balanced dataset



Inference:
- there is a bigger demand and market for Mid–Range phones
- there are limited phones available in the Top–End category

Radar charts

- A Radar Chart, also called as Spider Chart, Radial Chart or Web Chart, is a graphical method of displaying multivariate data in the form of a two-dimensional chart of three or more quantitative variables represented on axes starting from the same point.

Pros :

- Radar charts are excellent for visualizing comparisons between observations—you can easily compare multiple attributes among different observations and see how they stack up.
- It's easy to see overall "top performers"—the observation with the highest polygon area should be the best if you're looking at the overall performance.

Cons:

- Radar charts can get confusing fast—comparing more than a handful of observations leads to a mess.
- It can be tough to find the best options if there are too many variables—just imagine seeing a radar chart with 20+ variables.
- The variables have to be on the same scale—it makes no sense to compare student grades (ranging from 1 to 5) and satisfaction with some service (ranging from 0 to 100).

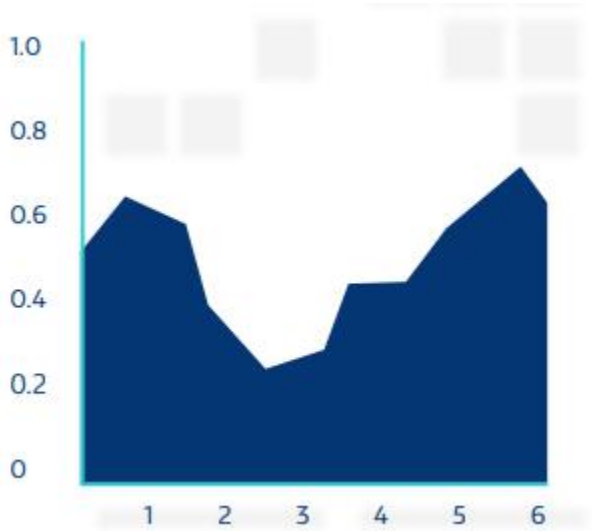
Waterfall charts

- Waterfall chart is a 2-dimensional plot that is used to **understanding the cumulative effects of sequentially added positive or negative values** for a given variable.

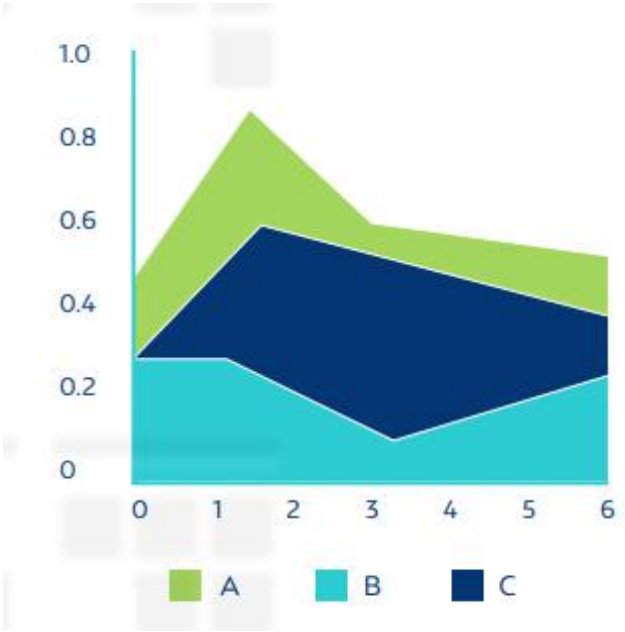
Area charts

- These represent the relationship of a series over time, but unlikely line charts, they can represent volume.
- Variations:
  - standard area: used to display or compare a progression over time
  - stacked area: used to visualize relationships as part of the whole, thus demonstrating the contribution of each category to the cumulative total.
  - 100% stacked area: used to communicate the distribution of categories as part of a whole, where the cumulative total does not matter.

Standard area



Stacked area



100% stacked area

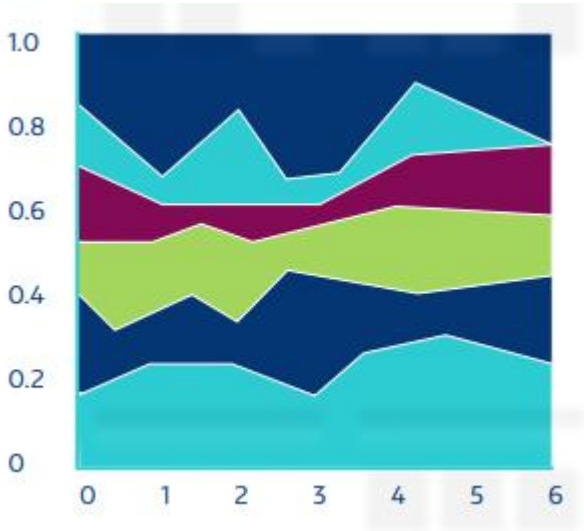
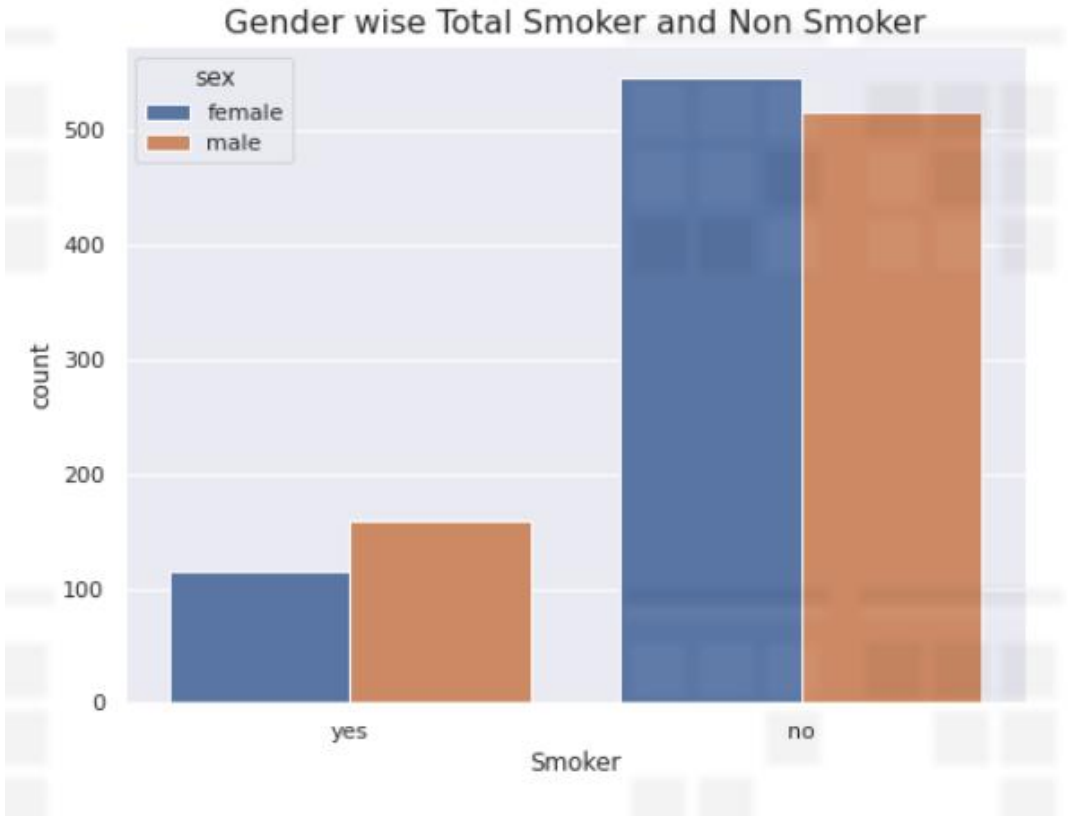


Image Source: Google Images

Categorical plots

- **Count Plot:** This type of plot helps us to show the **count of each category** of the categorical variables. It can be thought of as a histogram for categorical variables.

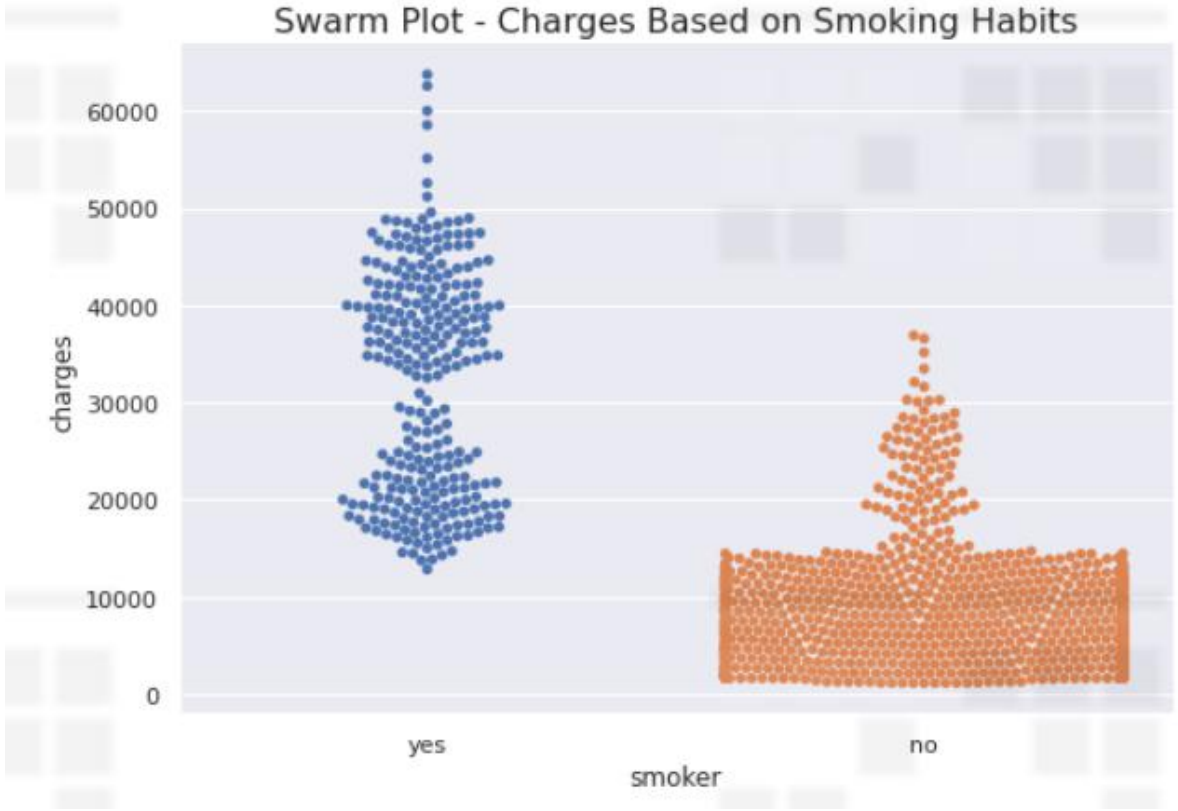


Inference

•More male smokers compared to female smokers.

Categorical plots

- **Swarm Plot:** This can be thought of as a scatter plot for the categorical variables. It shows all the data points in a figure which **helps us to identify the outliers.**

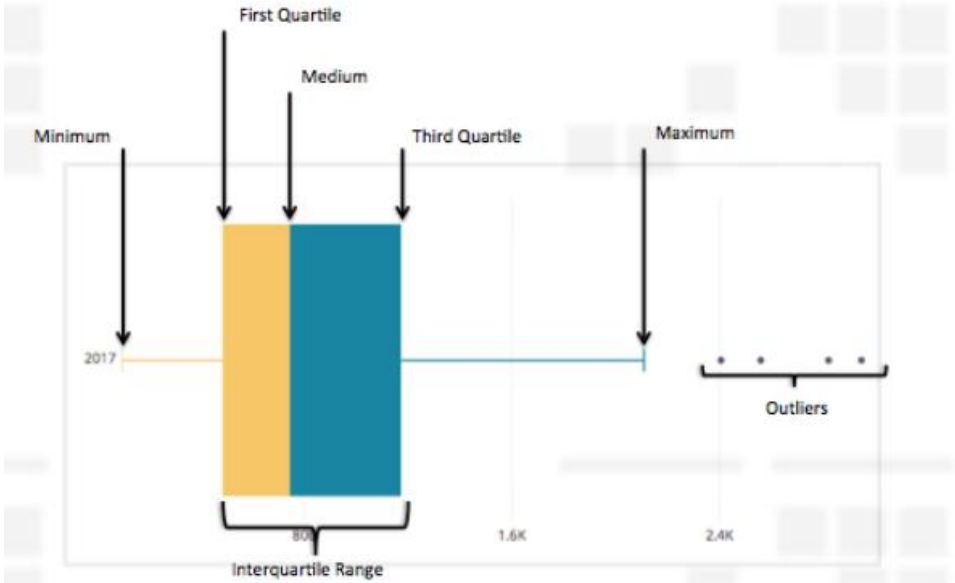


Inference

•Most non–smokers lie in the charges range of 100–1400.
•More outliers in the case of non–smokers compared to smokers.
•Smokers' charges vary more than non–smokers'.

Categorical plots

- **Box and Whisker Plot:** This is one of the most used plots in the field of Data Science. It shows the distribution of quantitative data in a way that facilitates comparisons between variables or across levels of a categorical variable. It helps us to **detect outliers** more easily compared to the swarm plots.



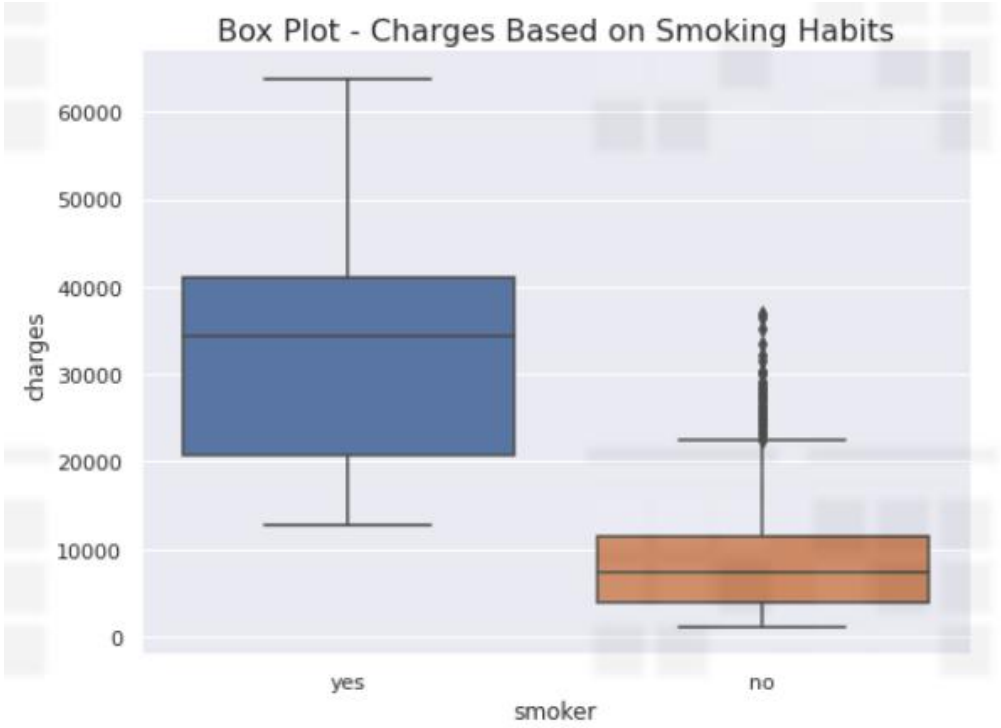


Image Source: Google Images

How to select a graphic?

Stephen Few (2009), a specialist in data visualization, proposes taking a practical approach to selecting and using an appropriate graphic:

- Choose a graphic that will capture the viewer's attention for sure.
- Represent the information in a simple, clear, and precise way (avoid unnecessary flourishes).
- Make it easy to compare data: highlight trends and differences.
- Establish an order for the elements based on the quantity that they represent, that is, detect maximums and minimums.
- Give the viewer a clear way to explore the graphic and understand its goals: make use of guide tags.

- **Part II**

Model explainability

- **What is model explainability?**
  - Model explainability refers to the concept of being able to understand the machine learning model
- **Why model explainability ?**
  - Being able to interpret a model increases trust in a machine learning model
  - Once we understand a model, we can detect if there is any bias in the model
  - Model explainability becomes important while debugging a model during the development phase
  - Model explainability is critical for getting models to vet by regulatory authorities, like Food and Drug Administration (FDA), National Regulatory Authority, etc. It also helps to determine if the models are suitable to be deployed in real life

How to develop model understanding ?

- **Option 1: build models that are inherently interpretable – Glass Box Models**

  For example – In a linear regression model of the form $y = b0 + b1*x$, we know that when x increases by 1% then y will increase by b1% keeping other factors constant

- **Option 2: Post-hoc explanation of pre-built models – Black Box Models**

  For example – In a deep learning model, the model developers are not aware of how the input variables have combined to produce a particular output.

Glass box models	**Black box models**
Simple	Complex
Interpretable	Not easily interpretable
Low accuracy	High accuracy
Examples: linear models, decision tree	Examples: random forest, deep learning

Ways to interpret a model

There are two ways to interpret the model – Local vs Global interpretation.

Local interpretation	**Global interpretation**
This helps in understanding how the model makes decisions for a single instance	This helps in understanding how a model makes decisions for the overall structure
Using local interpretation we can explain the individual predictions	Using global interpretation we can explain the complete behavior of the model
Local interpretation helps in understanding the behavior of the model in the local neighborhood	Global interpretation help in understanding the suitability of the model for deployment
Example – Understanding why a specific person has a high risk of a disease	Example – Predicting the risk of disease in patients

Ways to interpret a model: Local interpretation

- **LIME (local interpretable model-agnostic explanations)**
  - LIME provides a local interpretation **by modifying feature values of a single data sample** and observing its impact on the output.
  - It builds a surrogate model from the input (sample generation) and model predictions. An interpretable model can be used as a surrogate model.
  - Because LIME is a model agnostic technique, therefore it **can be used on any model**.

- **SHAP (SHapley Additive exPlanations)**
  - SHAP shows the impact of each feature by **interpreting the impact of a certain value compared to a baseline value**.
  - The baseline used for prediction is the **average of all the predictions**. SHAP values allow us to determine any prediction as a sum of the effects of each feature value.
  - The only disadvantage with SHAP is that the computing time is high. The Shapley values can be combined together and used to perform global interpretations also.

Ways to interpret a model: Global interpretation

- **PDP (Partial Dependency Plot)**
  - PDP explains the global behavior of a model by showing **the relationship of the marginal effect of each of the predictors on the response variable**.
  - It shows a relationship between the target variable and a feature variable. Such a relationship could be complex, monotonic, or even a simple linear one.
  - The plot assumes that the feature of interest (whose partial dependence is being computed) is not highly correlated with the other features.
  - We cannot plot PDP for all complex classifiers like Neural Networks

- **ICE (Individual Conditional Expectation)**
  - ICE is an extension of PDP(global method), but are more intuitive to understand as compared to PDP..
  - It visualizes the **dependence of the prediction on a feature for each instance** separately, resulting in one line per instance, compared to one line overall in partial dependence plots.

A case study

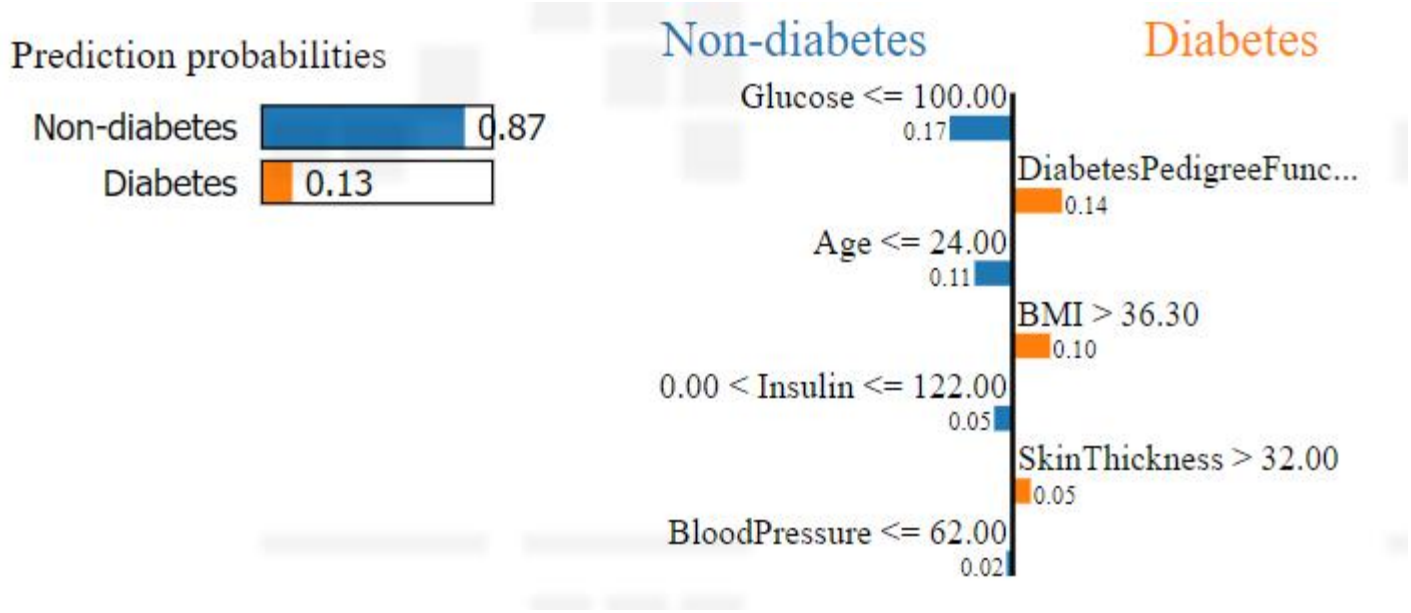
- Objective: Predict whether a patient has diabetes or not
- Dataset: Pima Indians Diabetes Database
- Model: Random forest

Explain weights (Module: ELI5 (eli5.explain_weights())

	Model	A case: y = diabetes (probability 0.736)	
Feature	Weight	Value	Contribution
Glucose	0.2326 ± 0.1904	196	0.155
BMI	0.1686 ± 0.1484	36.5	0.060
Diabetes Pedigree Function	0.1480 ± 0.110	0.875	0.130
Age	0.1445 ± 0.1298	29	-0.020
Blood Pressure	0.1214 ± 0.0904	76	-0.024
Skin Thickness	0.1004 ± 0.0842	36	0.027
Insulin	0.0844 ± 0.0768	249	0.061

A case study

- LIME: Explaining a case of non-diabetes

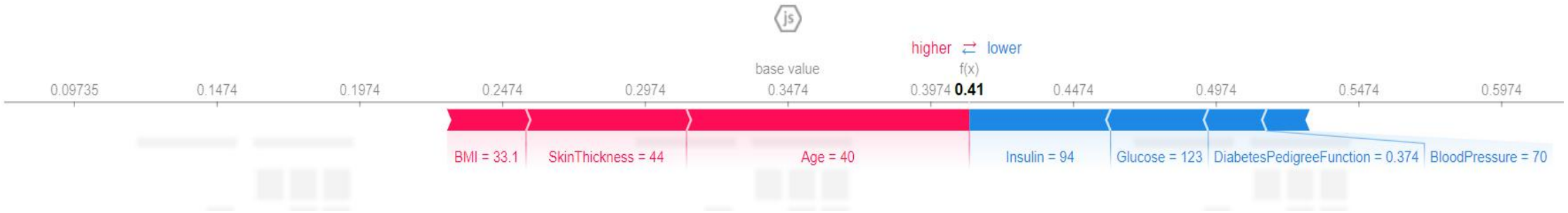


Feature	Value
Glucose	79.00
DiabetesPedigreeFunction	0.68
Age	23.00
BMI	43.50
Insulin	48.00
SkinThickness	42.00
BloodPressure	60.00

Inference: the features contributing towards Non-diabetes (denoted in blue) are Glucose<= 100, Age <=24, Insulin between 0-122, Bloodpressure <=62

A case study

- SHAP: Explaining a case of diabetes



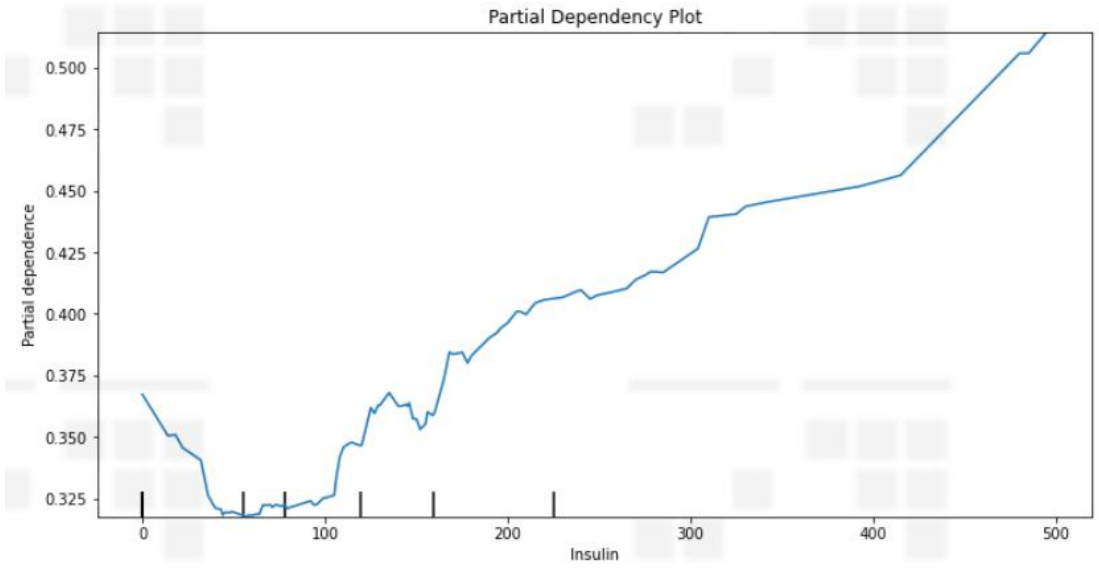
Inference:
This is a case of diabetes. Here, most of the features is moving towards the base value.
The features contributing towards explanation of diabetes are high values of BMI, skin thickness and Age

```
explainer = shap.TreeExplainer(rf_clf)
shap_values = explainer.shap_values(data_for_prediction_array)
shap.initjs()
shap.force_plot( explainer.expected_value[1],
                 shap_values[1],
                 data_for_prediction,
                 figsize=(20, 2) )
```

A case study

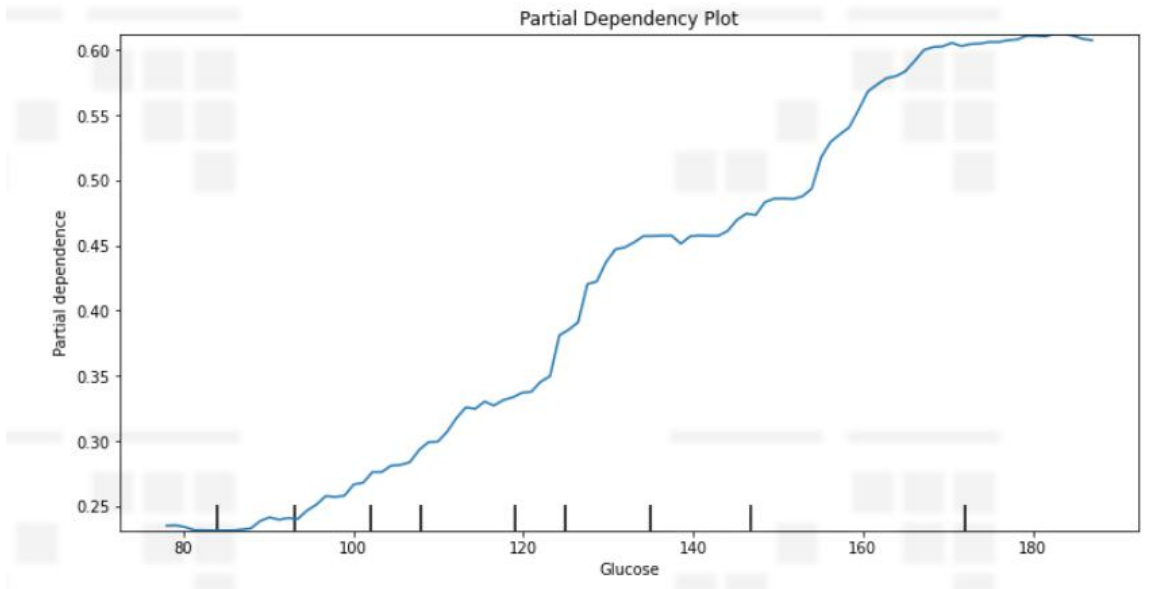
- Global explanation: Partial dependence plots

Effect of Insulin on Diabetes



Inference: initially diabetes level is high then it decreases at the level when insulin is normal. The chances of diabetes increase again with rising insulin

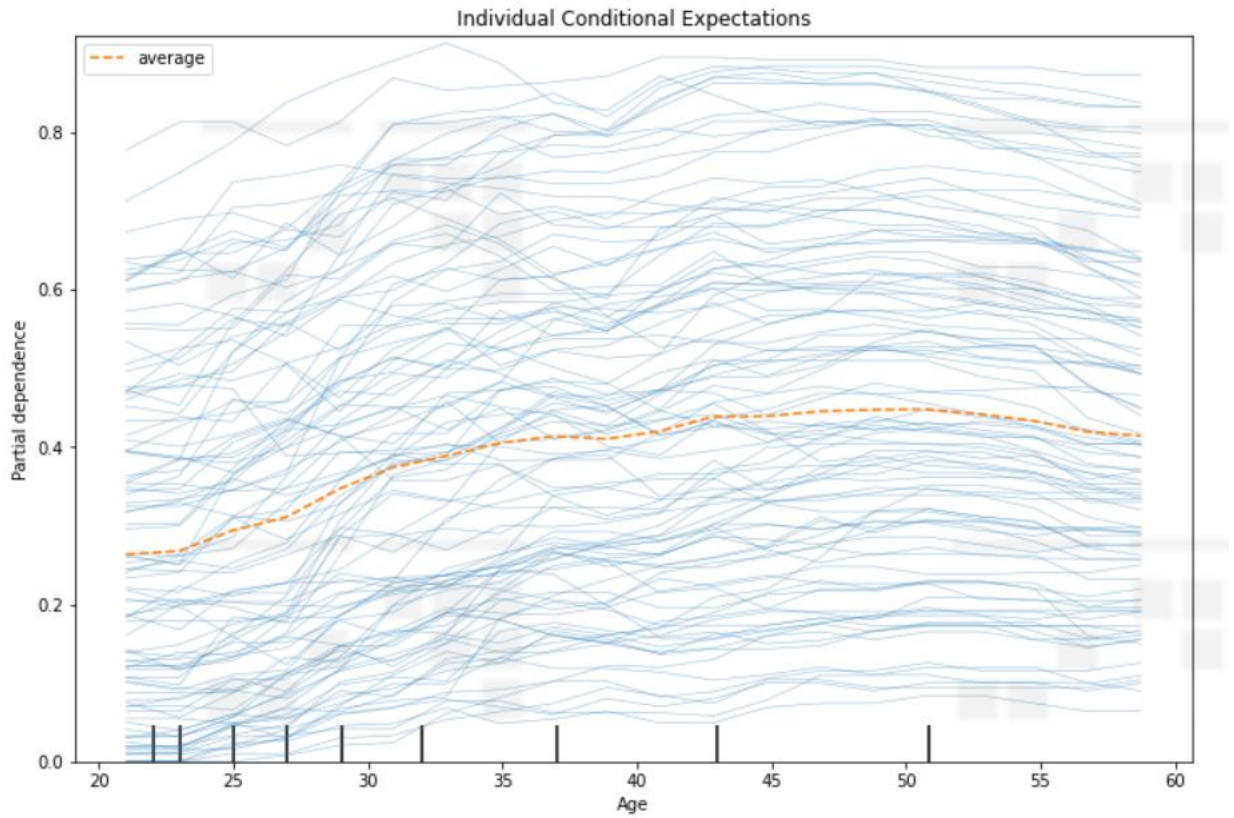
Effect of Glucose on Diabetes



Inference: with increasing level of glucose, the probability of diabetes increases

A case study

- **Global explanation:** Individual Conditional Expectations (ICE)

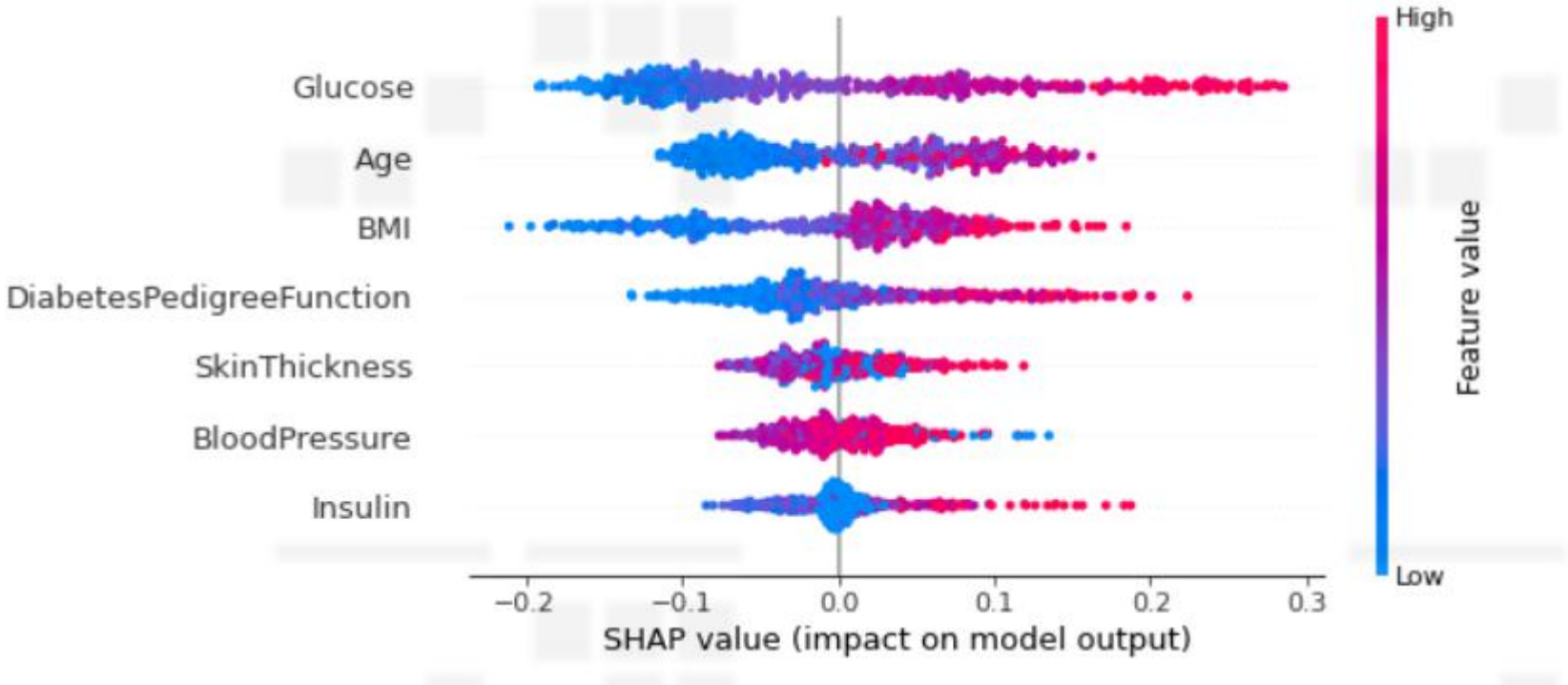


Inference:

- this shows the effect of Age on the target variable, keeping other features constant
- Here, the average is the PDP

A case study

- Global explanation: SHAP




```
Code snippet:
explainer = shap.TreeExplainer( rf_clf )
shap_values = explainer.shap_values( X_train )
shap.summary_plot( shap_values[1], X_train, plot_type = 'dot' )
```

Q&A