

📁 **ebtezcan / dsc-phase-4-project**

generated from [jirvingphd/osemn-project-template](#)

📄 View license

★ 0 stars 🍴 0 forks

☆ Star

👁 Unwatch ▾

- <> Code
- 🔍 Issues
- 🔗 Pull requests
- ▶ Actions
- 📁 Projects
- 📖 Wiki
- 🛡 Security

🔗 master ▾

⋮

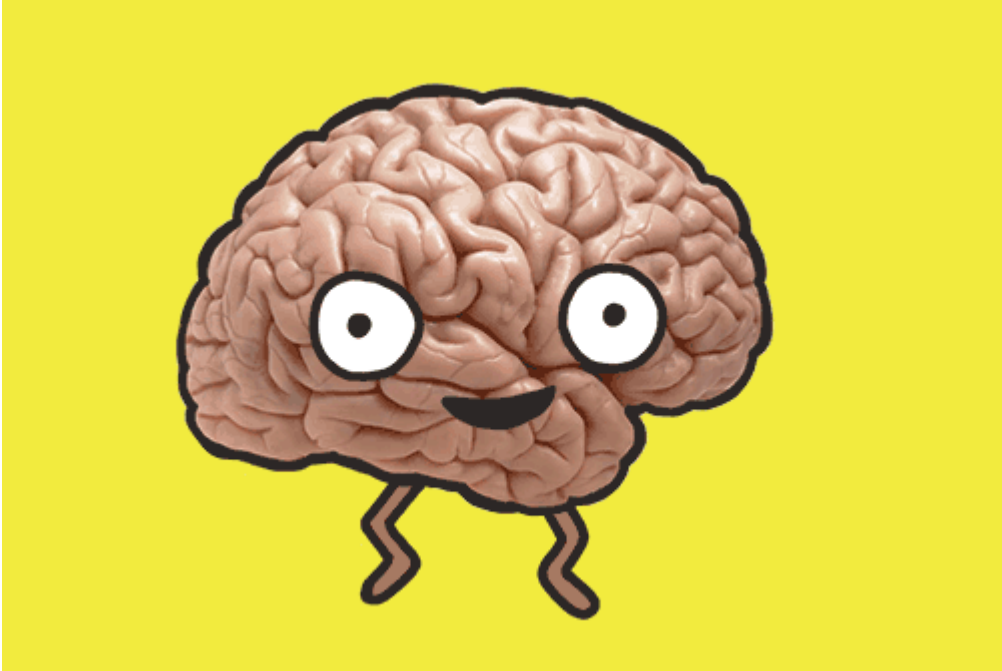
👤 **ebtezcan** Added multiclass classification with multiple models, minor tweaks... yesterday ⌚ 9

[View code](#)

☰ README.md ✎

Phase 4 Project

Final phase down -- you're absolutely crushing it! You've made it all the way through one of the toughest phase of this course. You must have an amazing brain in your head!



Project Overview

For this phase, you will choose a project that requires building one of these four models:

- Time Series Modeling
- Recommendation System
- Image Classification with Deep Learning
- Natural Language Processing

The Data

We have provided a dataset suitable to each model, but you are also encouraged to source your own dataset. If you choose your own dataset, **run the dataset and business problem by your instructor for approval** before starting your project.

How to Choose a Project

When choosing a project, consider:

1. **Depth:** Choose a project that similar to what you want to do for your capstone project (Phase 5). This will allow you to practice those methods in a group setting before needing to use it independently. This will help you build a better Capstone project and a portfolio that demonstrates the ability to deeply learn and implement one modeling approach.
2. **Breadth:** Choose a problem that you don't necessarily plan to use in your capstone project. This will allow you to develop applied experience with multiple modeling approaches. This will help you refine your areas of interest and build a portfolio that demonstrates the ability to learn and implement multiple modeling approaches.

If you are feeling overwhelmed or behind, we recommend you choose Problem #3: Image Classification with Deep Learning.

Problem 1: Time Series Modeling

If you choose the Time Series option, you will be forecasting real estate prices of various zip codes using data from [Zillow Research](#). For this project, you will be acting as a consultant for a fictional real-estate investment firm. The firm has asked you what seems like a simple question:

What are the top 5 best zip codes for us to invest in?

This may seem like a simple question at first glance, but there's more than a little ambiguity here that you'll have to think through in order to provide a solid recommendation. Should your recommendation be focused on profit margins only? What about risk? What sort of time horizon are you predicting against? Your recommendation will need to detail your rationale and answer any sort of lingering questions like these in order to demonstrate how you define "best".

There are many datasets on the [Zillow Research Page](#), and making sure you have exactly what you need can be a bit confusing. For simplicity's sake, we have already provided the dataset for you in this repo -- you will find it in the file `time-series/zillow_data.csv`.

The goal of this project is to have you complete a very common real-world task in regard to time series modeling. However, real world problems often come with a significant degree of ambiguity, which requires you to use your knowledge of statistics and data science to think critically about and answer. While the main task in this project is time series modeling, that isn't the overall goal -- it is important to understand that time series modeling is a tool in your toolbox, and the forecasts it provides you are what you'll use to answer important questions.

In short, to pass this project, demonstrating the quality and thoughtfulness of your overall recommendation is at least as important as successfully building a time series model!

Starter Jupyter Notebook

For this project, you will be provided with a Jupyter notebook, `time-series/starter_notebook.ipynb`, containing some starter code. If you inspect the Zillow dataset file, you'll notice that the datetimes for each sale are the actual column names -- this is a format you probably haven't seen before. To ensure that you're not blocked by preprocessing, we've provided some helper functions to help simplify getting the data into the correct format. You're not required to use this notebook or keep it in its current format, but we strongly recommend you consider making use of the helper functions so you can spend your time working on the parts of the project that matter.

Evaluation

In addition to deciding which quantitative metric(s) you want to target (e.g. minimizing mean squared error), you need to start with a definition of "best investment". Consider additional metrics like risk vs. profitability, or ROI yield.

Problem 2: Recommendation System

If you choose the Recommendation System option, you will be making movie recommendations based on the [MovieLens](#) dataset from the GroupLens research lab at the University of Minnesota. Unless you are planning to run your analysis on a paid cloud platform, we recommend that you use the "small" dataset containing 100,000 user ratings (and potentially, only a particular subset of that dataset).

Your task is to:

Build a model that provides top 5 movie recommendations to a user, based on their ratings of other movies.

The MovieLens dataset is a "classic" recommendation system dataset, that is used in numerous academic papers and machine learning proofs-of-concept. You will need to create the specific details about how the user will provide their ratings of other movies, in addition to formulating a more specific business problem within the general context of "recommending movies".

Collaborative Filtering

At minimum, your recommendation system must use collaborative filtering. If you have time, consider implementing a hybrid approach, e.g. using collaborative filtering as the primary mechanism, but using content-based filtering to address the [cold start problem](#).

Evaluation

The MovieLens dataset has explicit ratings, so achieving some sort of evaluation of your model is simple enough. But you should give some thought to the question of metrics. Since the rankings are ordinal, we know we can treat this like a regression problem. But when it comes to regression metrics there are several choices: RMSE, MAE, etc. [Here](#) are some further ideas.

Problem 3: Image Classification with Deep Learning

If you choose this option, you'll put everything you've learned together to build a deep neural network that trains on a large dataset for classification on a non-trivial task. In this case, using x-ray images of pediatric patients to identify whether or not they have pneumonia. The dataset comes from Kermany et al. on [Mendeley](#), although there is also a version on [Kaggle](#) that may be easier to use.

Your task is to:

Build a model that can classify whether a given patient has pneumonia, given a chest x-ray image.

Aim for a Proof of Concept

With Deep Learning, data is king -- the more of it, the better. However, the goal of this project isn't to build the best model possible -- it's to demonstrate your understanding by building a model that works. You should try to avoid datasets and model architectures that won't run in reasonable time on your own machine. For many problems, this means downsampling your dataset and only training on a portion of it. Once you're absolutely sure that you've found the best possible architecture and other hyperparameters for your model, then consider training your model on your entire dataset overnight (or, as larger portion of the dataset that will still run in a feasible amount of time).

At the end of the day, we want to see your thought process as you iterate and improve on a model. A project that achieves a lower level of accuracy but has clearly iterated on the model and the problem until it found the best possible approach is more impressive than a model with high accuracy that did no iteration. We're not just interested in seeing you finish a model -- we want to see that you understand it, and can use this knowledge to try and make it even better!

Evaluation

Evaluation is fairly straightforward for this project. But you'll still need to think about which metric to use and about how best to cross-validate your results.

Problem 4: Natural Language Processing (NLP)

If you choose this option, you'll build an NLP model to analyze Twitter sentiment about Apple and Google products. The dataset comes from CrowdFlower via data.world. Human raters rated the sentiment in over 9,000 Tweets as positive, negative, or neither.

Your task is to:

- Build a model that can rate the sentiment of a Tweet based on its content.

Aim for a Proof of Concept

There are many approaches to NLP problems - start with something simple and iterate from there. For example, you could start by limiting your analysis to positive and negative Tweets only, allowing you to build a binary classifier. Then you could add in the neutral Tweets to build out a multiclass classifier. You may also consider using some of the more advanced NLP methods in the Mod 4 Appendix.

Evaluation

Evaluating multiclass classifiers can be trickier than binary classifiers because there are multiple ways to mis-classify an observation, and some errors are more problematic than others. Use the business problem that your NLP project sets out to solve to inform your choice of evaluation metrics.

Sourcing Your Own Data

Sourcing new data is a valuable skill for data scientists, but it requires a great deal of care. An inappropriate dataset or an unclear business problem can lead you spend a lot of time on a project that delivers underwhelming results. The guidelines below will help you complete a project that demonstrates your ability to engage in the full data science process.

Your dataset must be...

1. **Appropriate for one of this project's models.** These are time series, recommendation systems, deep learning, or natural language processing.
2. **Usable to solve a specific business problem.** This solution must rely on your model.
3. **Somewhat complex.** It should contain thousands of rows and features that require creativity to use.
4. **Unfamiliar.** It can't be one we've already worked with during the course or that is commonly used for demonstration purposes (e.g. MNIST).
5. **Manageable.** Stick to datasets that you can model using the techniques introduced in Phase 4.

Once you've sourced your own dataset and identified the business problem you want to solve with it, you must to **run them by your instructor for approval**.

Problem First, or Data First?

There are two ways that you can source your own dataset: **Problem First** or **Data First**. The less time you have to complete the project, the more strongly we recommend a Data First approach to this project.

Problem First: Start with a problem that you are interested in that you could potentially solve using one of the four project models. Then look for data that you could use to solve that problem. This approach is high-risk, high-reward: Very rewarding if you are able to solve a problem you are invested in, but frustrating if you end up sinking lots of time in without finding appropriate data. To mitigate the risk, set a firm limit for the amount of time you will allow yourself to look for data before moving on to the Data First approach.

Data First: Take a look at some of the most popular internet repositories of cool data sets we've listed below. If you find a data set that's particularly interesting for you, then it's totally okay to build your problem around that data set.

There are plenty of amazing places that you can get your data from. We recommend you start looking at data sets in some of these resources first:

- [UCI Machine Learning Datasets Repository](#)
- [Kaggle Datasets](#)
- [Awesome Datasets Repo on Github](#)

The Deliverables

There are three deliverables for this project:

- A **GitHub repository**
- A **Jupyter Notebook**
- A **non-technical presentation**

Review the "Project Submission & Review" page in the "Milestones Instructions" topic for instructions on creating and submitting your deliverables. Refer to the rubric associated with this assignment for specifications describing high-quality deliverables.

Key Points

- **Choose your project quickly.** We've given you a lot of choices - don't get stuck spending too much time choosing which project to do. Give yourself a firm time limit for picking a project (e.g. 2 hours) so you can get on with making something great. Don't worry about picking the perfect project - remember that you will get to do a new, larger Capstone project very soon!
- **Your Jupyter Notebook should demonstrate an iterative approach to modeling.** This means that you begin with a basic model, evaluate it, and then provide justification for and proceed to a new model. This is a great way to add narrative structure to your notebook, especially if you compare model performance across each iteration.
- **You must choose and implement an appropriate validation strategy.** This is one of the trickiest parts of machine learning models, especially for models that don't lend themselves easily to traditional cross-validation (e.g. time series & recommendation systems).

Getting Started

Create a new repository for your project to get started. We recommend structuring your project repository similar to the structure in [the Phase 1 Project Template](#). You can do this either by creating a new fork of that repository to work in or by building a new repository from scratch that mimics that structure.

Project Submission and Review

Review the "Project Submission & Review" page in the "Milestones Instructions" topic to learn how to submit your project and how it will be reviewed. Your project must pass review for you to progress to the next Phase.

Summary

This project is your chance to show off your data science prowess with some advanced machine learning algorithms. Now that you've gone through all of the core course content, we're excited to see what you are able to do!

Releases

No releases published

[Create a new release](#)

Packages

No packages published

[Publish your first package](#)

Languages

● Jupyter Notebook 100.0%