Machine learning cousera

Look into:

Fundamental understanding of Calculus, Linear Algebra, Probability, and Statistics

Deep learning subset of machine learning and subset of AI.

AI breakthroughts:

1. Image classification: identification of for example dog and cat.
2. Machine translation: take a phrase and give the translation in another language.

AI:

Any program that can sense, reason act and adapt.

Ml:

Replicate intelligence behaviour and learn as more data is explosed to it

Deep learning:

Learn as more data is exposed to it and also uses algorithms.

Machine learning:

* Learn patterns as they are exposed to more data.
* Programs learn by repeatedly seeing data.
* There are features in machine learning and this is the attributes of the data.
* Target is the column to be predicted.

2 types of machine learning:

Supervised learning

Unsupervised learning

Supervised learning:

Has a target column

* Make predictions.
* E.g fraud detection

Unsupervised learning:

Does not have a target column

* Find structure in the data without labels
* E.g customer segmentation
* No right or wrong answer.

To identify fraudulent credit card transactions

We can determine features to be:

* Transaction time
* Transaction amount
* Transaction location
* Category of purchase.

The algorithm could learn what feature combinations suggest unusual activity. (we use transaction time, amount, location and purchase).

Machine learning limitations:

Suppose you want to determine if an image is of a cat or a dog:

What features would you use??

* Pixel -> different pixels that makes up the eyes or face. So there wll be too many featues
* Therefore pixel not good
* We use deep learning

Deep learning:

* Determine best representation of our data.
* Deals with larger datasets.
* If do not have a steady datasets use machine learning as it will do a better job

In image recognition:

* For machine learning, we need to identify the features ourselves before we feed the data to our models.
* Then we include the features in ml algorithms
* And then predict the picture

Deep learning:

* Combine step 1 and step 2 of machine learning. The neural netowek receives these pxiels as an input and neural networks extracts what is important. Highlists edges and makes shapes like nose etc.
* Good for image classification.

History of AI:

Early algorithms

AI winter late 1960-1970s

Exprt systems

Netural netoweks

AI winter 1980-90

Machine learning

Deep learning

**1950: early AI**

1950: Alan turing developed the turing test to test a machines ability to exhibit intelligent behaviour

* Seved as foundation threshold for AI
* Imitates human

1956: AI was accepted as a field at the Dartmouth Conference.

* At the conference. This is where AI name came from

1957: Frank Rosenblatt invented the perceptron algorithm. This was the precursor to modern neural networks.

1959: Arthur Samuel published an algorithm for a checkers program using machine learning.

* Arthur Samuel normalized the term ml

**First AI winter**

1966: ALPAC committee evaluated AI techniques for machine translation and determined there was little yield from the investment.

* During the cold war.

1969: Marvin Minsky published a book on the limitations of the Perceptron algorithm which slowed research in neural networks.

1973: the lighthil report highlights AI’s failure to live up to promises.

* Mathematician Lighthill

These reports led to govt cut

**1980s AI boom**

Expert systems: systems with programmed rules designed to mimic human experts.

* Ran on mainframe compuuters with specialized prog languaes(lisp)
* Gained huge popularity
* 1986: the “backpropagation” alogirthm is able to train multi layer perceptrons leading to new successes and interest in neural network research: for deep learning – important algorithm

**AI winter**

Expert system’s progress on solving business problems slowed.

i.e we could give them data and theyd make horrible output

* Expert systems began to be melded into software suites of general business applications(sap or oracle) that could run on PCs instead of mainframes.
* Neural networks didn’t scale to large problems.
* Interest in AI in business declined

**AI boom**

1990 – 2000: machine learning:

All solutuions had successes in speech recognition, medical diagnosis, robotics and many other areas

* Al algorithms were integrated into larger systems and became useful throught the industry
* The deep blue chess system beat world chess champiton garry Kasparov
* Google’s search engine launched using Aitechnology.

2006: Geoffrey hinton publishes a paper on unsupervised pretraining that allowed deeper neural networks to be trained.

* Neural networks are rebranded as deep learning.

2009: the imageNet database of human-tagged images is presented at the CVPR conference

* Dataset provided with million of labelled datasets

2010: algorithms compete on several visual recognition tasks at the first imageNet competition

2012: deep learning beats previous benchmark on the imageNet competition.

* Alex net
* Blew the competition

2013: deep learning us used to understand conceptual meaning of words.

2014: similar breakthrougts appeared in language translation

These have led to advancements in web search, document search, document summarization, and machine translation.

2014: Stanford team creates computer vision algorithm that can describe photos.

2015: deep learning platform tensorflow is developed.

2016: deepmind’s alphago developed by aja huang beats go master lee se dol

2018: waymo launches commercial self-driving car serve in suburbs of phoenix

2019: IBM project debater is able to have a full debate with rebuttal with champion human debater.

Modern AI:

Computer vision and natural language

Computer vision -> self driving cards: onbject detection

Healthcare: improved diagnosis

NLP: communication: language translation

How is this era of AI different:

* Bigger datasets required to large complex patterns
* Faster computers
* Neural nets
* Leads to cutting edge results in a variety of fields:
* Our phone -> detects our face etc etc

Transformative changes:

Health:

* Enhanced diagnosis
* Drug discovery
* Patient care
* Research
* Sensory aids

Industrial:

* Factory automation
* Predictive maintenance
* Precision agriculture
* Field automation

Finance:

* Algorithm trading
* Fraud detection
* Research
* Personal finance
* Risk mitigation

Energy:

* Oil and gas exploration
* Smart grid: optimize electrical grid for insfrastructure
* Operational improvement
* Conservation

Govt:

* Defense
* Data insights: treats, weather, citizens needs
* Safety and security
* Engagement
* Smarter cities.

Transport:

* Automous cards
* Automated trucking
* Aerospace
* Shipping
* Search and rescue

And more:

* Advertising
* Education
* Gaming
* Professional and it services
* Telco/media
* Sports

Applications

Ai omnipresence in transportation:

* Navigation: google and waze find the fastest route by processing traffic data.
* Ride sharing: uber ad lyft predict real time demand using Al techniques, machine learning and deep learning.

In social media:

Audience: facebook and twitter use AI to decide what content to present in their feeds to different audiences

Content: image recognition and sentiment analysis to ensure the content of the appropriate mood is being served.

In daily life:

* Natural language: we carry around powerful nlp al in our phones/computers
* Object detection: cameras like amazon’s deeplens or google clips use object detection to determine when to take a photo.

Latest dev: computer vision:

2012: deep learning proven to work for image classification

2015: models outperform humans on image classification

2016: object detection models beat previous benchmarks.

e.g: abandoned baggage detection:

we can automatically detect when baggage has been left unattended potentially saving lives.

The system relies on the breakthroughs we discussed:

* Cutting edge object detection
* Fast hardware on which to train model

Machine learning workflow:

Statistics: prob, calculating moments, baye’s rules/ linear algebra

Lib:

* Numpy
* Pandas
* Matplotlib
* Seaborn
* Scikit learn
* Tensorflow
* Keras

Ml workflow:

* Problem statement: what problem are u tryin to solve
* Data collection: what data do u need to solve it
* Data exploration and preprocessing: how should u clean ur data so ur model can use it?
* Modelling: build a model ur problem?
* Validation: did I solve ur problem?
* Decision making and deployment – communicate to stakeholders or put into production

Vocab:

Target: category or value that we are trying to predict

Features: properties of the data used for prediction (explanatory variables).

Example/observation: a single data point within the data(one row).

Label: the target value for a single data point.

Summary/Review

Introduction to Artificial Intelligence and Machine Learning

Artificial Intelligence is a branch of computer science dealing with the simulation of intelligent behavior in computers. Machines mimic cognitive functions such as learning and problem solving.

Machine learning is the study of programs that are not explicitly programmed, but instead these algorithms learn patterns from data.

Deep learning is a subset of machine learning in which multilayered neural networks learn from vast amounts of data.

History of AI

AI has experienced cycles of AI winters and AI booms.

AI solutions include speech recognition, computer vision, assisted medical diagnosis, robotics, and others.

Modern AI

Factors that have contributed to the current state of Machine Learning are: bigger data sets, faster computers, open source packages, and a wide range of neural network architectures.

Machine Learning Workflow

The machine learning workflow consists of:

* Problem statement
* Data collection
* Data exploration and preprocessing
* Modeling
* Validation
* Decision Making and Deployment

This is a summary of the common taxonomy for data in open source packages for Machine Learning:

* target: category or value you are trying to predict
* features: explanatory variables used for prediction
* example: an observation or single data point within the data
* label: the value of the target for a single data point

retrieving data:

reading CSV files:

* comma-separted files consist of rows of data, sperated by commas.
* In pandas, CSV files can typically be read using just a few lines of code:

Import pandas as pd

Filepath = ‘data/iris\_data.csv’

#import the data

Data = pd.read\_csv(filepath)

#print a few rows

Print(data.iloc[:5])

reading csv files: useful arguments:

Text

Description automatically generated

JSON files:

Javascript Objection Notation (JSON) files are a standard way to store data across platforms.

JSON files are very similar in structure to python dictionaries.

Reading json files into python:



SQL Databases:

Structured Query Language(SQL) represents a set of relational databases with fixed schemas.

There are many types of SQL databases, which function similarly with some subtle differences in syntax.

e.g of sql databases:

- Microsoft sql database

- postgres

- mysql

- aws redshift

- oracle db

- db2 family

Reading sql data:

Using sqlite23:

Text

Description automatically generated

NoSQl Databases:

Not-only SQL(NoSQL) databases are not relational, vary more in structure.

Depending on application, may perform more quickly or reduce technical overhead.

Most NoSQL databases store data in JSON format.

e.g of nosql db:

document databases: mongoDB, CouchDB

key value stores: riak, Voldemort, redis

graph databases: neo4j, hypergraph -> linkedin 1st 2nd 3rd level connection

wide-column stores: Cassandra, HBase

nosql



Using pymongo module to read files in mongoDB

Using conda databases.

Data is read into pandas by combining a query with this connection

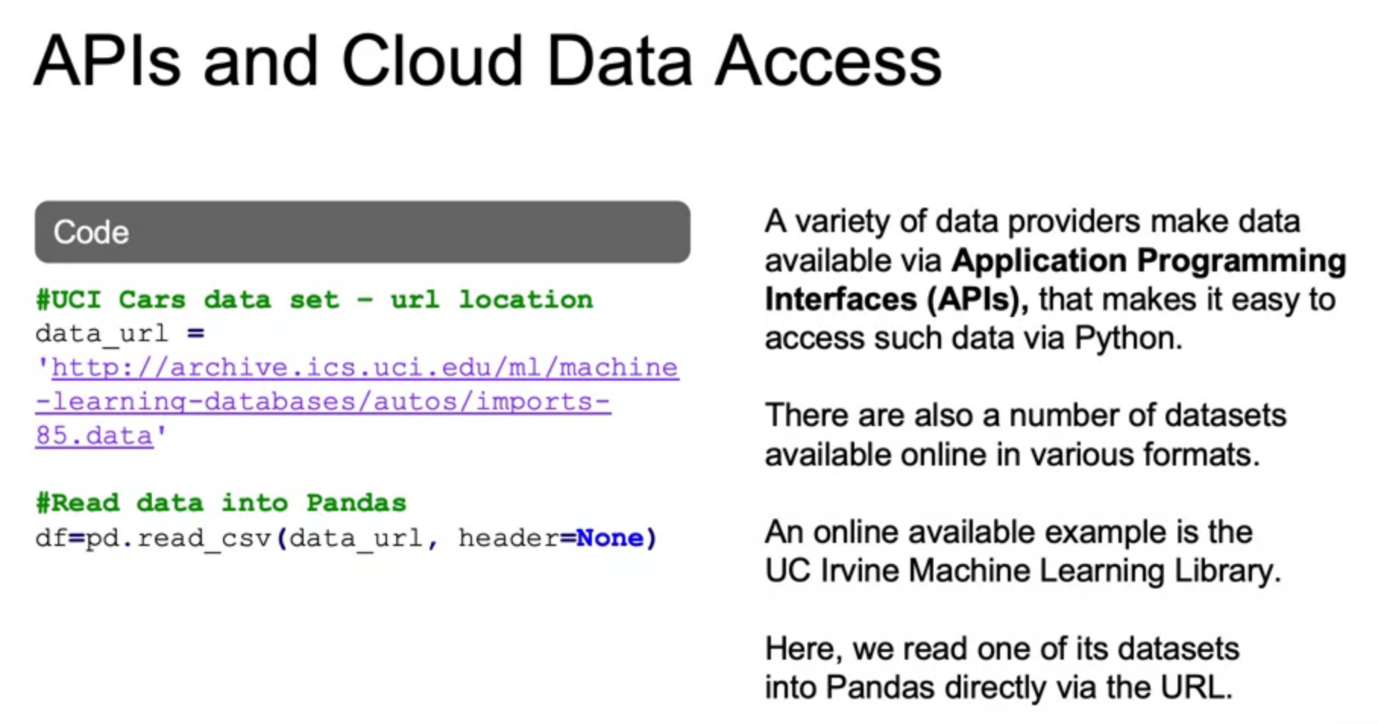
From collection name

Then, a query should be replaced with a mongoDB query string or {} to select all.

Cursor -> generator object with all json files

Once a list of python dict is obtained and is passed in pandas dataframe.

APIs and cloud data access.



Reading .txt file into pandas dataframe:

Pandas.read\_csv(‘data.txt’, header=None, sep=’’)

**Data cleaning**

Why is data cleaning important?

Decisions and analytics are increasingly driven by data and models

Key aspects of machine learning workflow depend on cleaned data:

* Observations: an instance of the data(usually a point or row in a dataset)

if a row is not clean, we are misrepreasenting our target and featues

* Must make sure all are labelled approprialtely. If mislabel, mislead model
* Algoithms: algo learned assuming data represent the real world
* Features: info we have for each observation.
* Model: model assuming actual data representing the real world
* Messy data can leda to garbage in and garbage out effect and unreliable outcoms
* Therefore we must ensure our data is cleaned

Main problems companies face:

* Lack of data
* Too much data: data eng problem, therefore must relook through
* Bad data

Therefore must ensure data is ready for use.

How can data be messy:

* Duplicate or unnecessary data.
* Inconsistent text and typos
* Missing data
* Outliers
* Data sourcing issues
* Multiple systems
* Different database types
* On premises, in cloud, etc

How to work with duplicated data:

* Pay attention to duplicate data and research why there are multiple values
* Must look into features u r bringing on and filter data as necessary but
* We should not filter too much as features can be later used.

**Best practices with handling missing values and outliers**

Policies for missing data:

* Remove the data: remove the row(s) entirely
* Impute the data: replace with substituted values. Fill in the missing data with the most common value, the average value, etc
* Mask the data: create a category for missing values.

What are the pros and cons for each of these approaches?

Remove the data:

* Models will not accept blank values.

Pros: quickly clean dataset

Cons: lose too much information

Impute the data:

Pros: do not lose full rows or columns

Cons: add another level of uncertainty. As we think of what the values should have been for missing values

Mask the data:

Pros: do not lose rows/columns which may be important to our model

Cons: add another level of uncertainty to our model as it is now based of the assumption that all missing values are alike

**Outliers:**

An outlier is an observation in data that is distant from most other observations

Typically, these observations are aberrations and do not accurately represent the phenomenon we are trying to explain through the model.

If we do not identify and deal with outliers, they can have a significant impact on the model.

It is important to remember that some outliers are informative and provide insights into the data.

For example sales: if outlier is between 1 and 50 and u get a value of 3000, then it can have significant impact on ur model

But we cannot just throw off the 3000. It could give us an indications of why it happened.

How to find outliers?

* Use plots: histogram, density plot, box plot.
* Staticstics: interquartile range, std deviation
* Residuals: standardized, deleted, studentized

How to detect outlirs: plots:

#plot a history and desntiy pplay

Sns.distplot(data, bins=20);

#plot a boxplt

Sns.boxplt(data);



Detecting outliers: residuals:

Residuals(Differences between actual and predicted values of the outcome variable)

It represents model failure.

Approaches to calculating residuals:

* Standardized: residual divided by standard error
* Deleted: residual from fitting model on all data excluding current observation
* Studentized: deleted residuals divided by residual standard error( based on all data or all data excluding current observation)

Once we detect an outlier:

* Remove them
* Assign the mean/medium value.
* Transform the variable.
* Predict what that value should have been
* Using similar observations to predict likely values
* Using regression
* Keep them but focus on models that are resistant to outliers

**Exploratory data analysis**

Exploratory data analysis is an approach to analyzing data sets to summarize their main characteristics, often with visual methods.

Why is EDA useful?

* Eda allows us to get an initial feel for the data.
* This lets us determine if the data makes sense or if further cleaning or more data is needed.
* Eda helps to identify patterns and trends in the data. (these can be just as important as findings from modelling).

Techniques for eda?

* Summary statistics: average, median, mix, max, correlations etc
* Visualizations: histograms, scatter plots, box plots etc

Tools for eda:

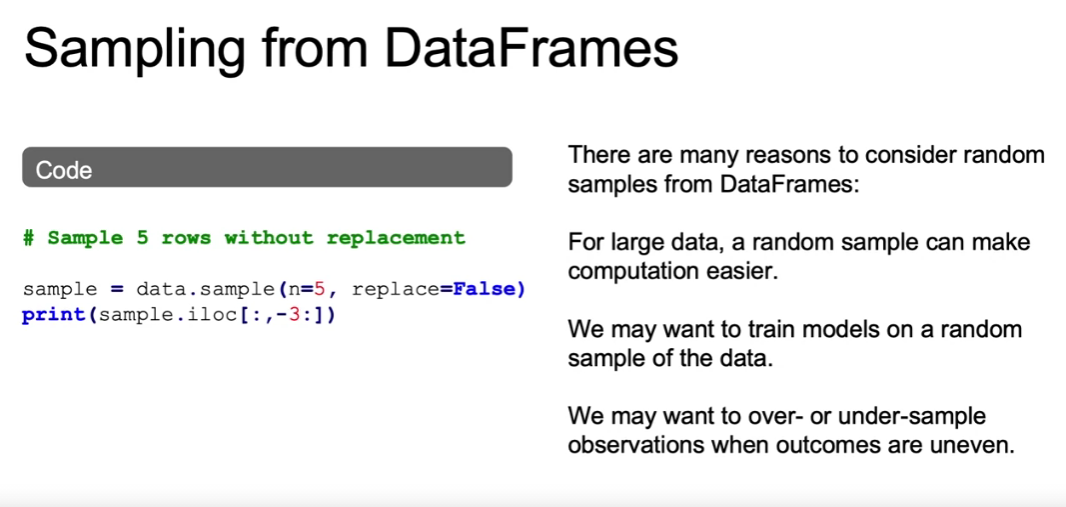
* Data wrangling: pandas
* Visualizations: matplotlib, seaborn

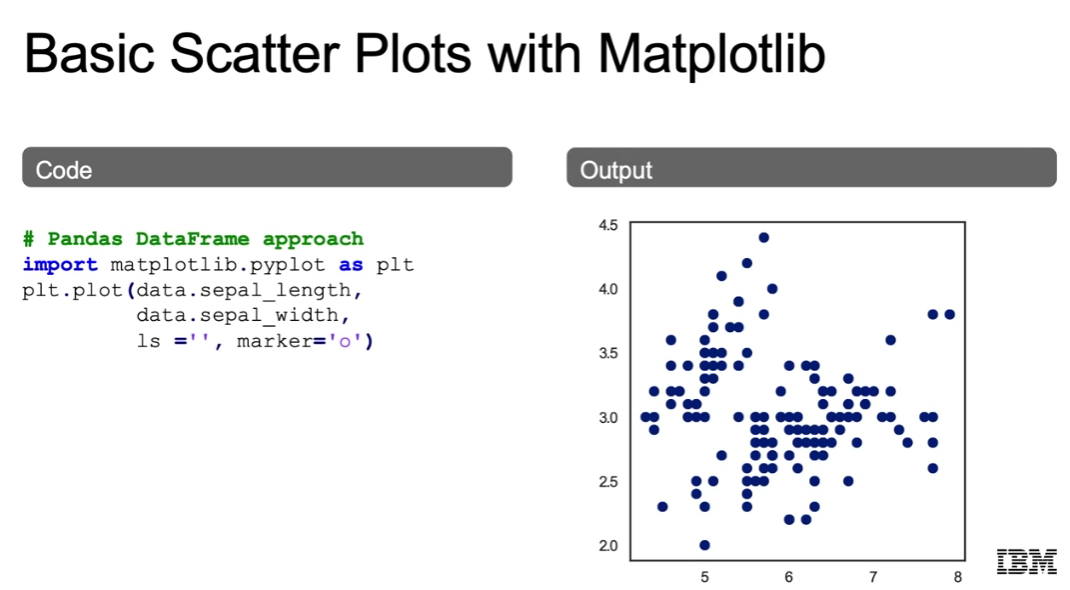
Eda: job applicant summary statistics:

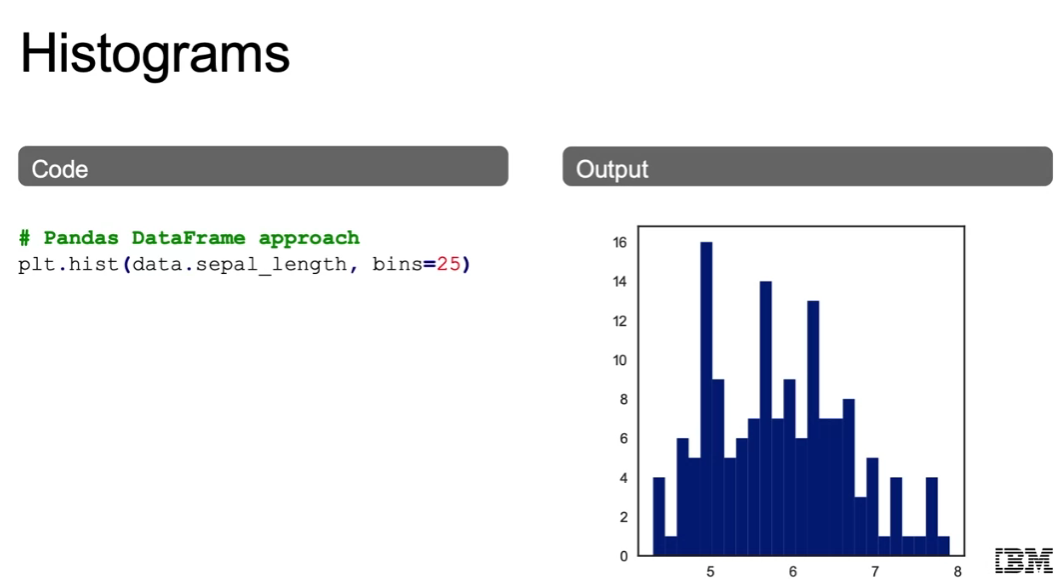
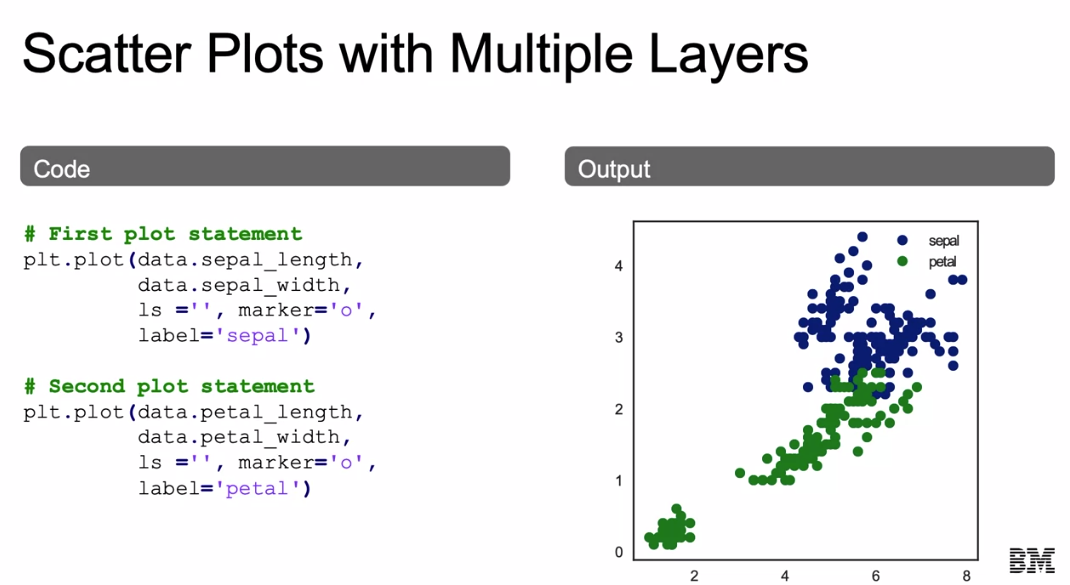
Suppose we want to examine characteristics of job applicants:

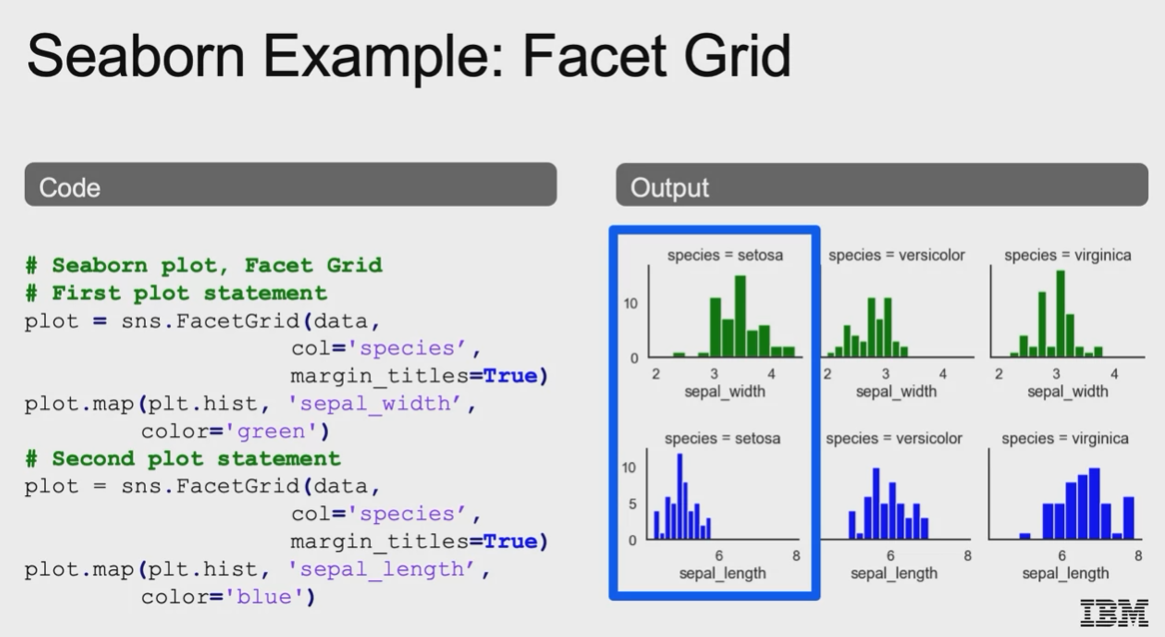
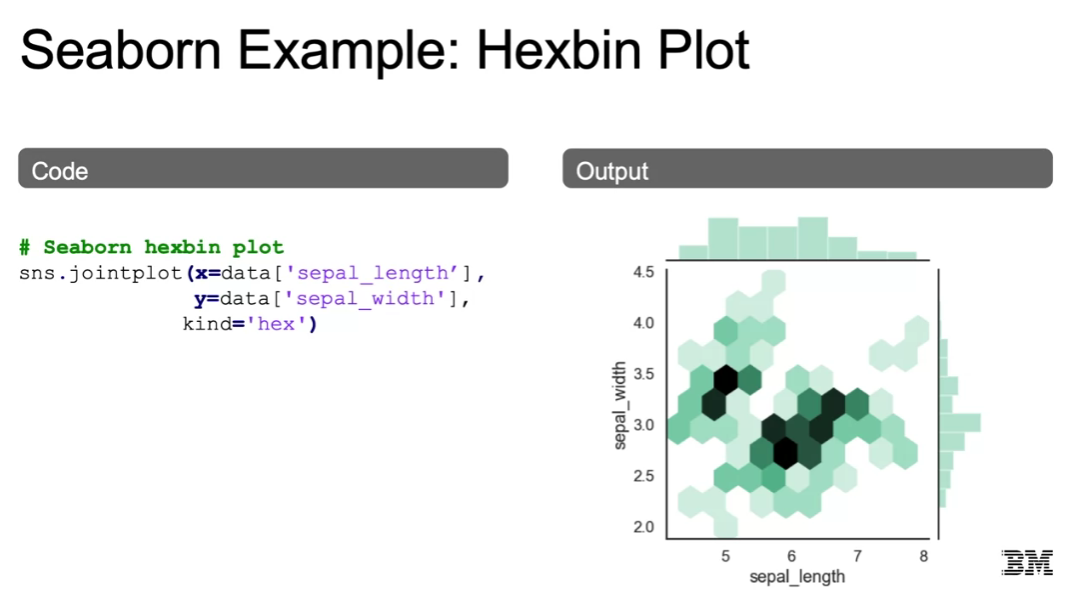
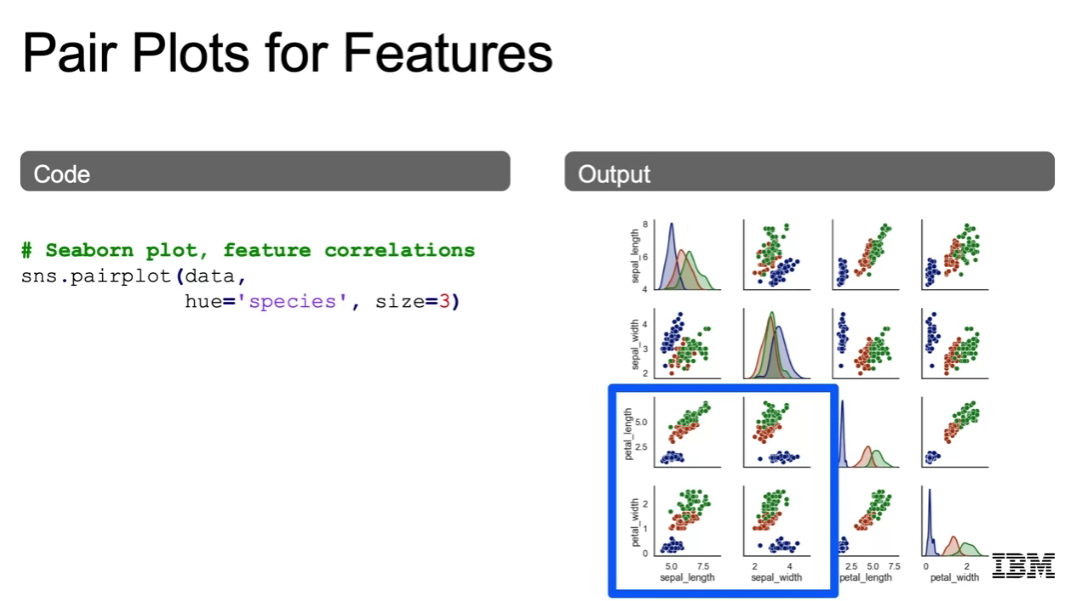
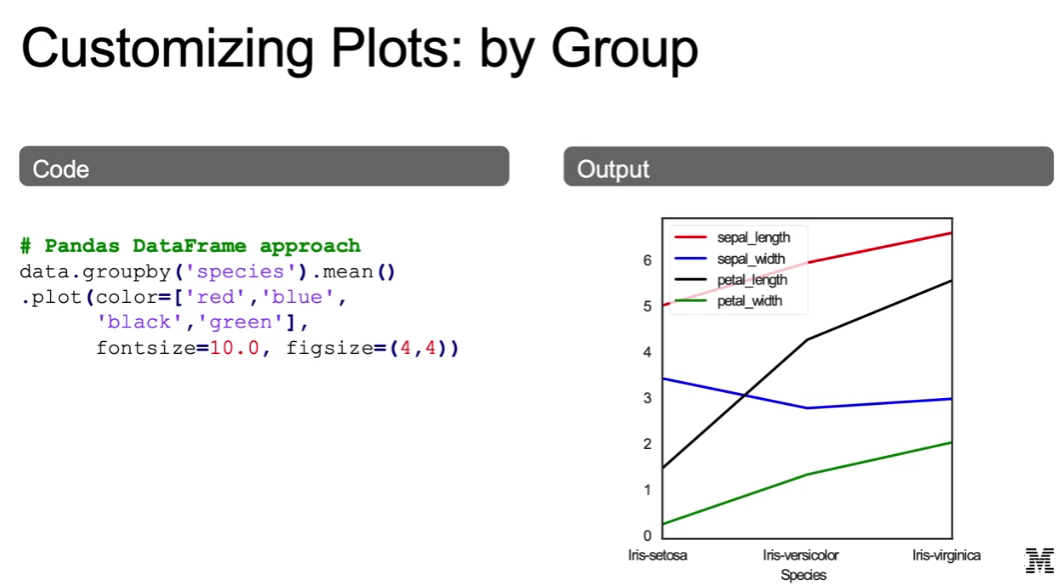
We can take a look at the:

* Avg: we could look at the average of all interview scores(perhaps by city or job function)
* Max: we could look at most common words applicants use in application materials
* Correlations: we could look at the correlations between technical assessments and years experience (perhaps by type of experience)









**Feature engineering and variable transformation**

Transforming data: background

Models used in machine learning workflows often make assumptions about the data.

A common example is the **linear regression model.**

This assumes a linear relationship between observations and target (outcome) variables.

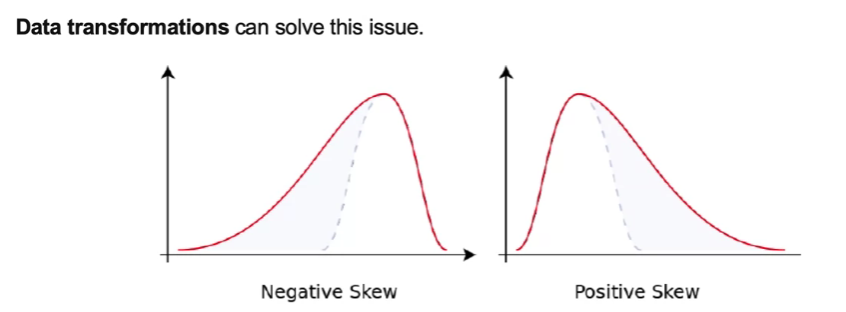
An example of a linear model relating (feature) variables x1 and x2 with target (label) variable y, is:

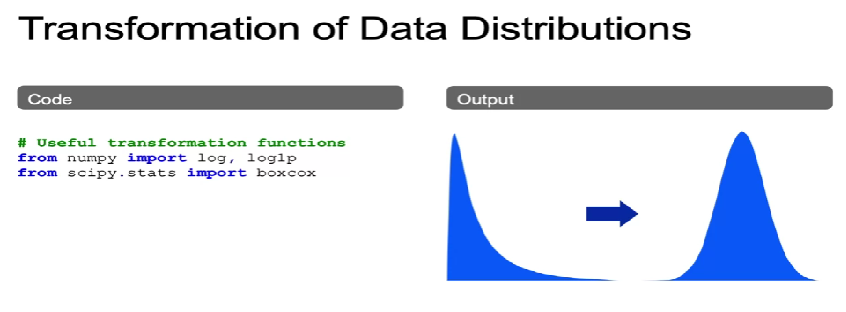
Here, represent the model’s parameters.

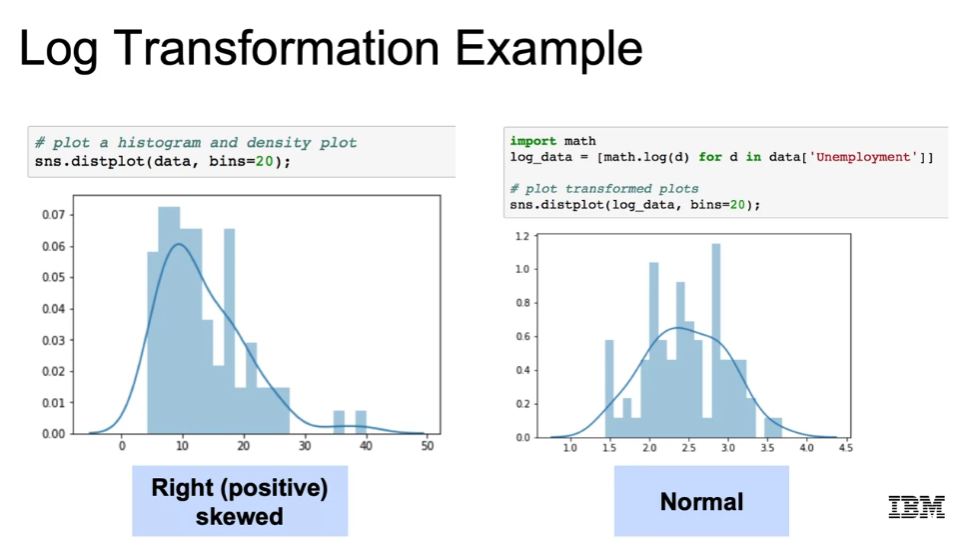
Transformation of data distributions.

Predictions from linear regression models assume residuals are normally distributed.

Features and predicted data are often skewed (distorted away from the center).







Transformations: log features:

Log transformations can be useful for linear regression.

When underlying raw data may not have a linear relationship.

Resulting algo will be a linear regression since outcome is a linear combinations of the features.

Just the features has been now transoformed by doing feature engineering, variable transformation, now we can input those into the linear regression.

The linear regression model involves linear combinations of features.

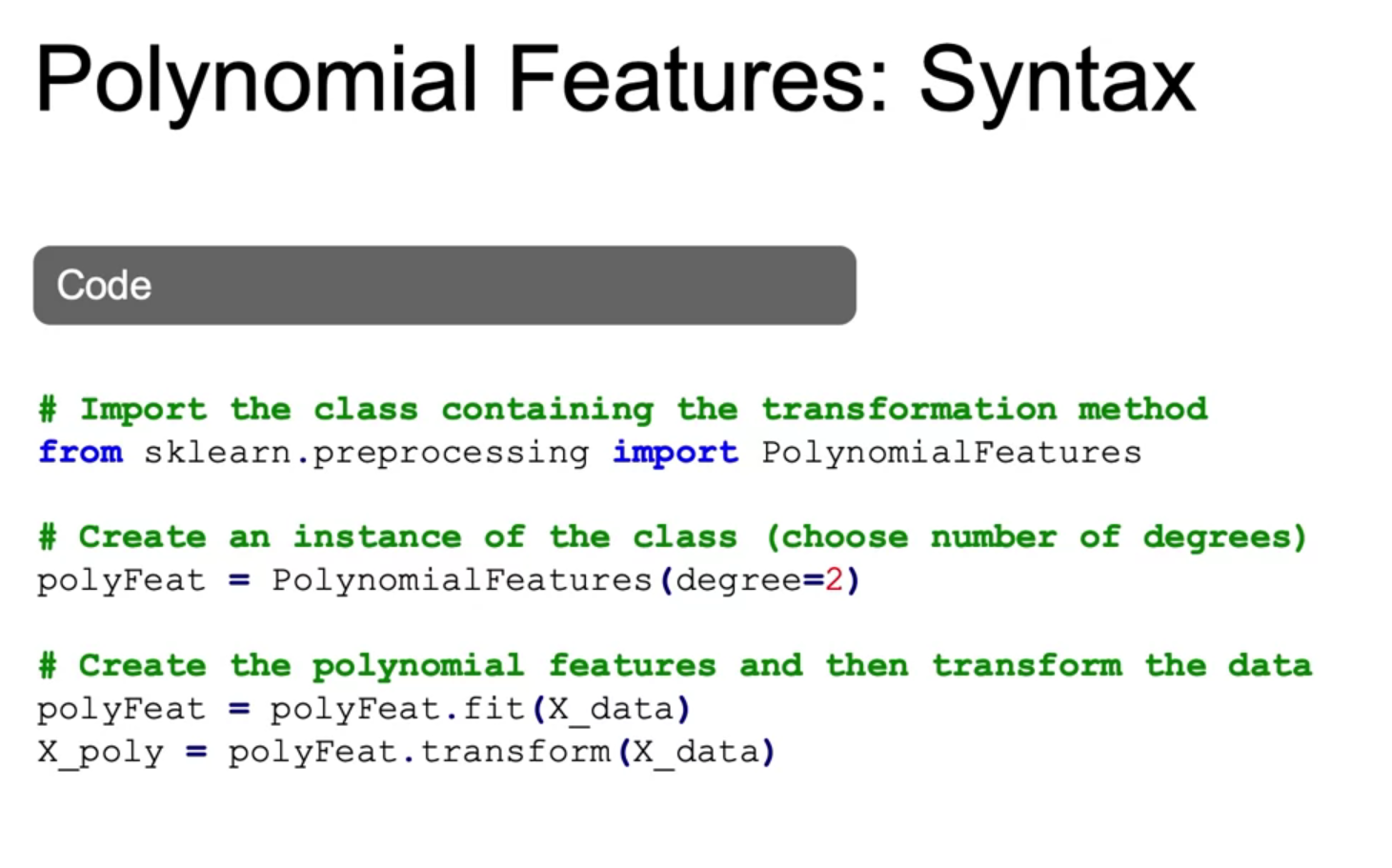
New feature is log of x instead of x.

Transformations: polynomial features.

We can estimate higher-order relationships in this data by adding polynomial featues.

This allows us to see the same ‘linear’ model.

Changing our features but maintaining a linear model.



Variable selection: background.

Variable selection involves choosing the set of features to include in the model.

Variables must often be transformed before they can be included in models.

In addition to log and polynomial transformations, this can involve:

* Encoding: converting non-numeric features to numeric features.
* Scaling: converting the scale of numeric data so they are comparable.

The appropriate method of scaling or encoding depends on the type of feature.

Feature encoding: types of features:

Encoding is often applied to categorial features that take non numeric values.

2 primary types:

* Nominal: categorical variables takes values in unordered categories. (e.g: red, blue, green; true, false).
* Ordinal: categorical variables takes values in ordered categories. (e.g: high, medium, low).

Feature encoding: approaches.

There are several common approaches to encoding variables:

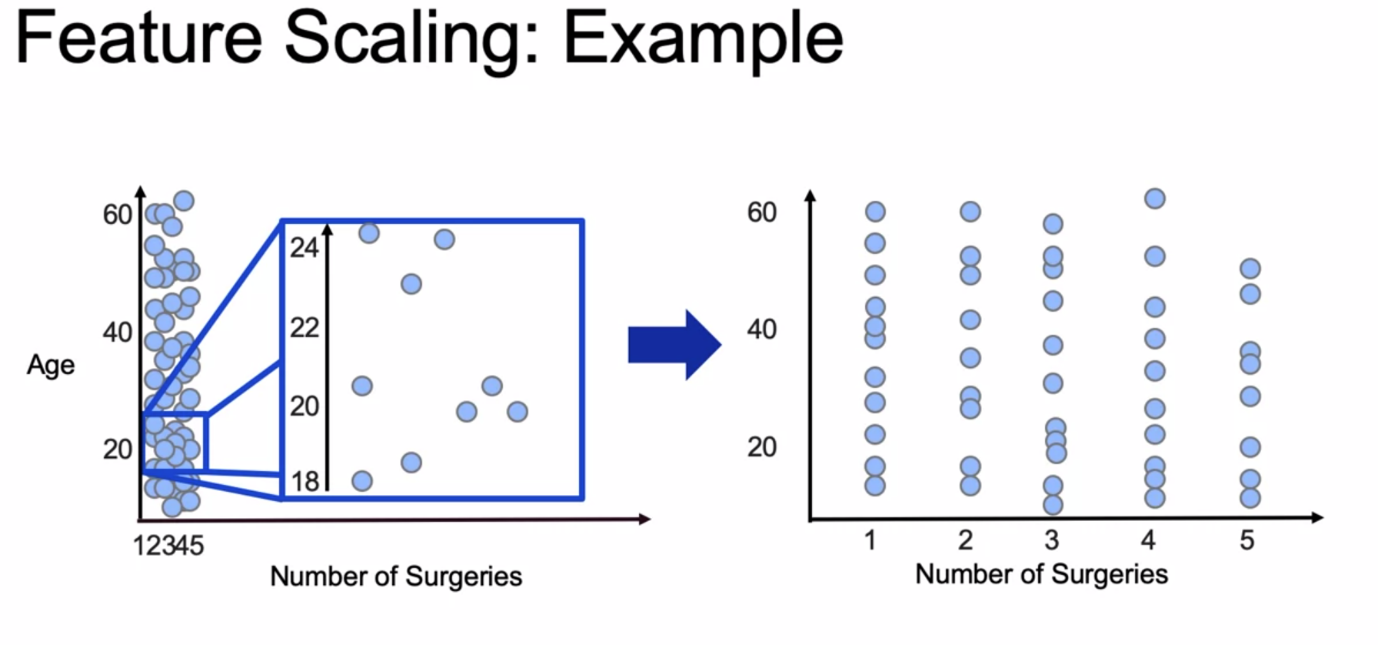
* Binary encoding: converts variable to either 0 to 1 and is suitable for variables that take 2 possible values (e.g: true or false).
* One-hot encoding: converts variable that take multiple values into binary (0,1) variables, one for each category. This creates several new variables.
* Ordinal encoding: involves converting ordered categories to numerical values, usually by creating one variable that takes integer equal to the number of categories. (e.g: 0,1,2,3).

Feature scaling: background.

Feature scaling involves adjusting a variable’s scale.

This allows comparison of variables with different scales.

Different continuous (numeric) fetuers often have different scales.



Feature scaling: approaches

There are many approaches to scaling features.

Some of the most common approaches include:

* Standard scaling: converts features to standard normal variables.

(by substracting the mean and dividing by the standard error).

* Min-max scaling: converts variables to continuous variables in the (0,1) interval by mapping min values to 0 and max values to 1.

This type of scaling is sensitive to outliers.

* Robust scaling: is similar to min-max scaling but instead maps the intequartile range (the 75th percentile value minus the 25th percentile value) to (0,1).

The means the variable itslf takes values outside of the (0,1) interval.

Common variable transformations.

|  |  |
| --- | --- |
| Feature type | Transformation |
| Continuous: numerical values | * Standard, mix-max, robust scaling |
| *Useful functions for scaling variables:*  *From sklearn.preprocessing import StandardScaler, MinMaxScaler, RobustScaler* |  |
| Nominal: categorical, unordered features(true or false) | * Binary, one hot encoding (0,1) |
| *Useful functions for encoding categorical variables:*  *From sklearn.preprocessing import LabelEncoder, LabelBinarizer, OneHotEncoder*  *From pandas import get\_Dumies* |  |
| Ordinal: categorical, ordered features  (movie ratings) | * Ordinal encoding (0,1,2,3) |
| *Useful functions for encoding ordinal variables:*  *From sklearn.feature\_extraction import DictVectorizer*  *From sklearn.preprocessing import OrdinalEncoder* |  |

(True/False) Feature scaling allows better interpretation of distance-based approaches.

True.

## Summary/Review

## Retrieving Data

You can retrieve data from multiple sources:

* SQL databases
* NoSQL databases
* APIs
* Cloud data sources

The two most common formats for delimited data flat files are comma separated (csv) and tab separated (tsv). It is also possible to use special characters as separators.

SQL represents a set of relational databases with fixed schemas.

## Reading in Database Files

The steps to read in a database file using the sqlite library are:

* create a path variable that references the path to your database
* create a connection variable that references the connection to your database
* create a query variable that contains the SQL query that reads in the data table from your database
* create an observations variable to assign the read\_sql functions from pandas package
* create a tables variable to read in the data from the table sqlite\_master

JSON files are a standard way to store data across platforms. Their structure is similar to Python dictionaries.

NoSQL databases are not relational and vary more in structure. Most NoSQL databases store data in JSON format.

## Data Cleaning

Data Cleaning is important because messy data will lead to unreliable outcomes. Some common issues that make data messy are: duplicate or unnecessary data, inconsistent data and typos, missing data, outliers, and data source issues.

You can identify duplicate or unnecessary dataCommon policies to deal with missing data are: remove a row with missing columns, impute the missing data, and mask the data by creating a category for missing values.

Common methods to find outliers are: through plots, statistics, or residuals.

Common policies to deal with outliers are: remove outliers, impute them, use a variable transformation, or use a model that is resistant to outliers.

## Exploratory Data Analysis

EDA is an approach to analyzing data sets that summarizes their main characteristics, often using visual methods. It helps you determine if the data is usable as-is, or if it needs further data cleaning.

EDA is also important in the process of identifying patterns, observing trends, and formulating hypothesis.

Common summary statistics for EDA include finding summary statistics and producing visualizations.

## Feature Engineering and Variable Transformation

Transforming variables helps to meet the assumptions of statistical models. A concrete example is a linear regression, in which you may transform a predictor variable such that it has a linear relation with a target variable.

Common variable transformations are: calculating log transformations and polynomial features, encoding a categorical variable, and scaling a variable.

Statistics and hypothesis testing:

