WOMEN WHO CODE®
/data-science

Build a Data Science Project from Scratch - SESSION 1

JUNE 9
10 PM ET

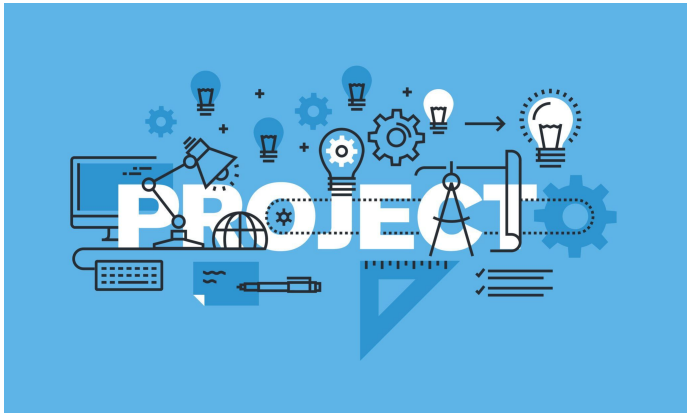
womenwhocode.com/datascience/events

Lindsey Robertson

Analyst and CRM Manager
@ iKadre

Series Agenda

1. Introduction and Overview
2. EDA and Visualisations
3. Data Wrangling and Feature Engineering
4. Baseline Model Building
5. Increased Model Complexity
6. *Bonus Session - learning outcomes!



Data

GEORGEMCINTIRE · UPDATED 6 YEARS AGO

393 | New Notebook | Download (101 kB)

Spotify Song Attributes

An attempt to build a classifier that can predict whether or not I like a song

Data Card | Code (63) | Discussion (1)

About Dataset

Context

A dataset of 2017 songs with attributes from Spotify's API. Each song is labeled "**1**" meaning I like it and "0" for songs I don't like. I used this to data to see if I could build a classifier that could predict whether or not I would like a song.

I wrote an article about the project I used this data for. It includes code on how to grab this data from the Spotipy API wrapper and the methods behind my modeling.

https://opendatascience.com/blog/a-machine-learning-deep-dive-into-my-spotify-data/

Content

Each row represents a song.

There are **16** columns. **13** of which are song attributes, one column for song name, one for artist, and a column called "target" which is the label for the song.

Usability
7.35

License
Unknown

Expected update frequency
Not specified

Session 1 Agenda

1. Data Overview - Metadata
2. Problem Statement Overview
3. Quick project requirements & setup review
4. Load the data!
5. Q & A

Data Science Problem Types

Problem Type	Description	Possible Model Solutions
Regression	Predicting continuous target variables	Linear regression, Decision Trees, Random Forest, Neural Networks
Binary Classification	Predicting two discrete class labels	Logistic Regression, Decision Trees, Random Forest, Support Vector Machines, Neural Networks
Multiclass Classification	Predicting multiple discrete class labels	k-Nearest Neighbors, Decision Trees, Random Forest, Support Vector Machines, Neural Networks
Clustering	Group similar data points together without prior knowledge of categories	K-Means, Hierarchical Clustering, DBSCAN, Gaussian Mixture Models, Affinity Propagation, Spectral Clustering
Anomaly Detection	Identifying unusual patterns or outliers within a dataset	Isolation Forest, One-Class SVM, Local Outlier Factor, Autoencoders, DBSCAN, Elliptic Envelope
Time Series	Predicting future values or events based on historical time-series data	Autogressive Models (ARIMA, SARIMA) Exponential Smoothing, State Space Models, LSTM, GRU, Prophet, Facebook's NeuralProphet
Recommender System	Providing personalized recommendations to users	Collaborative Filtering, Content-based Filtering, Hybrid Recommender Systems, Matrix Factorization, Deep Learning
Dimensionality Reduction	Reducing the number of features while retaining important information	Principal Component Analysis (PCA), t-distributed Stochastic Neighbor Embedding (t-SNE), Linear Discriminant Analysis (LDA), Autoencoders, UMAP
Natural Language Processing	Analyzing and understanding human language data	Bag-of-Words, Word Embeddings, recurrent Nerual Networks (RNN), LSTM, GRU, Transformer Models (BERT, GPT), SpaCy, NLTK, Hugging Face Transformers

Our Focus

This table of models was created with the help of ChatGPT. For a review of Machine Learning Models and Topics. Review the earlier WWCode Data Science Machine Learning Study Group

Problem Statement Overview

1. **Context:** Why are you creating this initiative and/or project? What is the pain point or issue at hand?

2. **Criteria for Success:** What main criteria or solution will indicate a success?

3. **Scope of Solution:** What is the use case and/or specific items of focus for the initiative?

4. **Constraints for Solution:** What potential roadblocks might prevent the initiative from succeeding?

5. **Stakeholders to provide key insight:** Who are the key decision makers that need to be involved in the project? Who will help source data? Who do you present your progress and findings to?

6. **Key data sources:** what are the key features or data you need to answer questions proposed from the problem you are trying to solve?

Our Problem Statement

1. **Context:** One of the aspects of Spotify's song recommendations is using user data on songs that have been liked in the past. We need to analyze the liked songs in order to know what similar song a user might also enjoy.
2. **Criteria for Success:** Based on a user's Spotify data, can we predict if a song will be liked or not?
3. **Scope of Solution:** Create a basic binary classification model to predict a like or no like on songs. Our chosen solution scope does not create recommendations, but simply labels for provided songs with a 0 for not like or a 1 for like. An extension could be to create a content-based filtering engine as a recommender.
4. **Constraints for Solution:** Data is from one user and does not include geographical data.
5. **Stakeholders:** Product Managers, Senior Leadership, Data Engineers, User.
6. **Data:** Historical user data on songs listened to and like reactions. Data includes song attributes.

Target Variable / Dependent Variable

1. The feature or outcome you are trying to predict.
2. Differs slightly based on type of problem:
    a. Classification - categorical outcome
    b. Regression - continuous or numeric result
    c. Recommendation systems - user preference

What is our target variable?

- Binary Classification -> categorical outcome
- Song popularity
    - Like vs don't like

Technology we will use

Language: Python (others might include R)

Web Application / IDE: Google Colab (others might include Jupyter Notebook, Visual Studio, Spyder, PyCharm, RStudio)

Google Colab Quick Start

Session 1 Notebook

1. **Open a Colab notebook and save it**: https://colab.research.google.com/
2. **Notebook settings:** Select "Runtime" -> "Change runtime type" to adjust the runtime environment -> select Python version and set hardware accelerator to none if you are not using GPUs to enhance performance in processing).
3. **Write code:** Use code cells to enter Python code, which can be executed by pressing Shift+Enter or clicking the "Play" button.
4. **Add text:** Click "+ Text" to insert markdown cells for documentation, explanation, or instructions.
5. **Import files:** Use the "Files" tab in the left sidebar to upload data or files from your computer or Google Drive.
6. **Save and share:** Click "File" -> "Save" to save your notebook to Google Drive, or use "Share" to invite collaborators and manage access permissions.
7. **Export options:** Choose "File" -> "Download" to export your notebook in various formats (e.g., .ipynb, .py, or PDF).

Join us on Slack to ask questions and keep the discussion going!

Use the channel:

#build-a-ds-project