

Week 3: Feature Engineering and Intro to ML



MAISI



Building Interpretable AI in Healthcare
Michigan Data Science Team - Winter 2026

Week 3 Agenda

01.

Icebreaker!

Every great project work session starts off with a great icebreaker.

02.

Feature Engineering

What is feature engineering, and why is it an important step to getting meaningful model output?

03.

Intro to Machine Learning

What is machine learning, what can we do with it, and what kinds of models can we use?

04.

Hands-On Data Science!

It's time to start training those models, engineering those features, and getting closer to our end goal!

01 Fun Icebreaker!!

Get to know us and start to learn about
each other with a fun icebreaker!



Icebreaker – Week 3

What would your dream exotic pet be? (No cats, dogs, hamsters, etc.)

Share with the people around you!! :)



Note: that's actually a Fennec Fox!



02

Feature Engineering

What is feature engineering, and why is it an important step to getting meaningful model output?





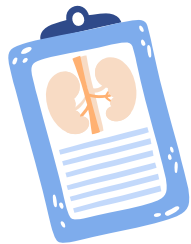
What is Feature Engineering?

- Feature engineering is the process of turning cleaned (yay for EDA!) input data into interpretable and ready-to-use features in a machine learning model.
 - The first steps in this are EDA and feature selection, which we covered last week.
- Processes Involved in Feature Engineering
 - Feature Creation - combining existing features into composite measures
 - Feature Selection - choosing the most relevant features (preprocessing)
 - Feature Scaling - making sure all features contribute equally to improve model



What does Feature Engineering Do?

- Improves Model Accuracy
 - Choosing the right features helps the model learn better
 - Better learning, better predictions!
- Reduces Overfitting
 - By using fewer features that hold more importance, we prevent the model from memorizing the data on which we train it
- Boosts Interpretability!!
 - When we choose our features well, we can establish a better understand of why our model makes its predictions



03

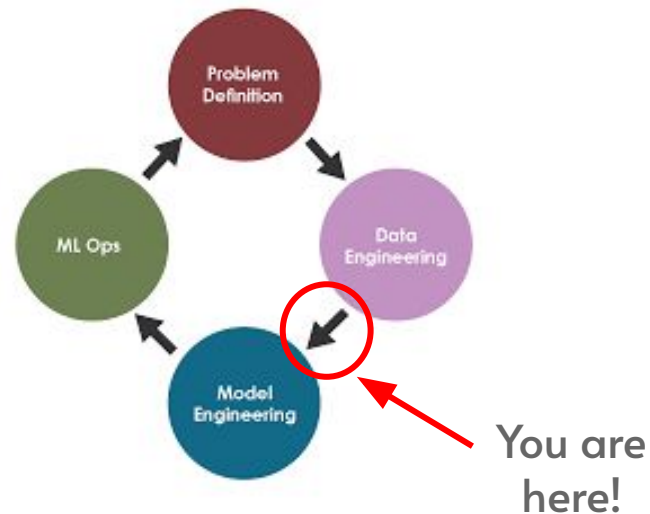
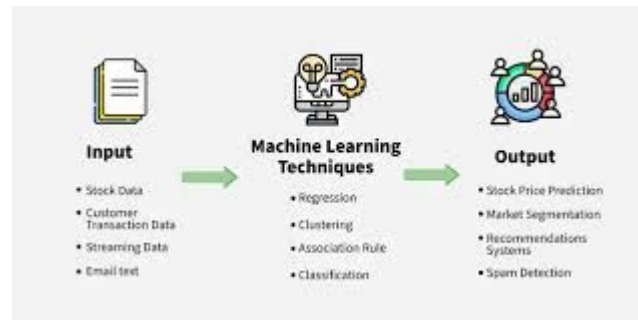
Intro to Machine Learning

What is machine learning, what can we do with it, and what kinds of models can we use?



What is Machine Learning?

- **Machine Learning vs. AI:** Machine Learning is a subset of AI that enables computers to *learn* from data and make predictions, without needing to be explicitly programmed for every specific task.
- **Main Types of Machine Learning**
 - **Supervised Learning:** models trained on labeled training data to predict outcomes.
 - **Unsupervised Learning:** models that find patterns in unlabeled data.
 - **Reinforcement Learning:** models learn through rewards system implemented by developers.



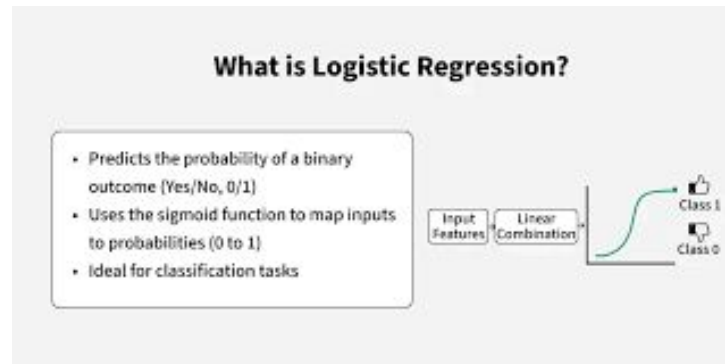
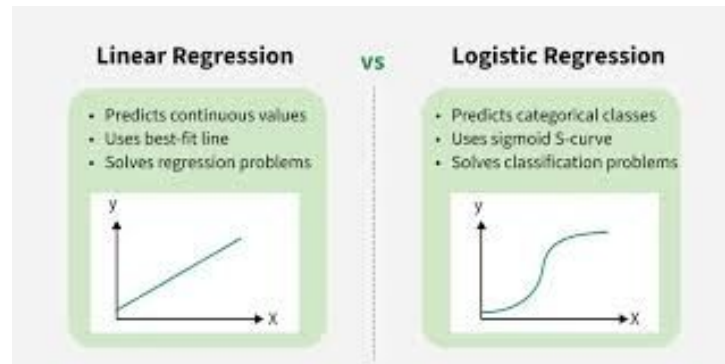


The Supervised Machine Learning Process

- **Collect/Process the Data:** In order to do supervised machine learning, we need data that is “labeled,” which means that each observational unit (row) has a known correct target value, which we call the “label.”
- **Split the Dataset:** We need to do a train-test split, where we divide our data into groups we train our model on (typically about 70–80%) and that on which we test it.
- **Train Your Model:** Feed the data into a model of your choice (which we will cover in a second), giving inputs as ‘x’ and your target variable as ‘y.’
- **Validate and Test:** Run the model on your training data and evaluate the accuracy of your output. If there is more to be desired, try adjusting weights or changing model!

ML Algorithms – Binary Logistic Regression

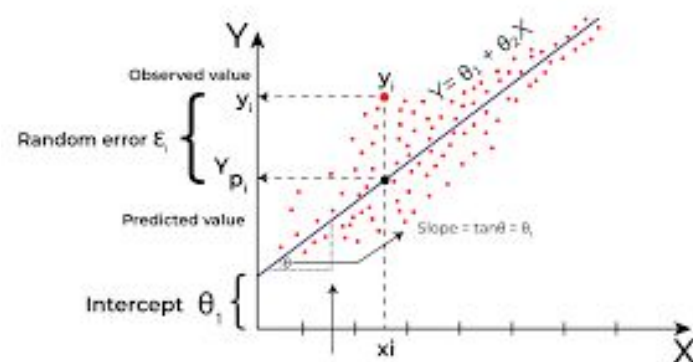
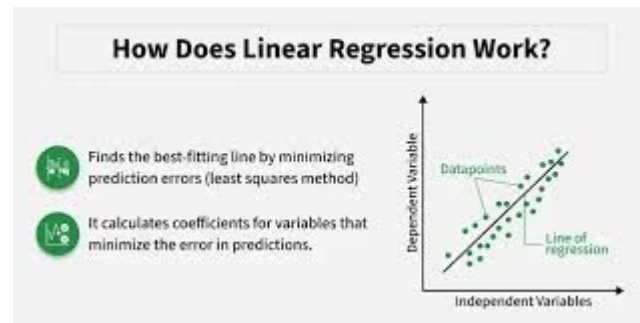
- **Binary logistic regression** works well in cases where we have a *binary target variable*
 - Ex. Stroke/No Stroke, Lung Cancer/None
- Considered a classification algorithm, as it attempts to fit points along a sigmoid curve ranging from $[0, 1]$ for our binary target
- Models the **log-odds** ($\ln(\text{odds of an event})$) as a linear combination of independent variables 'x,' where a certain threshold (usually 0.5) is used to determine to which category an observation with given log-odds should be assigned.



ML Algorithms - (Multiple) Linear Regression

- Boils down to a very familiar concept: $y = mx + b$
 - If you're doing multiple linear regression, this formula includes multiple input vars:
$$y = b + m_1x_1 + m_2x_2 + \dots + m_nx_n$$

- Plots a “best fit” line through the data that *minimizes squared errors* of observed vs. expected values.
 - Always fitted **linearly**
- Goal: find a best fit line that most accurately models the spread of data points for the target.
- Key Assumptions
 - Assumes linear relationship between x and y
 - (For multiple) No multicollinearity





References

Here are a handful of useful guides to high-utility functions, Git demos, and Colab explanations!

Our Demo! (for reference)



Link:

<https://mdst-ai-in-healthcare.streamlit.app/>

Project Timeline

Week 1 (01/25): Icebreaker/EDA intro, choosing a good dataset

Week 2 (02/01): Data Visualization, Further EDA

Week 3 (02/08): Feature Engineering and Intro to Machine Learning

Week 4 (02/15): Machine Learning and Model Optimization

Week 5 (02/22): Conformal Predictions, Risk Control, and Multi-Class Calibration

NO MEETING 03/01 OR 03/08 – SPRING BREAK

Week 6 (03/15): SHAP Values and Feature Importance Interpretation

Week 7 (03/22): Intro to Streamlit and Project Work Time

Weeks 8 + 9 (03/29 + 04/05): Final Project work time + Data Science Night prep!



Important Functions to Know (pt. 1)

Syntax	Description
<code>import pandas as pd</code>	Import the pandas library, using the alias pd for convenience
<code>import numpy as np</code>	Import the NumPy library, using alias np for numerical operations
<code>import seaborn as sns</code>	Import Seaborn for statistical data visualization
<code>df = pd.read_csv('file.csv')</code>	Load a CSV file into a pandas DataFrame for data manipulation
<code>df.head()</code>	Returns the first 5 rows of the DataFrame, useful for quickly inspecting data
<code>df.info()</code>	Provides a concise summary of the DataFrame, including data types and non-null values

Important Functions to Know (pt. 2)

Syntax	Description
<code>df.describe()</code>	Generates descriptive statistics of numerical columns (mean, median, quartiles, etc.)
<code>df['column_name']</code>	Access a specific column in the DataFrame, works like a key in a dictionary
<code>df.drop(columns=['col'...])</code>	Drops specified columns from the DataFrame, use <code>inplace=True</code> to modify the original DataFrame
<code>df.index</code>	Returns the index labels of the DataFrame
<code>df.isnull()</code>	returns a DataFrame of boolean values , where each entry indicates whether the corresponding value in df is NaN (missing)

Important Functions to Know (pt. 3)

Syntax	Description
<code>df['column'].value_counts()</code>	Returns the count of unique values in a specific column
<code>df['column'].mean()</code>	Returns the mean value of a numerical column
<code>df.corr()</code>	Computes correlation for numerical columns to understand relationships
<code>df.columns</code>	Lists all column names in the DataFrame , useful for renaming or viewing dataset structure
<code>df.shape</code>	Returns # of rows and columns

Important Functions to Know (pt. 4)

Syntax	Description
<code>df.rename(columns={'old' : 'new'}, inplace=True)</code>	Renames specific columns to new names
<code>df.groupby('category')['value']</code>	Groups the DataFrame using a specified column to perform aggregate functions (e.g., <code>.sum()</code> , <code>.mean()</code>)
<code>df['new_column'] = df['column'].apply(function)</code>	Applies a function to each element or column/row
<code>df.shape</code>	Returns a tuple representing the dimensions (rows, columns) of the DataFrame
<code>df.dropna()</code>	Removes rows or columns containing missing values

Important Functions to Know (pt. 5)

Syntax	Description
<code>df.to_numpy()</code>	Converts Pandas DataFrame into a NumPy array
<code>np.shape</code>	Returns a tuple representing the dimensions (rows, columns) of the array (similar to <code>df.shape</code>)
<code>np.reshape()</code>	Reshapes an original numpy array to the specified dimensions (rows, columns)
<code>np.array()</code>	Creates an N-dimensional array (ndarray) which is more memory efficient than standard python lists
<code>np.unique()</code>	Returns the unique elements in a NumPy array (also returns the counts of each element given the keyword argument, <code>return_counts=True</code>)

Important Functions to Know (pt. 6)


Syntax	Description
<code>pd.cut()</code>	Turns continuous numerical data into categorical data
<code>pd.get_dummies(drop_first = True)</code>	Dummy encodes categorical data (used with <code>drop_first</code> argument)
<code>OrdinalEncoder()</code>	Function for encoding categorical information that has an order
<code>train_test_split(X, y, test_size=0.2, random_state=42)</code>	Splits data into a training set and a testing set . The test set size can be changed
<code>LinearRegression()</code>	Used for initializing a linear regression model
<code>LogisticRegression()</code>	Used for initializing a logistic regression model

Important Functions to Know (pt. 7)

Syntax	Description
StandardScaler()	Turns continuous numerical data into categorical data
model.fit(X_train, y_train)	Fits a model to the training data
model.predict(X_test)	Uses the test data to make predictions
.coef	Attribute for getting the models coefficients (for linear and logistic regression)
mean_squared_error()	Calculates the average squared distance between the predicted and actual data points
mean_absolute_error()	Calculates the average distance between the predicted and actual values

How to Upload Files into Colab

Click on the folder in the sidebar



Commands

+ Code

+ Text

Run all

Files

..

sample_data

Week 1 - Pandas Practice

Here is where you import the libraries necessary to perform the following tasks!

```
import pandas as pd
import seaborn as sns

# Allows you to provide a path to a Google Drive address rather than a local file path
from google.colab import drive
drive.mount('/content/drive')
```

Mounted at /content/drive

Load the Google Forms .csv into a Pandas dataframe.

```
df = pd.read_csv('/content/MDST Week 1 - Pandas Practice.csv')
```

Print out the .head() and the datatypes.

```
df.head()
```

Timestamp	What is your name?	What is your major?	What year will you graduate?	If not, leave blank. (Answer in state code	If you're American, what state are you from?	If you're not American, what country are you from? If American,	(Approximately) How many years of coding experience do you have?	How many pets do you have?	How many credits are you taking this semester?	How many roommates do you have?	How many Football games have you been to?
-----------	--------------------	---------------------	------------------------------	--	--	---	--	----------------------------	--	---------------------------------	---

Disk

68.47 GB available

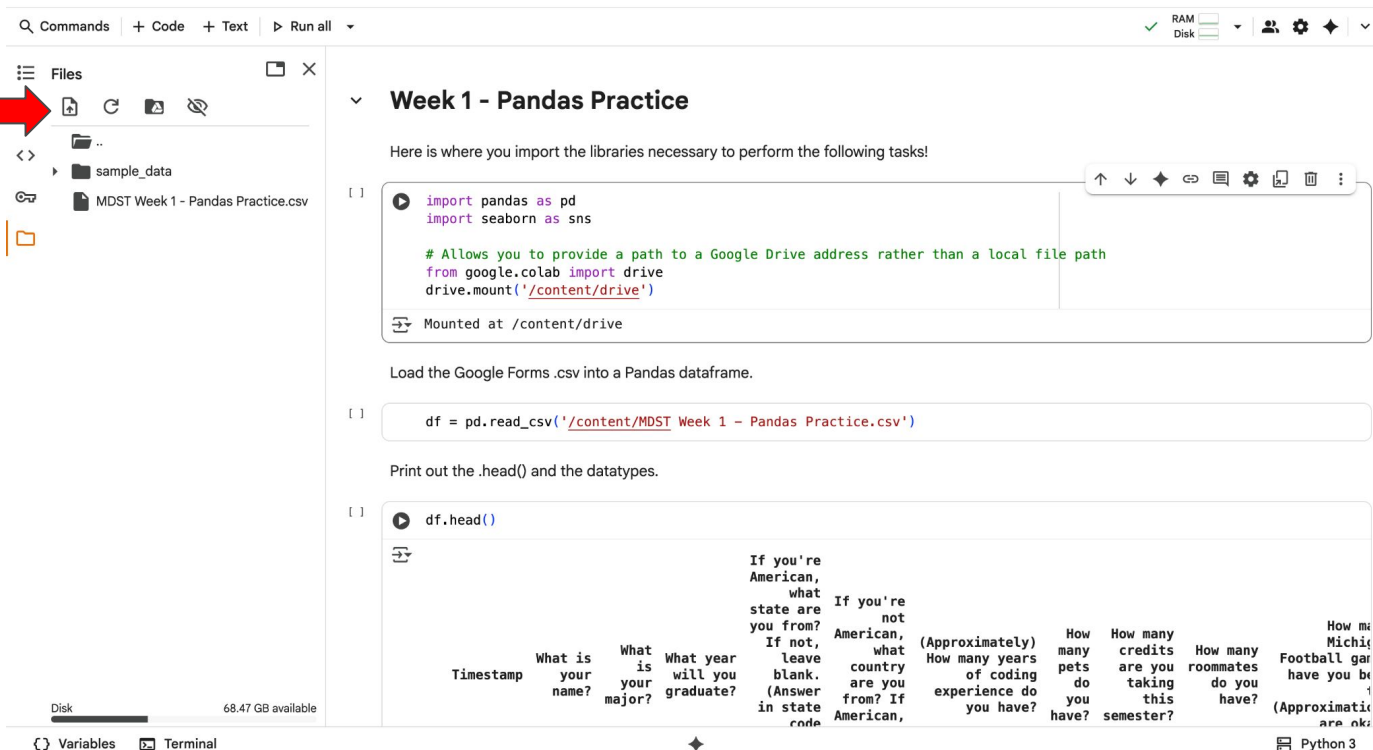
Variables

Terminal

Python 3

How to Upload Files into Colab

Click the upload button and select the file you want to upload



The screenshot shows the Google Colab interface. On the left, the 'Files' pane displays a directory structure with a folder named 'sample_data' and a file named 'MDST Week 1 - Pandas Practice.csv'. A red arrow points to the upload icon (a document with a plus sign) in the Files pane. The main code editor area is titled 'Week 1 - Pandas Practice' and contains the following Python code:

```
import pandas as pd
import seaborn as sns

# Allows you to provide a path to a Google Drive address rather than a local file path
from google.colab import drive
drive.mount('/content/drive')
```

Below the code, a message indicates 'Mounted at /content/drive'. The next code block shows the file being loaded into a Pandas dataframe:

```
df = pd.read_csv('/content/MDST Week 1 - Pandas Practice.csv')
```

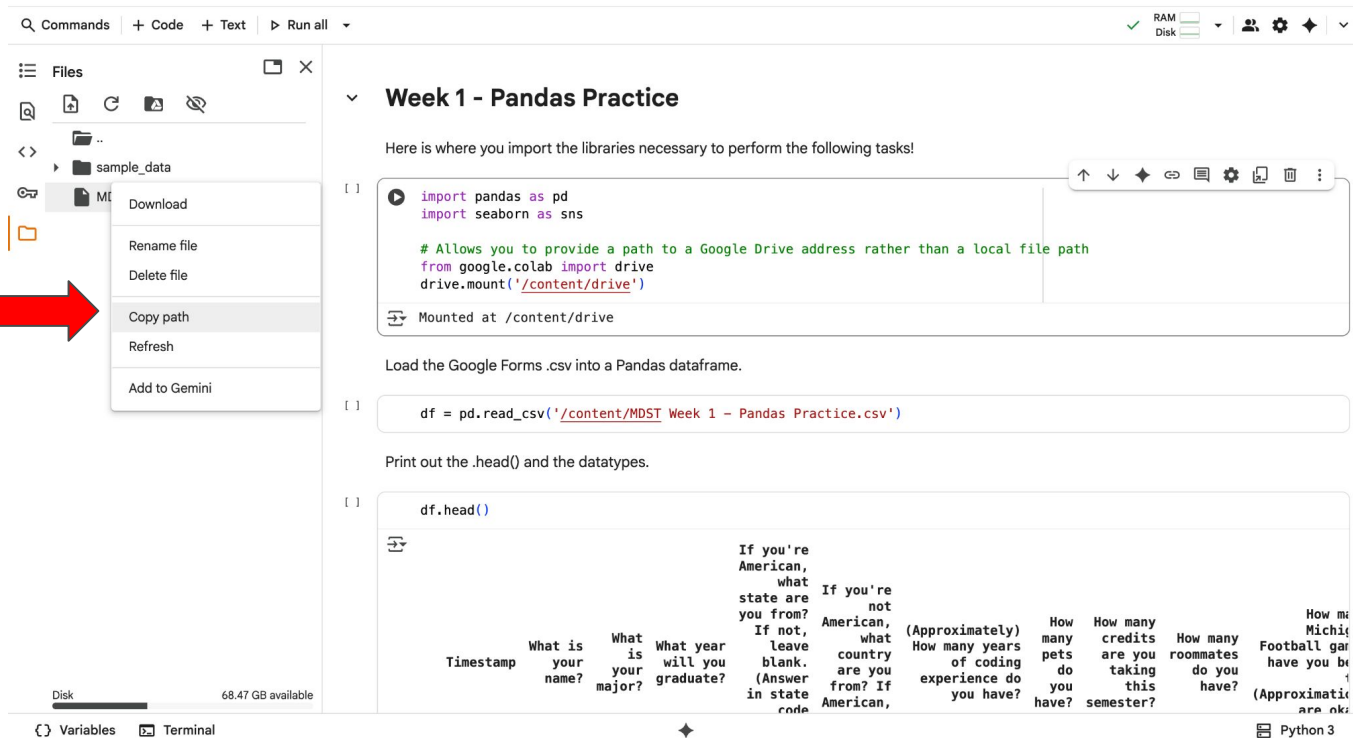
Below this, a prompt says 'Print out the .head() and the datatypes.' The final code block shows the command to print the first few rows of the dataframe:

```
df.head()
```

The output of the code is a table with columns: Timestamp, What is your name?, What is your major?, What year will you graduate?, If you're American, what state are you from?, If not, leave blank. (Answer in state code), If you're not American, what country are you from? If American, (Approximately) How many years of coding experience do you have?, How many pets do you have?, How many credits are you taking this semester?, How many roommates do you have?, Football game have you been to?, and How many Michigan are you not? The output shows the first few rows of data.

How to Upload Files into Colab

Click the three dots and copy the path. Put this in your read function



The screenshot shows the Google Colab interface. On the left, the 'Files' pane displays a folder named 'sample_data' containing a file 'MDST Week 1 - Pandas Practice.csv'. A context menu is open for this file, with the 'Copy path' option highlighted. A red arrow points from the text instruction to this option. The main area shows a code cell titled 'Week 1 - Pandas Practice' with the following code:

```
import pandas as pd
import seaborn as sns

# Allows you to provide a path to a Google Drive address rather than a local file path
from google.colab import drive
drive.mount('/content/drive')

Mounted at /content/drive
```

Below the code cell, text instructions state: 'Load the Google Forms .csv into a Pandas dataframe.' and 'Print out the .head() and the datatypes.' The next code cell contains:

```
df = pd.read_csv('/content/MDST Week 1 - Pandas Practice.csv')

df.head()
```

The output of the second code cell shows the first five rows of the CSV data, which are survey questions and answers. The bottom of the interface shows the 'Variables' and 'Terminal' tabs, with 'Python 3' selected.

How to Mount Your Google Drive

```
from google.colab import drive

drive.mount('/content/drive')

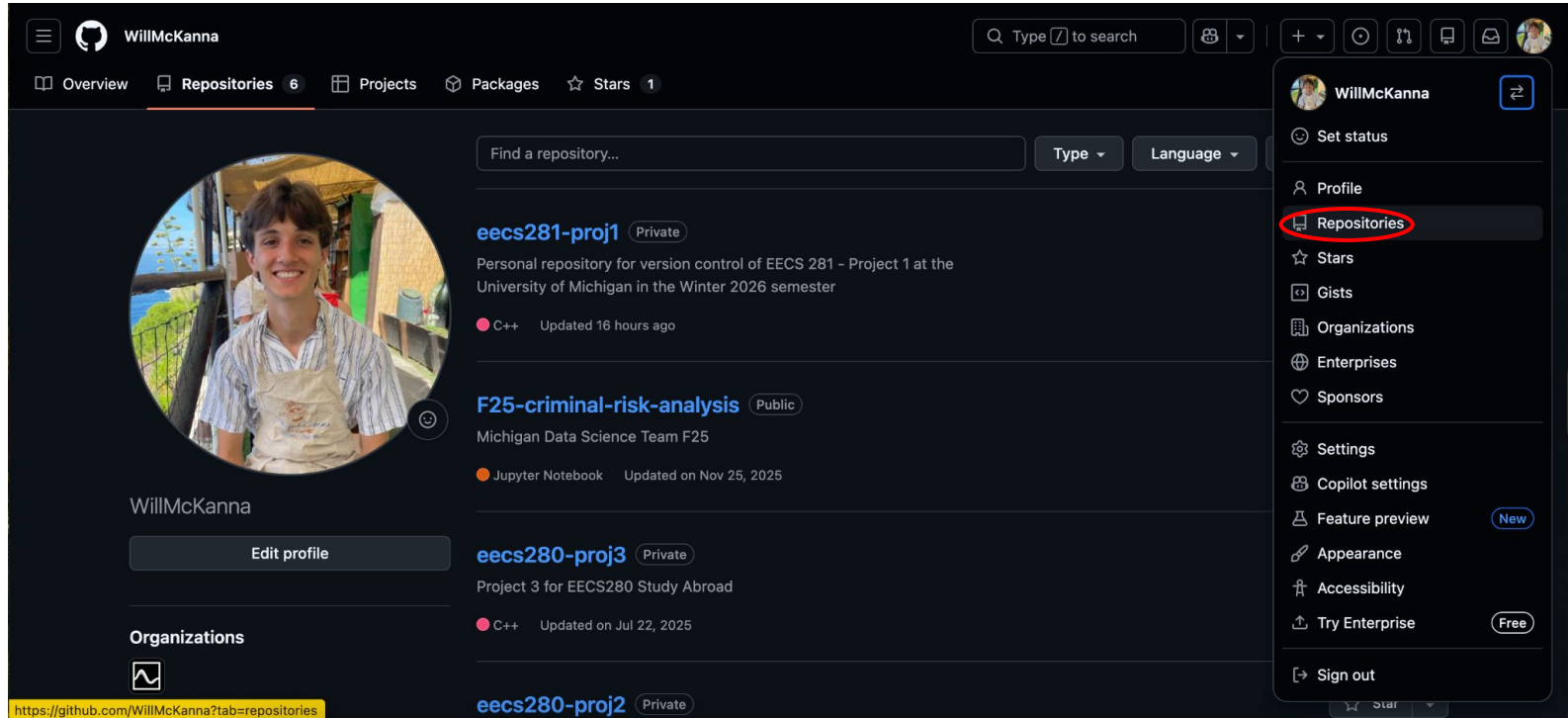
pd.read_csv('/content/drive/MyDrive/[FILE NAME]')
```

****NOTE:** If you saved your dataset in a folder (not just loose in your MyDrive folder), you will need to add further code to the file path before the file name in the final line. For example, if I saved my data called “alzheimers.csv” in a folder called “MDST-W26,” my read_csv statement would read:

```
pd.read_csv('/content/drive/MyDrive/MDST-W26/alzheimers.csv')
```

How to Create a GitHub Repository (yay for version control!)

- I. Navigate to GitHub, click your profile picture, and select “Repositories.” Then, click the green “New” button next to the “Language” dropdown menu



The screenshot shows the GitHub profile page for user WillMcKanna. The 'Repositories' tab is selected in the top navigation bar and highlighted with a red circle in the right-hand sidebar. The profile section on the left includes a circular profile picture of a young man, the name 'WillMcKanna', and an 'Edit profile' button. Below the profile picture, the 'Organizations' section is visible. The main content area displays a list of repositories: 'eecs281-proj1' (Private, C++, updated 16 hours ago), 'F25-criminal-risk-analysis' (Public, Michigan Data Science Team F25, Jupyter Notebook, updated on Nov 25, 2025), 'eecs280-proj3' (Private, C++, updated on Jul 22, 2025), and 'eecs280-proj2' (Private). The right-hand sidebar contains a dropdown menu with options: 'Set status', 'Profile', 'Repositories' (highlighted with a red circle), 'Stars', 'Gists', 'Organizations', 'Enterprises', 'Sponsors', 'Settings', 'Copilot settings', 'Feature preview' (with a 'New' badge), 'Appearance', 'Accessibility', 'Try Enterprise' (with a 'Free' badge), and 'Sign out'.

WillMcKanna

Overview Repositories 6 Projects Packages Stars 1

Find a repository... Type Language

eecs281-proj1 (Private)
Personal repository for version control of EECS 281 - Project 1 at the University of Michigan in the Winter 2026 semester
C++ Updated 16 hours ago

F25-criminal-risk-analysis (Public)
Michigan Data Science Team F25
Jupyter Notebook Updated on Nov 25, 2025

eecs280-proj3 (Private)
Project 3 for EECS280 Study Abroad
C++ Updated on Jul 22, 2025

eecs280-proj2 (Private)

WillMcKanna

Edit profile

Organizations

WillMcKanna

Set status

Profile

Repositories

Stars

Gists

Organizations

Enterprises

Sponsors

Settings

Copilot settings

Feature preview (New)

Appearance

Accessibility

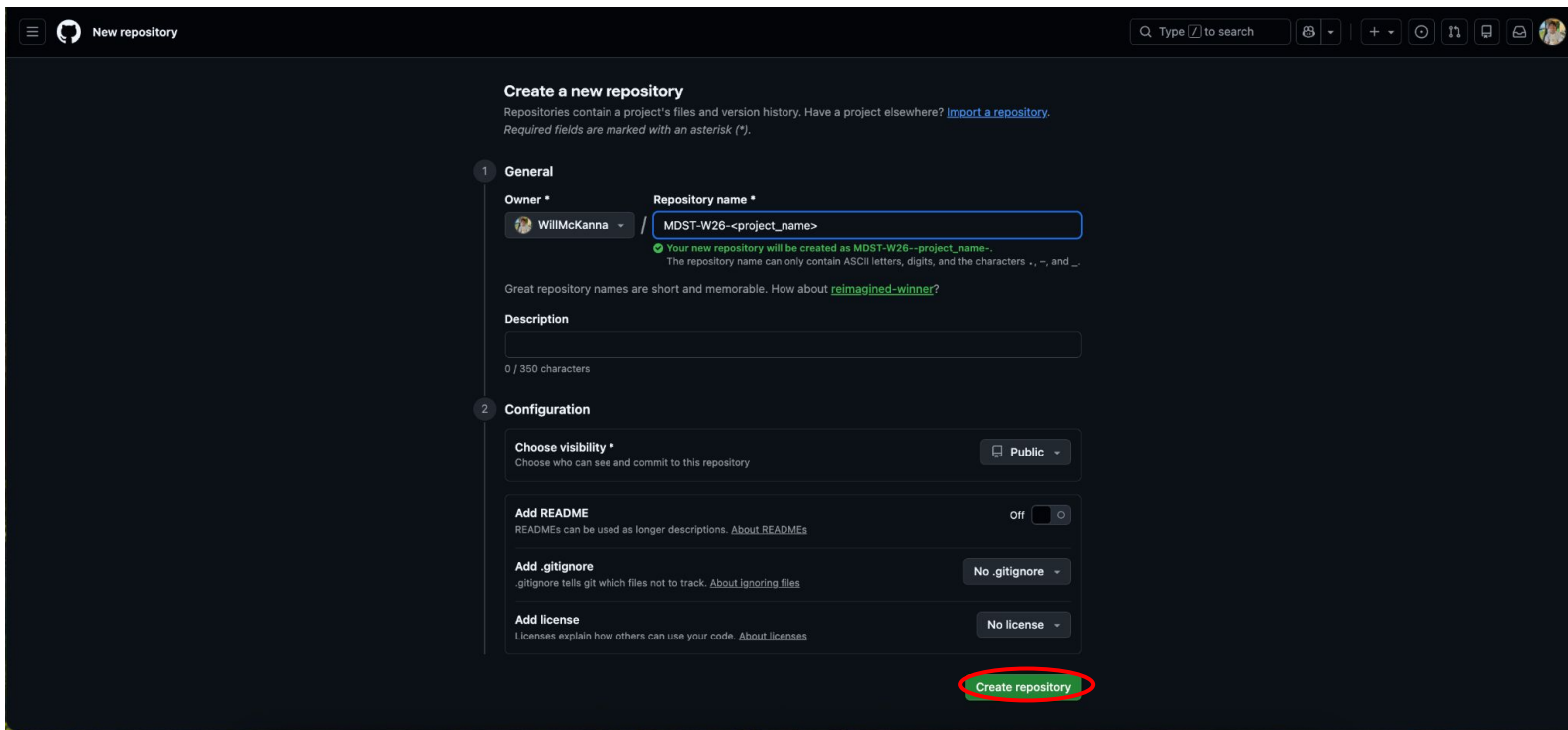
Try Enterprise (Free)

Sign out

<https://github.com/WillMcKanna?tab=repositories>

How to Create a GitHub Repository (yay for version control!)

2. Name your repository something memorable, then click “Create Repository”



Create a new repository

Repositories contain a project's files and version history. Have a project elsewhere? [Import a repository](#).
Required fields are marked with an asterisk (*).

1 General

Owner * WillMcKanna / **Repository name *** MDST-W26-<project_name>

✓ Your new repository will be created as MDST-W26-<project_name>-.
The repository name can only contain ASCII letters, digits, and the characters -, ., and _.

Great repository names are short and memorable. How about [reimagined-winner](#)?

Description

0 / 350 characters

2 Configuration

Choose visibility *
Choose who can see and commit to this repository Public

Add README
READMEs can be used as longer descriptions. [About READMEs](#) Off

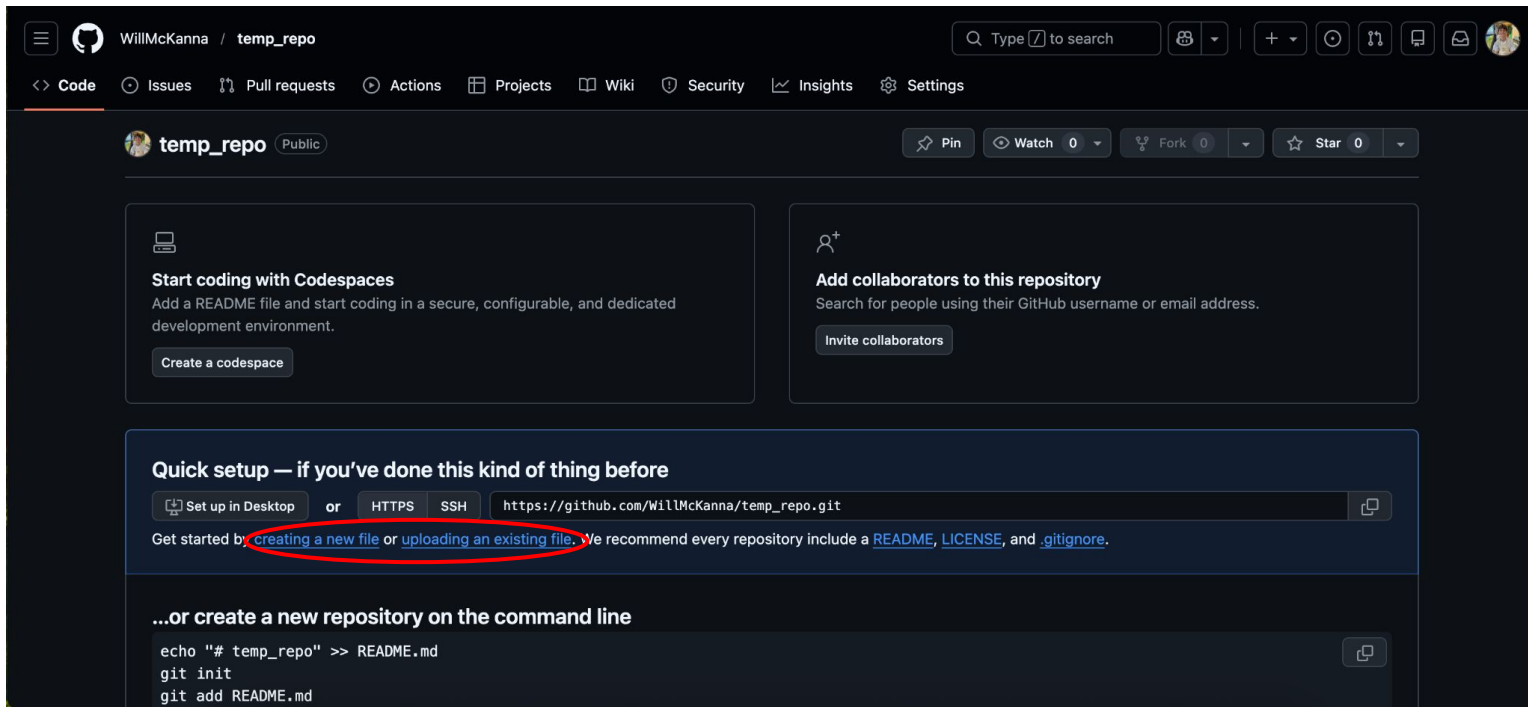
Add .gitignore
.gitignore tells git which files not to track. [About ignoring files](#) No .gitignore

Add license
Licenses explain how others can use your code. [About licenses](#) No license

Create repository

How to Create a GitHub Repository (yay for version control!)

3. Then, populate your repo either with your files, OR click “creating a new file”



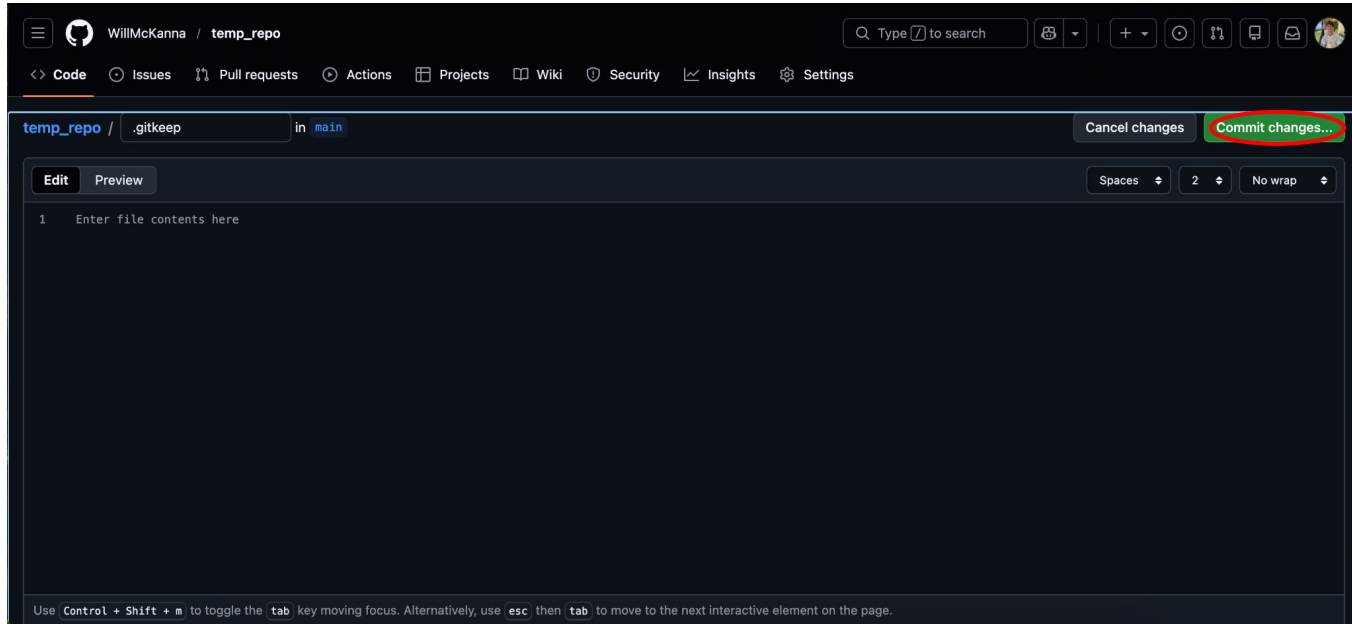
The screenshot shows the GitHub interface for a repository named 'temp_repo' by user 'WillMcKanna'. The repository is public. The page offers several options for getting started:

- Start coding with Codespaces:** A section with a laptop icon and text: "Add a README file and start coding in a secure, configurable, and dedicated development environment." Below this is a button labeled "Create a codespace".
- Add collaborators to this repository:** A section with a person icon and text: "Search for people using their GitHub username or email address." Below this is a button labeled "Invite collaborators".
- Quick setup — if you've done this kind of thing before:** A section with a dark blue background. It shows three options: "Set up in Desktop", "HTTPS", and "SSH". The "SSH" option is selected, and the URL "https://github.com/WillMcKanna/temp_repo.git" is displayed. Below this, it says "Get started by creating a new file or uploading an existing file. We recommend every repository include a [README](#), [LICENSE](#), and [.gitignore](#)." The phrase "creating a new file or uploading an existing file." is circled in red.
- ...or create a new repository on the command line:** A section with a dark blue background showing the following commands:

```
echo "# temp_repo" >> README.md
git init
git add README.md
```

How to Create a GitHub Repository (yay for version control!)

3(b). If you click “create a new file,” entitle it “.gitkeep” and click “Commit changes”



How to Create a GitHub Repository (yay for version control!)

4. Your repo is made! Continue adding to it by clicking the “Add file” dropdown menu

The screenshot shows the GitHub interface for a repository named 'temp_repo' by user 'WillMcKanna'. The repository is public. The top navigation bar includes links for Code, Issues, Pull requests, Actions, Projects, Wiki, Security, Insights, and Settings. Below the repository name, there are buttons for Pin, Watch (0), Fork (0), and Star (0). The main content area shows the 'main' branch with 1 branch and 0 tags. A search bar 'Go to file' is present. The 'Add file' dropdown menu is circled in red. Below this, there is a commit history section showing a commit by 'WillMcKanna' titled 'Create .gitkeep' with a timestamp of 'a168bb7 · now' and '1 Commit'. Below the commit history, there is a section for the README file, which is currently empty. The README section has a heading 'Add a README' and a button 'Add a README'. The right sidebar contains sections for 'About' (No description, website, or topics provided), 'Activity' (0 stars, 0 watching, 0 forks), 'Releases' (No releases published, Create a new release), and 'Packages' (No packages published, Publish your first package).