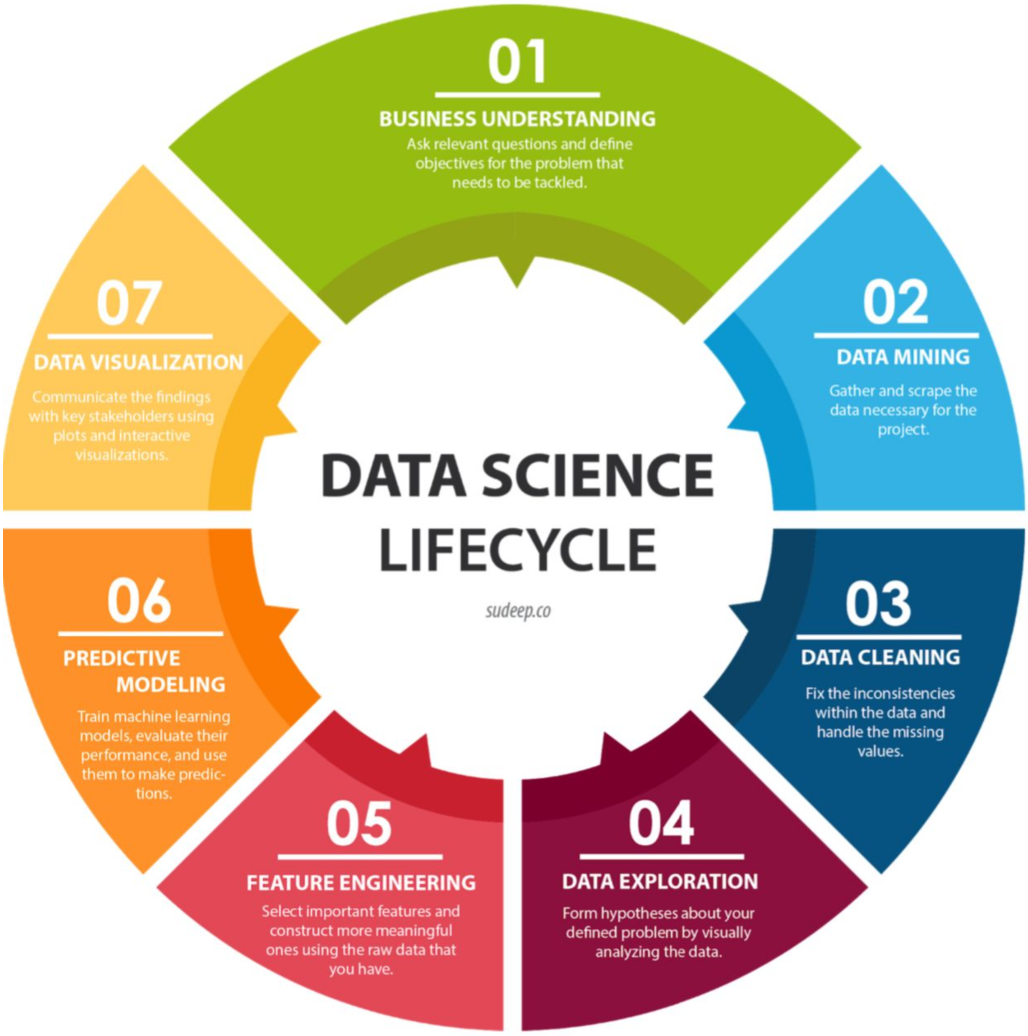
Project 2: Milestones, Parts, Lifecycle

Reminder: Data Science Lifecycle


DATA SCIENCE LIFECYCLE
sudeep.co
01
BUSINESS UNDERSTANDING
Ask relevant questions and define objectives for the problem that needs to be tackled.
02
DATA MINING
Gather and scrape the data necessary for the project.
03
DATA CLEANING
Fix the inconsistencies within the data and handle the missing values.
04
DATA EXPLORATION
Form hypotheses about your defined problem by visually analyzing the data.
05
FEATURE ENGINEERING
Select important features and construct more meaningful ones using the raw data that you have.
06
PREDICTIVE MODELING
Train machine learning models, evaluate their performance, and use them to make predictions.
07
DATA VISUALIZATION
Communicate the findings with key stakeholders using plots and interactive visualizations.

Part 0: Framework

- Timeframe, time schedule of the project
- Resources (people, finances, energy, prioritisation)
- Necessary computational resources (computers, AWS, ..)
- Metric of success for the project

Part 1: Business Understanding

- What is the business domain?
- Business setting
- Problem Statement, Regression and/or Classification problem
- Carefully read data dictionary (if not present -> request)
- Business Value
- What are business interests and potential benefits?
- Why do we care?
- What are relevant questions, which could be answered by Data Science?
- What insights can be gained, what are interesting questions?
- How's
    - Ask business stakeholder
    - Own research
        - Google
        - Google Scholar
        - Company Websites
        - Competitor Websites
        - Company intranet/wiki (if available)
        - Books
        - What other Data Scientists have done in the same domain (e.g. GitHub)

Part 2: Data Mining

- Data provided (e.g. csv files, json, hdf5)
- Databases (SQL, PySpark)
- Webscraping

Part 3 and 4: Data Cleaning and Data Exploration

- Explore Dataset

- What is the target variable?
- What are the feature variables?
- Numerical or categorical variables?
- .describe(), .info(), .head(), .tail(), ….
- Do Jupyter Notebook search for Data Cleaning/Exploration techniques, methods:
    - Plotting (scatter plots, histograms, boxplots, barplots, distplots, lmplots, regplots, factorplots, …)
    - maybe already here: correlation matrix, heatmap, get insight into what features are important (form hypothesis)
    - Change variable dtypes
    - Detect/count missing values (package missingno), and outliers
    - Imputing missing values (ffill, bfill, mean, drop values, drop variables, ...), drop outliers

Additions:

- data loaded correctly (e.g. zipcodes started with 0s)
- data types, numerical, categorical
- renaming (removing whitespace, special characters, …)
- multiple datasets? combine datasets, truncate (redundancies), modify, resample, same scale (e.g. time zones, currencies, naming conventions)
- missing values (-999, NaN, 0, …)
- handle missing data
    - impute (mean, median, interpolation, majority vote, with educated guess, values that make sense, e.g.)
    - analyse connections between columns, investigate if data can be inferred (e.g. sqft_basement), before deleting
    - delete row, observations
    - delete column, feature
    - create dummy for missing (0=missing, 1=present)
    - talk to business (reason why data is missing)
- checking for duplicates (row/observation duplicated), only one feature duplicated (reasonable?!, ID, e.g. house sold more than

once), or all features (essentially copies), depending on features (observations actually can duplicate, by random chance)

- ID column? denoting unique observations (if IDed correctly)
- check for inconsistencies (e.g. house with 33 bathrooms)
- check for relevancy of data with respect to problem statement (e.g. fictitious drug in drug survey)
- Understanding target, and features (column names, predictors, independent variables)
- Understanding/looking at rows/observations for each column/feature
- Shape/size of the dataset, how big is it
    - if large: random sample? (after cleaning)
    - features: PCA (part feature engineering)
    - if too small or imbalanced: ask/look for more data, similar data from different data sources, synthetic data generation

- EDA:
- Summary Statistics (.describe())
- Visualise numerical features, distributions (matplotlib, seaborn)
- Categoricals: barplots, pieplots
- Explore relation of target variable with predictor variables (scatterplots, …)
- Check for outliers
- Check for normality
- Make advanced plots to understand interrelations between one or more features and the target variable (groupbys, 2D hists, …)

Part 4: Feature Engineering

- Transforming skewed continuous variables to more normally distributed values
- Normalise or standardise numerical features (MinMaxScaler, StandardScaler)
- Dimensionality reduction (e.g. PCA)

- get_dummies for categorical variables
- Store final cleaned dataset into new .csv file

- Extract features from existing features (e.g. from ‘name’ extract ‘gender’)
- Create new categories
    - binning numerical values into categories
    - regrouping categories into new categories
- Impute values for features (this can be a part of Data Cleaning as well as Feature Engineering)
- Assign weights to features denoting feature importance

Part 6: Predictive Modelling

- Correlation matrix, pair plots
- Balancing the data (classification)
- Train, test, split the data
- What is the Naive Predictor Performance? (dummy benchmark)

- Select Statistical Learning Models:
    - (Supervised Learning Models:)
    - Regression: Linear, Multiple Linear, Polynomial
    - Lasso (Feature Selection), Ridge Regression
    - Logistic Regression
    - KNN, k nearest neighbour
    - Naive Bayes
    - Decision Trees
    - Random Forests (Ensemble of many, small Decision Trees)
    - Ensemble Methods:
        - Bagging (reduce Variance)
        - Boosting (reduce Bias)

        - Stacking (Max Vote, Avg, Weighted Avg)
    - PCA (this is feature engineering?!)
    - Support Vector Machine

- train, predict modelling pipeline
- Model Evaluation:
    - Test scores
    - k-fold cross validation
    - Evaluation metrics
    - Confusion matrix
- Model Tuning:
    - (Using different distance metrics?!)
    - GridSearchCV
    - RandomizedSearchCV
- Final Model Evaluation on test dataset
- Feature Importances

Additional:

- Pick models to use that are appropriate for the current challenge, and available resources (kaggle stacking sometimes not feasible)
- Train-test-split
- Balancing the dataset
- What is the baseline model performance metric which should be outperformed by a model
- Evaluate model performance (confusion matrix, …)
- The pre-determined metric of success from business understanding determines which of the evaluation metrics are most appropriate for the model evaluation
- compare models (at least three) against evaluation metrics
- Identifying most important features, predictors across all models created
- (hypothesis testing for marketing, sales, healthcare)
- visualise model based on most important features (predicted against actual data)

Part 7: Data Visualisation

- Technical:
    - Distributions, Target-Feature Dependencies...
    - Regarding model performance
- Business
    - Try to make insightful and compelling visualisations regarding the domain specific business questions