Data Science Concepts

Artificial Intellegence(AI) - Programming of computers to perform human tasks
- older + Larger than ML

Intersection of ML + AI

- Once a ML program collects info + supplies outputs, AI can be used to create a new algo so that a machine learning program can update its algo

*Deep Learning (DL) : Subset of ML + AI that uses neural network algos w/ 3 or more hidden layers to solve problems.

* Structured vs. Unstructured Data vs. Big Data *

Big Data: • Massive volume of both structured/unstructured data that's so large its difficult to process w/ traditional database *

*4 "V's" of Data : • Volume • Variety • Velocity • Veracity (relevancy)

Unstructured Data : • Does not fit a predefined data model
ex) Images, emails, videos, audio

Structured : can fit into a predefined data model

Building Models

- Turns ~~data~~ into actionable information

Data Modeling

Parameter

- A configuration variable that is internal to the model
- values are estimated or learned from data when the model is being "fit"

Hyperparameter

- "You create hyper parameters"
- Configuration that is external to the specific model
- Value Not estimated from data
- Used to customize the algo for a given problem so that it can more effectively estimate model parameters

Features

- Attributes, X, predictor, or independent variables that are used in the model, the "knowns"

Input variables to a model

Feature Engineering

- Using domain knowledge of the data to create new features

ex) number_of_bed_or_bath_rooms

Data Planning

Goal:
- Clearly define your goal (write it down)
  - Measures of success, and plans on how to achieve that

Deliverable:
- Documentation of your goal
- If you haven't defined success, you won't know when you have achieved it

How You Get There: By answering questions about the final product & formulating or identifying any initial hypotheses.

Acquisistion

AKA: Data gathering, Data Import, Data Wrangling (Acquisistion + Prep)

Goal: Create a path from original data sources to the enviornment in which you will work w/ the data

Deliverable: A file, acquire.py, that contains the function(s) needed to reproduce the acquisition of data

How to get There: SQL: clean-up, integration, aggregation or other manipulation of data in the SQL Enviornment

Pylib: pandas

- May use Spark and/or Hive when acquiring data from a distributed enviornment such as HDFS.
- Examples of source types:
  - RDBMS
  - HDFS
  - Static local flat files (CSV, txt, xls

Data Preperation

(AKA: Data Data Wrangling)

The Goal: Have Data, split into 3 samples:

1) Train
2) Validate
3) Test

- In a format that can easily be explored, analyzed, and visualized.
- Helps understand the generality of the model

Deliverable:
- A file, prep.py, that contains the function(s) necessary to reproduce the prep of the data
- Resulting dataframes should be 3 samples:
  1) Dataframe for training the algorithms
  2) Dataframe for validating the models
  3) Dataframe for testing the best performing model

How to get There
- Python libraries: pandas, matplotlib, Seaborn
- Use pandas to perform task such as handling null values, outliers, normalizing text, binning of data, changing data types, etc.
- Use matplotlib or seaborn to plot distributions of numeric attributes and target.
- Use SciKit-learn to split the data into train and test samples

Exploration and Pre Processing

* where you find most of your takeaways *

* we do this on the training data *

AKA: Exploratory Analysis/Visualization, Feature Engineering, Feature Selection

Goal: To discover features that have the largest impact on the target variable

ie. (provide the most information gain; drive the outcome)

Deliverable: A file, preprocess.py, that contains the function(s) needed to reproduce the pre-processing of ~~the data~~.

- DataFrame resulting from these functions should be pre-processed (ready to be used in modeling)
  - Reduced to features
  - Features are in numeric form
  - No missing values
  - Continuous and/or ordered values are scaled to be unitless

How to get There: Python libs: Pandas, statsmodels, scipy, numpy, matplotlib, seaborn, scikit-learn

- Perform Statistical testing to understand correlations, significant differences in variable, variable interdependencies, etc
- Create visualizations
- Use domain knowledge and/or information
- Remove features that are noisy, provide no value, redundant
- Use scikit-learns pre-processing algorithms to turn attributes into features

Data Modeling

Goal: Create a robust and generalizable model that is a mapping between features and a target outcome.

Deliverable: A file, model.py, that contains functions for training the model (fit), predicting the target on new data, and evaluating results

How to get There: Py libs: Scikit-learn

- Identify regression, classification, cross validation, and/or other algorithms that are appropriate
- Build Your Model:
  1) Create the model object
  2) Fit the model to your training or in-sample observations
  3) Predict target value on your training observations
  4) Evaluate results on the in-sample predictions
  5) Repeat as necessary w/ other algorithms or hyperparameters
  6) Using the best performing model, predict on test, out-of-sample, observations.
  7) Evaluate results on the out-of-sample predictions.

Data Delivery

Goal: To enable others to use what you have learned or developed through all the previous stages

Deliverable:

- Could be various types:
  - A pipeline.py file that takes new observations from acquisition to prediction using the previously built functions
  - A fully developed model
  - A reproducible report and/or presentation w/ recommendations of actions to take based on original project goals.
  - Predictions made on a specific set of observations
  - A dashboard for observing/monitoring the key drivers, or features, of the target variable

How to Get There

- Python Sklearn's pipeline method
- Tableau for creating a report, presentation, story, or dashboard.
- Jupyter notebook for creating a report or a framework to reproduce your research, e.g.
- Flask to build a web server that provides a gateway to our models predictions

Command Line

- Command Line Interface (CLI) - Text-based interface
  - Bourne-Again Shell (BASH)
- Only interprets text commands

terminal - Graphical interface to the shell
Shell - Program that reads/produces your output
- Controls interaction w/ operating system

Prompt
ex) Users-Macbook-Air:~user$

- Issue commands: Return/Enter

BASH - Software that powers the shell

- Find version of BASH

IN: bash --version

$ - Logged in as a regular user

Listing Files: $ ls

* ls → folder shows folder were looking 4

Specify a Directory: ex) $ ls Documents

Flags: - modifiers
- Can change the way a command behaves

ex) -a: all files
-l: long format
-h: human readable output

ex) $ ls -l

Command Line

- pwd - Where am I? / Current directory (Absolute)
  "print working directory"

File Paths

- Specifies a unique location in a file system
  ex) Documents/Secret_Diary/My.Crush.txt

"/" - Seperates one directory from another in a path
- By itself, it represents the "root directory"

Relative Path vs. Absolute Path

Relative: the gas station is two blocks to the left

Absolute: The full address or lat/long of that exact gas station

"~" - Points to home directory

Navigating

- pwd - "You are here" Always gives Absolute Path.

  IN: $ pwd
  /Users/username

- Change Directory
  - Move around our filesystem

  IN: $ cd

  - Specify where you want to go
    ex) $ cd /Users/john

* Navigating Filesystem *

"." - Single dot, represents our current directory

".." - Double dot, represents parent directory (directory immediately above our current directory

ex) # Go up one directory
$ cd ../

# Go up two directories
$ cd ../../

# Go up three directories
$ cd ../../../

* Creating Files + Directories *

mkdir - Creates directory

touch - creates an empty file

ex) $ touch newfile.txt

ex) Creating multiple files/directories:

$ touch one two three four

$ mkdir one two three four

Moving Files

- mv - needs: 1) original file you want to move
  2) The location where you want to move it

ex) $ mv ~/newfile.txt ~/Desktop/newfile.txt

- Change Name

  $ mv current_filename.txt new_filename.txt

- ctrl-C - Interrupts programs
- help: -h
  --help
  help
- Man Pages
  - Manual pages
  - Serve as a reference for most commands

  ex) man COMMAND_NAME

  - d - go down a half page
  - U - go up a half page
  - q - to quit
  - / - to search for something

Comments : Shell ignores

. code - Launches VS Code

• grep - Searches for text in a file

ex) grep PATTERN myfile.txt

• rm - remove file / files

• find - Searches for files

• cat - Displays contents of a file

• less - Displays file content

• open - will guess which program to use to open a given file

• head / tail - View first / last few lines of a file

• pbpaste / pbcopy - shows / sets the contents of clipboard

Ctrl - a - go to beginning of the line

Ctrl - e - go to end of the line

Ctrl - k - delete from the cursor to end of the line

Ctrl - l - clear the screen

Ctrl - w - delete one word backwards

Git

- Distributed version control system.
- Formulized way to keep track of changes + allows users to view changes over time

- repository - Directory managed by git
- add - Tells git which files to look at for the next commit
- commit - creates a "save point"
- push - Uploading to git (back-up

- Initializing a Repository:

  git init - creates a new repository
  - Tells git to track files in directory

- Adding Files - Takes files from untracked → tracked
  - Stages files to be ready to add to repository
  - Will be available to commit

Commit - Makes official record of the changes

- Git Status - Runs current status

ex) git add Codeup/my_new_file.txt

GIT

- **Committing Changes**: sending a copy of backup to github

commit: git commit

or
git commit -m 'Detailed commit message'

- **Remotes**: copy of our repository in different location
  - origin (for pushing/pulling)

ex) git remote add origin git@github.com:YourUsername/your-repo-name.git

Push/Pull
- Update / retrieve code from our remote repository

Push - git push origin master
Pull - git pull origin master

.gitignore
- what files repo ignores

ex) put "ipython.log" in .gitignore folder