DeepSense Onboarding

Contents

Resources

Wiki

https://docs.deepsense.ca/index.php?title=DeepSense_Documentation

Tutorials

DeepSense YT - YouTube

Engagement

Expectations of DeepSense Staff

When you work with us on a project, here is what you can expect from us:

1. We respond to emails within 24 business hours

Expectations of DeepSense Students

When you work with us on a project, here is what we can expect from you:

1. Respect
2. Ask questions now, don't wait
3. To clearly understand your problem
4. Respond to emails/surveys in a timely manner
5. Active involvement and engagement in the project
6. Exercise general business etiquette (Punctual for meetings, etc.)
7. That you have watched this video (https://www.youtube.com/watch?v=rZJ733w5GfI&t=2826s)

Expectations of DeepSense Industry Projects

Working with an industry partner is a privilege not all students experience. This opportunity is giving you a chance to build you resume, network and portfolio:

1. Have clear discussions and establish agreement with the company about when they expect you to be available to meetings and the regular daily work hours you will spend on this project
2. Know who within the company is your primary point of contact. Confirm how they want to engage with you. Do they want weekly meetings, a teams or slack channel? Do they want you to send questions constantly or compile a weekly email summary
3. Outline weekly goals and measure your performance against the goals. Share you weekly work plan with others to hold yourself accountable.

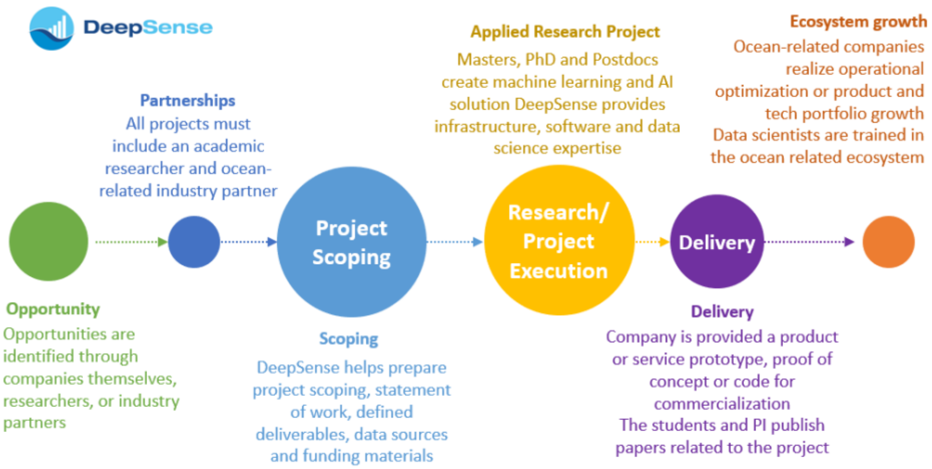
Project Identification

DeepSense provides support to complete academic applied research project. The following outlines how DeepSense may provide you with support. All deepSense projects are ocean related projects designed to help expand the use of data, AI, machine learning and visualizations in the ocean sector. There are a range of methods for a project to be developed and launched. Often projects are identified through DeepSense or Academic researchers. As projects are fleshed out in greater detail, effort may be taken to seek out interested in students to engage in projects. It is also common that researchers already have established relationships with students, for example those enrolled in their classes, and these students are recruited to contribute to project deliverables.

Seeking a project? If you are a student looking to complete a project, please reach out to DeepSense at info@deepsense.ca. We recommend providing details about:

1. Your current program, year of study institution
2. Your background and areas of interest
3. Reasons why you are interested in a career in ocean data.

DeepSense process


DeepSense
Partnerships
All projects must include an academic researcher and ocean-related industry partner
Applied Research Project
Masters, PhD and Postdocs create machine learning and AI solution DeepSense provides infrastructure, software and data science expertise
Ecosystem growth
Ocean-related companies realize operational optimization or product and tech portfolio growth
Data scientists are trained in the ocean related ecosystem
Project Scoping
Research/ Project Execution
Delivery
Opportunity
Opportunities are identified through companies themselves, researchers, or industry partners
Scoping
DeepSense helps prepare project scoping, statement of work, defined deliverables, data sources and funding materials
Delivery
Company is provided a product or service prototype, proof of concept or code for commercialization
The students and PI publish papers related to the project

Project Support - Monthly surveys or meeting

DeepSense expect the projects to run smoothly achieving the key milestones of the projects according to the approved schedule. Monthly project meeting or monthly survey is the tool to ensure that. DeepSense Project Manager will schedule monthly project meeting and/or send out monthly survey to capture the status of the project, resource and infrastructure usage. Users must provide required information by completing the survey or attending project meeting.

How to seek support

If you are looking for assistance with your project, we recommend that you reach out as soon as possible. Struggling for a few days to try to figure out errors or understand how to submit jobs can greatly delay you effort. We recommend you keep this document and our wiki accessible to ensure you can complete your own troubleshooting. If you cannot easily find an answer, reach out. This will help us understand where we need to improve and enhance DeepSense processes and instructions.

Cloud Access

Request access DeepSense cloud resources

Currently DeepSense has established systems on two cloud providers, Amazon Web Services and Google Cloud Platform. Users can choose their own environment from Notebook (i.e. serverless) or independent Compute Instance (i.e. virtual machines) options. Users will have console access to use the cloud services on AWS or GCP as per allocation.

*Please provide specification information to cloud administrator by email.
RAM (GB):
Data format:

Data Storage (GB or TB):
Number of GPUs: *if applicable
GPU Memory (GB): *if applicable

Your DeepSense cloud account

Once a DeepSense project has been identified and approved, your cloud account will be created, and you will receive an introductory email. It will contain links to this document, our wiki (docs.deepsense.ca), and other useful information. Your account will remain active for the duration of your project.

How to access DeepSense cloud resources

Users can access DeepSense's cloud resources from their own computers/laptops or the computers in the labs on DAL's campus. For users using notebook instances (e.g., AWS SageMaker and Google Colab) can access the notebook by directly logging in to the cloud console. For users using dedicated instances (e.g., EC2 instances, Compute instances -vm) will be launch the instance from cloud console and they need a terminal to interact with DeepSense's cloud instances. On Mac machines, users can find the Terminal application in Applications. However, on Windows systems, users can download and install Putty which is a very good SSH client widely used. The detailed instruction of installing Putty can be found here. Users can also user other SSH clients as their preferences. Users can use the Terminal or SSH clients to type their Linux commands to interact with the DeepSense cloud instance as most of them are Amazon Linux based on Linux.

Users need to learn some basic Linux commands to access their directories and files. A brief introduction of Linux commands can be found here.

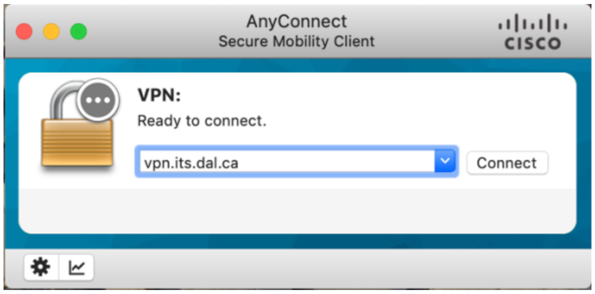
Using DeepSense cloud resources

Please follow the wiki page for in detailed steps here.

VPN

For compute instances you are required to use If you are on Dalhousie's campus and you have the access to Dalhousie's network, you are able to access DeepSense's systems directly. To connect to the DeepSense platform from outside of the Dalhousie Campus, you'll need to use a VPN. If you are student, staff or faculty, you can use the Dalhousie VPN (https://wireless.dal.ca/vpnsoftware.php).

Once you install the AnyConnect software, open the application. You will need to enter the address for the VPN, which is vpn.its.dal.ca:



VPN:
Ready to connect.
vpn.its.dal.ca

Once you click on Connect, you will be prompted to enter your netID and password.

If you are not a Dalhousie staff, student, or faculty but require offsite access and cannot use the Dalhousie VPN then contact your project leader or support@deepsense.ca to make different arrangements.

DeepSense cloud resource systems

Users will have console access to AWS or Google Cloud as per the allocation. Amazon Deep Learning AMI is used to launch all compute resources on Amazon, so all environments are pre-configured and ready to use. The notebooks are also configured on AWS SageMaker Notebook instances and Google Colab.

Data Processing and Training

Cleaning data

Data cleaning is a necessary step in any problem. Most people could tell that a grainy image, or an audio clip with static could be classified as not clean. Not everyone would realise that unclean data means more than just data with noise. It can also mean misspelled text, data in the wrong field of a spreadsheet, or mistakes in transcribing. If you're collecting data from different sites, they may not all have the same data fields, or may be missing some fields. While you generally want as much data as you can gather, the quality of the data needs to be stressed, as well as quantity. Some data may need to be left out as a result of the cleaning.

We have found that data cleaning is a large hurdle that must be overcome before any machine learning models can be trained. Unfortunately, we've also found it to be quite time consuming. Make sure you are aware of this when defining the timeline of your project. Sometimes the cleaning process can be automated using scripts. If you need help with this, don't hesitate to contact support@deepsense.ca.

Creating training, test and validation sets

When training a machine learning model, one needs to already have training, testing and validation sets created. These are subsets of your data, and can vary in size. The training set is the largest set, usually containing 70-80% of your data. The test and validation sets are usually the same size, and thus would be between 10-15% of your data each.

It is important that the sets be created randomly, but in such a way that they are representative of the dataset. This is most important with time-sensitive data. For example, if you were collecting data over a 10-month period, using months 1-8 for training, month 9 for testing and month 10 for validation would likely not yield very good results. This is because there could be anomalies in the data at different times. If this was weather data, and there was a hurricane in month 10, but none in the other months, the model would not have been trained on such data, and so would have a very poor validation accuracy.

It is also important that other variables be properly represented in your three sets. If you were collecting data at more than one site, you'd want to have data from each site in each of the three sets, which may not happen if you randomly select the data. This is especially true if the sites produce different amounts of data.

There is no one right answer for how to separate the data into these three sets. You will need to carefully consider how you do it. If you don't, and your model doesn't work as well as you'd hope, you might think it was the fault of your model, and spend months tinkering with it – even though the problem could be with your datasets.

Best practices for managing multiple versions of data

It is very important to have multiple copies of your data in case you lose one. But managing all of that data can be tricky. We will outline some best practices for managing your data. Though, the implementation may be slightly different depending on the type of data you are using, the overall ideas will be the same.

When performing any type of data analysis, there will be intermediate steps in which a new version of the data is stored. Typically, this will include raw data, cleaned data, processed data and analysis results.

First, it is very important to have multiple copies of your raw data. Collecting data can be very time consuming and costly. If you only had your raw data stored on one computer, and it fails, you wouldn't just lose your data. You would also have wasted all the time and effort put into collecting the data in the first place. It is recommended that you store your raw data in several different places, such as an external hard drive, one or more computers, or a larger data center. Whichever option you go with, there are two important things to remember. The first is that hard drives fail. We hope it doesn't happen often, but we must be prepared in case it does. Ideally, wherever you store it will have some redundant hard drives, such as a raid array. This means that if one of the drives fails, you don't lose the data. You can simply replace the drive, and rebuild the raid array. The second is to make sure your data is stored in different physical locations. It doesn't matter how many copies of the data you have, if it is all in the same room. A fire, or water leak, could destroy all of them at once. You would want to have copies in different rooms, and ideally, in different buildings.

Once you have the raw data properly stored, and begin to clean/process/analyze the data, you will find that you end up with multiple versions (not just copies) of the data. You should never overwrite the raw data when you are processing it, in case you make a mistake. Even when you are simply cleaning the data, you should store that as a new copy. However, there may also be several steps in cleaning, and processing of the data. You may not need a new copy for each step, as the amount of storage space you require could balloon quickly. However, any step that is done by hand should be saved in a new file. Any automated steps, like using a piece of software or script, can be easily replicated, while anything manually done cannot.

It is important to have a file naming convention for your data. As an example, software versioning is usually sequential such as *major.minor[.build[.revision]]* (example: 1.2.12.102). However, with datasets it isn't so obvious. After each step in the cleaning/processing pipeline, you'll want to add that information to the filename. Not just the last step done, but each step that has been done, as the order may change. The hard part is keeping the filename a reasonable length, while not being so succinct as to lose information.

At each step you should also make sure you take detailed notes of what you have done to the data. Ideally, you'd want to have a script or other form automation that you can use for each step in the pipeline. It is possible to find out that the processing was done with a wrong parameter, or wrong method. In which case, you want the ability to go back and rerun all these steps without having to start from scratch. It is also likely that you will continue to collect data, or obtain new datasets, so you will want to have the ability to process this quickly.

One problem we've encountered before is that the processing pipeline is still being developed as someone is collecting data. Often, there will be mistakes made in the beginning, or incorrect parameters used that results in incorrect data. People often label this by changing the file name to indicate is incorrect and shouldn't be used. As they collection and processing progress, there are fewer mistakes made, and so newer datasets are often very clean and don't contain extra files. However, it is important to go back to the earlier datasets and remove the erroneous files once you have your pipeline finalized, or possibly even rerun the analyses steps.

Training

Which algorithm is the best

Depending on your project, you can work with your PI or reach out to DeepSense to discuss the Algorithms & Software specialist to brainstorm the various types of models you may want to try.

For a more comprehensive list, please access [INSERT LINK]

Frameworks to consider

Keras, TensorFlow, PyTorch and Caffe are the popular deep learning frameworks.

Keras is an open source framework that provides high level APIs for large machine learning applications such as neural networks. Its simple and easy to use architecture facilitates fast development of models. It is most suitable for Rapid Prototyping and Small Datasets.

TensorFlow is an open source AI framework developed by Google. It provides both high and low-level APIs that has library for numerical computations and large-scale machine learning applications like neural networks. It is most suitable for Large Dataset, High Performance, and Object Detection.

PyTorch is an open-source Machine learning library developed by Facebook. It is based on the torch library and provides lower-level API that helps the user to customize the layers and optimize tasks. It offers python like coding, distributed training, debugging capabilities and supports dynamic computation graph. It is preferred where single homogeneous computation is not needed for example such as Natural Language Processing (NLP).

Caffe (Convolutional Architecture for Fast Feature Embedding) is an open source deep learning framework originally developed at University of California. It supports many different types of deep learning designs such as CNN (Convolutional Neural Network), RCNN(Region-based Convolutional Neural Networks), and fully connected neural network that ease image classification and image segmentation effectively.

Caffe2 is launched by Facebook in 2017 with the addition of new features to Caffe such as Recurrent Neural Networks (RNN), flexibility, large-scale distributed training, and support for mobile deployment. It has been merged into PyTorch in 2018 to create PyTorch 1.0 that is suitable for both research and production. It is well-suited for the applications that hold large-scale image classification and object detection.

Evaluate the model

Once you are finished with building a model, you need to check the performance of the model by using the various evaluation metrics. These metrics will tell you how accurately your model will perform and predict the new values. Application of model for prediction without checking its performance is of no use. The various metrics for evaluating the model are:

- **Accuracy** gives proportion of total number of correct predictions out of total number of predictions. Higher the accuracy, better is the model.
- **Precision** gives proportion of actual positive cases that were correctly identified.
- **Recall** gives the proportion of actual positive cases out of correctly identified cases.
- **F1-Score** is the harmonic mean of Precision and Recall and is useful in the case when the size of the positive class is relatively small.
- **AUC (Area under ROC curve)**: The ROC curve is a probability curve plotted with true positive rate against the false positive rate. AUC represents degree of separability between classes. Higher the AUC, better the model is at distinguishing between classes. Its value lies between 0 and 1. 1 means better model.
- **RMSE (Root Mean Square Error)** is the standard deviation of the errors which occur when a prediction is made on a dataset with continuous values in target. Lower the value, better is the performance of the model.
- **R squared is** also known as the **coefficient of determination**. It indicates how close the predicted values to the actual data values in case of continuous data. Its value lies between 0 and 1 where 0 indicates worst fit and 1 indicates best fit to the dataset.

***RMSE and R-**Squared are specifically for analyzing the performance of regression model.

When are you finished?

Start with baseline model. Check the performance using evaluation metrics and try to improve the performance by tuning the parameters. If you are trying more than one model, then compare the models using various evaluation metrics and choose one that shows high performance. Use that model for the prediction of new values. Once you satisfied with the prediction, model is ready for use.

Project Completion

Project / data Removal

When a project starts, it must have completion date. Once a user leaves DeepSense, a project is completed, or the completion date passes, it is expected that user will take their results, and remove their data in a timely fashion. Data may be purged 30 days after the project is completed. This can be extended to three months, if required. To do so, contact (support@deepsense.ca). Please include a paragraph justifying your need.

For users working on multiple projects, their home directory will remain until they are done with all projects.

How to delete confidential info

Some project will involve the use of confidential data. When such a project finishes, the user should be aware of the need to properly delete anything confidential. There is no need to do anything extra on the DeepSense platform, as long as you delete the data.

However, it is important to think about where else you have that data. You may have a copy on your laptop, or external drive. You may have data stored as an email attachment. It is important to make sure all of this data is properly deleted.

Data Removal ~ Risk

Some projects will have more than one phase, so they may continue even after you are done with your particular piece. In which case, they data and scratch directories for that project will remain stored on our system for future use. It is expected most projects will take no more than 12 months.

Possible uses for data to consider

After your project is completed, you don't necessarily have to delete your data from our system and leave. There could be future research that could be done on the same topic. We've had several projects that successfully finished, but still had open questions. Could we add new data sources to the existing data? Could we try a different type of model? Some of the projects continued with another student attempting to answer these questions.

DeepSense projects often start as a proof of concept. In this case, the company may wish to continue research to fine tune their models. Though, if the data is confidential, there may be details that need to be worked out in order for a new student to continue using it. It's even possible the company is looking to hire the student to continue the work.

It is certainly possible that when your work finishes, the project is complete and the data can be removed and deleted from our system. However, you should check with the company and with DeepSense to see if there are any other potential uses for the data.

Reporting to company or funders

For your project, the DeepSense team may have provided some support to prepare funding materials. Please note, it is up to the student (and their PI if applicable) to manage and complete all funding reporting requirements. DeepSense is not accountable for seeking or producing any materials.

Paper prep

Typically, the work a student does on a project is suitable for publication. The company and/or funding agency usually requires a report of the work you've done, so the first step is to start with that report and expand upon it as publications will typically have more technical details than your reports. If you are inexperienced publishing papers, the DeepSense team is here to support you. A typical machine learning paper will have sections like:

- Abstract
- Introduction
- Data (sources/processing)
- Machine Learning Model (development/training/tuning)
- Analysis
- Conclusion

We are also able to help with choosing the right journal/conference to submit your paper to. Depending on the project we may look to submit it to a journal in the oceans space, or it may be more suitable to submit it to a computer science journal. Either way, we have experience submitting articles and we're happy to help you get your work published.

Equipment

If provided access to use DeepSense cloud, software and computing resources, all work with only be used for the intended purpose.

DeepSense acknowledgement

Please acknowledge DeepSense and your industry partners when publishing results that used DeepSense resources. DeepSense resources include computing hardware, computing software, and staff expertise. The exact wording of the acknowledgement may vary. Here is an example:

"This research was enabled in part by support provided by (industry partner)(web address) and DeepSense (www.deepsense.ca)."

We would appreciate references to any published acknowledgments so they can be included in reports on the impact of DeepSense.

DeepSense can also help advertise your success and collaborate on media releases. Please contact us if you have news to share about research using DeepSense resources.

Exit survey

Exit survey is an important tool to capture the experience of the user during DeepSense project. Users must complete the exit survey sent by DeepSense Project Manager as one of the requirements of project completion. DeepSense will review the exit survey and analyze the responses. This may lead into review project processes and other resources to provide better user experience in upcoming projects.