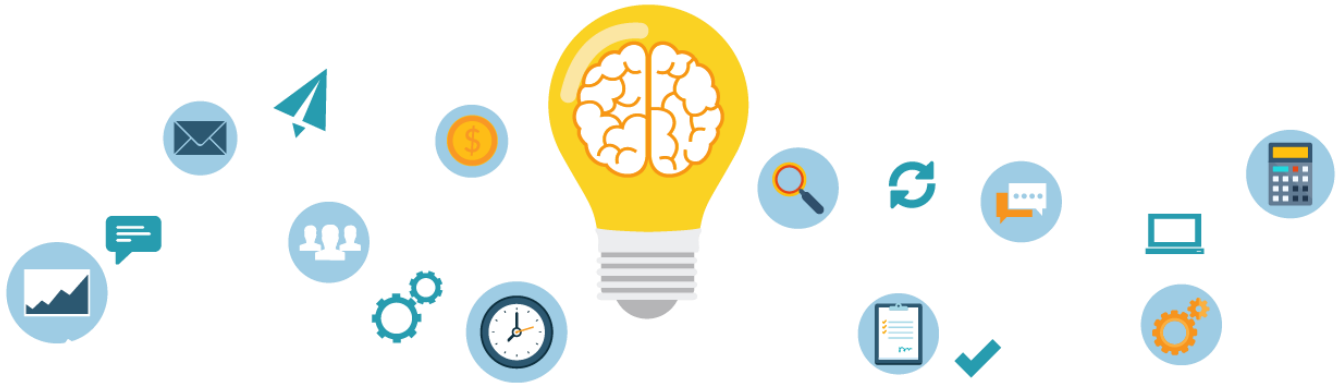
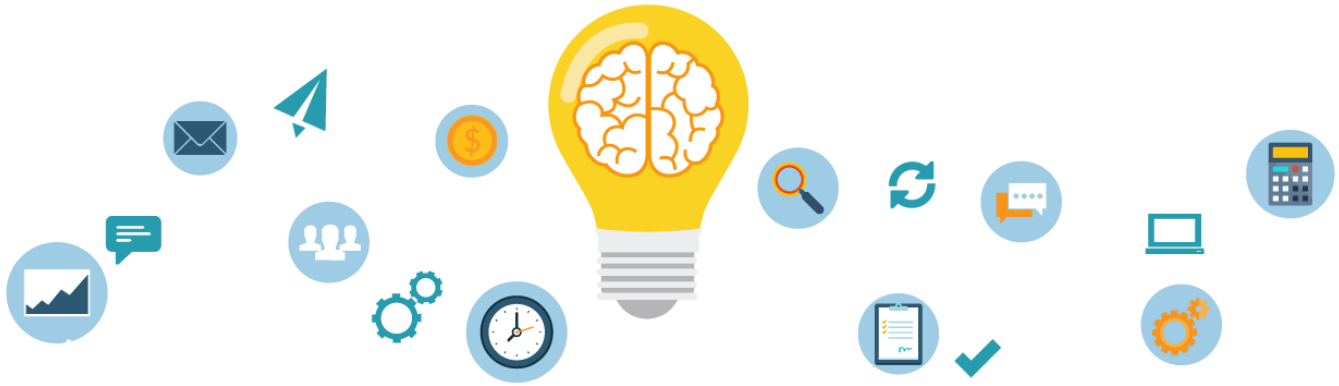


Python Programming

Day 1



Jupyter Notebook Familiarization

Python Programming

Exercises

1. Multiple variables in a 2D Plot
2. New Variables
3. Querying : Advanced filters
4. Clearing outliers and regression on plots
5. Plotting different series
6. Researching sampling methods and how to do KDE Plots
7. Creating a function simulating linear regression

Multiple variables in a 2D Plot

Plot the following variables in one graph

- num_critic_for_reviews
- IMDB score
- gross
- Steven Spielberg against others

New Variables

- Compute sales (gross – budget) and store it in the same data frame

Querying : Advanced filters

Which directors garnered the most total sales?

1. Get the top 10 directors
2. Filter the data frame for only these directors
3. Proceed to plot

Which actors garnered the most total sales?

1. We have three actor fields. For now, get only the first one.
2. Filter the data frame for only these actors
3. Proceed to plot
4. Bonus: Create a series / dictionary for the three actor fields to their sales

Clearing outliers and regression on plots

Plot sales as a function of movie_facebook_likes. Plot it as a scatterplot. Fit it with a line.

1. Create a function for the Tukey's method above.
2. Remove sales and movie_facebook_like outliers for a better understanding.
3. Add jitter.
4. Plot if there is a good linear correlation.
5. Bonus: Try a nonlinear fit, i.e. polynomial of order 2, 3, 4?

Plotting different series

Which of these genres are the most profitable? Plot their sales using different histograms, superimposed in the same axis.

- Romance
- Comedy
- Action
- Fantasy

Researching sampling methods and how to do KDE Plots

Plot a Kernel Density Estimation plot of the following variable combinations:

- Duration and Gross
- Duration and IMDB Scores

To review, for clearer plotting, you have the following options:

- Sampling - research on this
- Jittering
- Outlier removal

Preparing for Machine Learning

For this exercise, we will simulate a common algorithm, used in statistics, machine learning and beyond, linear regression

1. Create a function for z-normalization.
2. Standardize your sales variable and save it to a new variable in the same data frame.
3. Compute average actor likes, which averages the three actor's Facebook likes. Standardize this variable as well.
4. Create a function that takes (1) a scalar, (2) theta and (3) a bias variable to output a value as close as possible to gross. Call this function, predict_score.

$$score = b + \sum_j (\theta_j * x)$$

$$score = \theta_1 * average\_actor\_likes + bias$$

Preparing for Machine Learning

1. Create the RMSE function. Create a function that compares two vectors and outputs the root mean squared error / deviation.

$$\mathrm{RMSD}(\hat{\theta}) = \sqrt{\mathrm{MSE}(\hat{\theta})} = \sqrt{\mathrm{E}((\hat{\theta} - \theta)^2)}$$

2. Create the best possible thetas by brute-forcing against the RMSE function. Create predictions for your entire dataset. Compare your predictions against the score. Achieve the smallest RMSE you can.
3. Plot your best theta, bias variable against the imdb score for each movie. For a cleaner plot, you should:
    1. Compile your average_actor_likes, imdb_scores and predicted to a new dataframe
    2. Limit the bounds of your predicted ratings
    3. Use a combination of scatter plot for the independent variables / features and a line plot for the predicted score

Preparing for Machine Learning

1. Convert your hypothesis function to use more variables. Standardize your new variables.

$$score = \theta_1 * average\_actor\_likes + \theta_2 * movie\_facebook\_likes + \theta_3 * sales + bias$$

2. Compile your theta values to a new pandas dataframe which consists of the following columns:

θ_1	θ_2	θ_3	$RMSE$
0.1	0.1	0.1	10000
0.2	0.2	0.2	2000

3. Plot how each theta parameter influence the RMSE. Which one seems to be most influential?