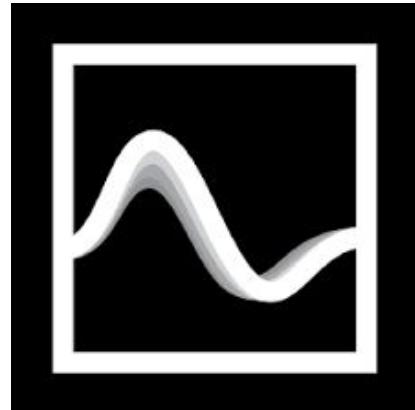


Week 3: Feature Engineering and Intro to ML



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?

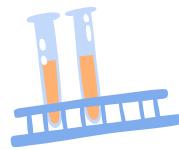
- Improves Model Accuracy
 - Choosing the right features helps the model learn better
 - Better learning, better predictions!
- Reduces Overfitting
 - By using fewer features that hold more importance, we prevent the model from memorizing the data on which we train it
- Boosts Interpretability!!
 - When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



03

Intro to Machine Learning

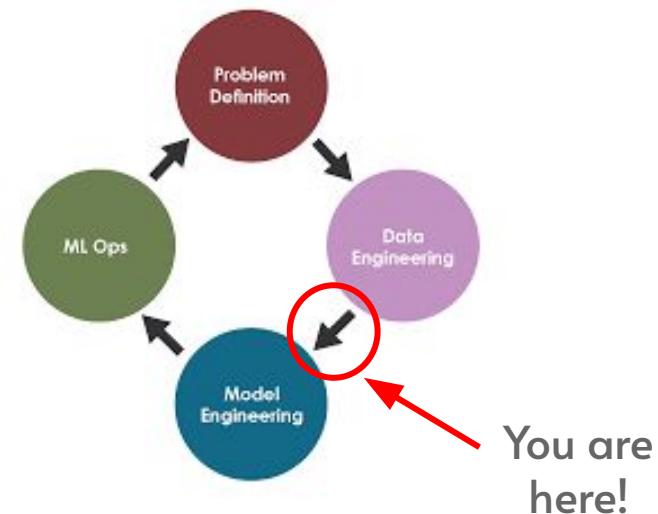
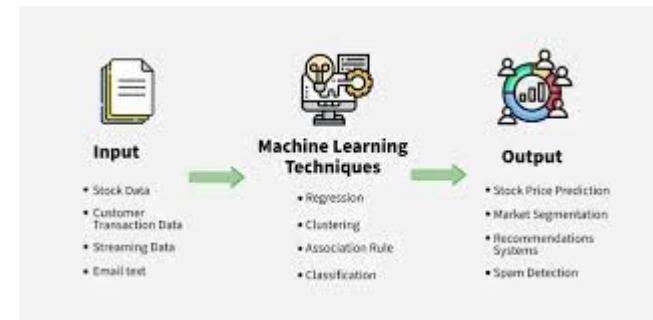
What is machine learning, what can we do with it, and what kinds of models can we use?





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.





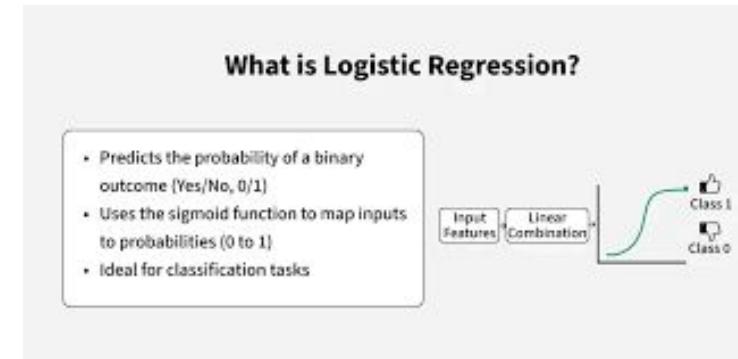
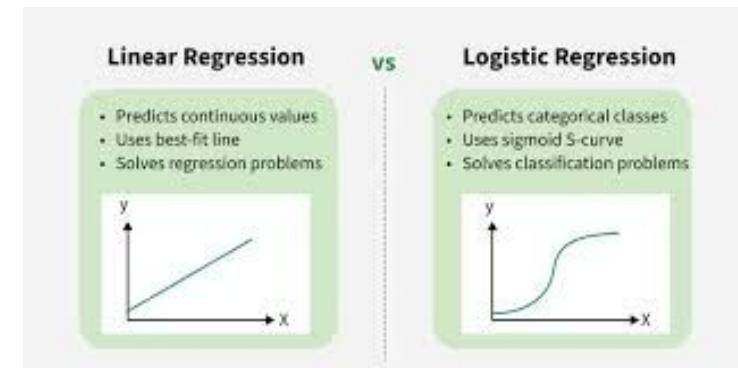
The Supervised Machine Learning Process

- **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

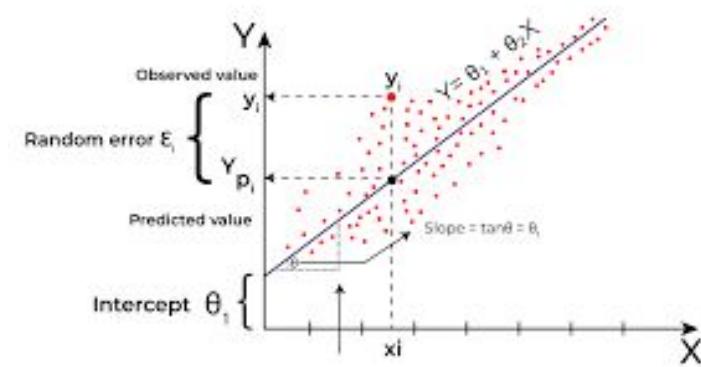
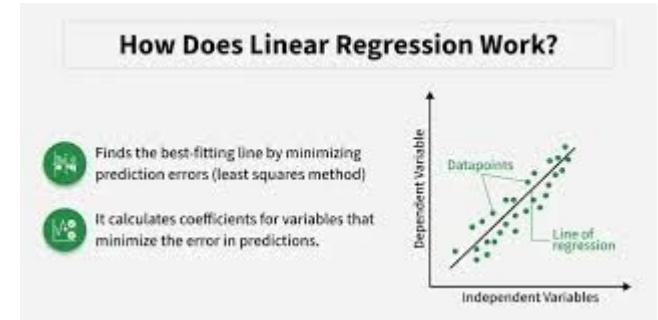
- **Binary logistic regression** works well in cases where we have a *binary target variable*
 - Ex. Stroke/No Stroke, Lung Cancer/None
- Considered a classification algorithm, as it attempts to fit points along a sigmoid curve ranging from [0, 1] for our binary target
- Models the **log-odds** ($\ln(\text{odds of an event})$) 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.





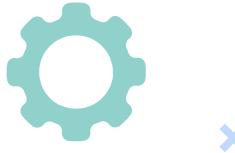
ML Algorithms – (Multiple) Linear Regression

- 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





0
4



References

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





Our Demo! (for reference)

The screenshot shows a Streamlit application interface. On the left is a dark sidebar with a navigation menu. The menu items are: 'main' (which is highlighted in a light gray bar), 'about data', 'about model', 'data analysis', 'model analysis', and 'predictions'. To the right of the sidebar is the main content area. At the top of the content area, there is a title 'MDST W26 'Building Interpretable AI in Healthcare'' in large white font. Below the title, it says 'By Will McKanna and Seena Simkani'. Underneath that, there is a descriptive text: 'Analyzing data from a synthetic dataset of Alzheimer's patients. A simple Random Forest model is used to predict if an individual has Alzheimer's given a set of cognitive diagnostics/assessments.' At the bottom of the content area, there is a section titled 'Project Description' in bold white font. Below this section, there is a detailed paragraph of text in a smaller white font. At the very bottom of the page, there are two small circular icons: one with a globe and another with a crown.

main

about data
about model
data analysis
model analysis
predictions

MDST W26 'Building Interpretable AI in Healthcare'

By Will McKanna and Seena Simkani

Analyzing data from a synthetic dataset of Alzheimer's patients. A simple Random Forest model is used to predict if an individual has Alzheimer's given a set of cognitive diagnostics/assessments.

Project Description

This project explores the intersection of Machine Learning interpretability and healthcare analytics by examining various datasets. This project teaches core data science skills, including data cleaning/pre-processing, Exploratory Data Analysis (EDA), and machine learning. Through the development of predictive models, members will investigate the extent to which patient factors influence model outcomes using SHAP (SHapley Additive exPlanations) and Principal Component Analysis (PCA), among other methods. Members will also learn about quantifying model uncertainty and controlling risk using MAPIE. This project aims to develop an understanding of statistical patterns, model interpretability, and the broader implications of deploying predictive algorithms in clinical settings.



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



The screenshot shows the Google Colab interface with the following elements:

- Top Bar:** Includes "Commands", "+ Code", "+ Text", and "Run all" buttons.
- File Sidebar:** Shows a "Files" section with a folder icon, a file icon, and a refresh icon. A red arrow points to the folder icon. Below it is a "sample_data" folder.
- Code Editor:** A section titled "Week 1 - Pandas Practice" contains 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
- Output:** Displays the output of the code execution, showing the first few rows of a CSV file named "MDST Week 1 - Pandas Practice.csv".

```
df.head()
```

| Timestamp | What is your name? | What is your major? | What is your graduate? | What year will you leave blank. | If you're American, what state are you from? If not, leave blank. | If you're American, what country are you from? If not, leave blank. | (Answer in state code) | (Approximately) How many years of coding experience do you have? If you're American, what country are you from? If not, leave blank. | How many pets do you have? If you're American, what country are you from? If not, leave blank. | How many credits are you taking? If you're American, what country are you from? If not, leave blank. | How many roommates do you have? If you're American, what country are you from? If not, leave blank. | Football games have you been to? | How many have you been to? | Michigan |
|---------------------|--------------------|---------------------|------------------------|---------------------------------|---|---|------------------------|--|--|--|---|----------------------------------|----------------------------|----------|
| 2023-10-01 14:30:00 | John Doe | Computer Science | Graduate | Blank | Michigan | United States | Python | Approximately 1 year | 1 pet | 15 credits | 2 roommates | 1 football game | 1 | Michigan |
- Bottom Navigation:** Includes "Variables", "Terminal", "Python 3", and disk usage information ("Disk 68.47 GB available").



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, there's a sidebar titled 'Files' with a red arrow pointing to the 'Upload' button (a cloud icon). Below it, there's a folder named 'sample_data' and a file named 'MDST Week 1 - Pandas Practice.csv'. The main area has a section titled 'Week 1 - Pandas Practice' with instructions and code. The code cell contains:

```
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 cell, it says 'Mounted at /content/drive'. The next section is 'Load the Google Forms .csv into a Pandas dataframe.' followed by the code:

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

Then, it says 'Print out the .head() and the datatypes.' followed by the code:

```
df.head()
```

The output of the 'df.head()' command shows a table of data:

| Timestamp | What is your name? | What is your major? | What is your graduate? | If you're American, what state are you from? If not, what country are you from? If answer from? If in state code | (Approximately) How many years of coding experience do you have? If you're American, what country are you from? If answer from? If in state code | How many pets do you have? If you're American, what country are you from? If answer from? If in state code | How many credits are you taking this semester? If you're American, what country are you from? If answer from? If in state code | How many roommates do you have? If you're American, what country are you from? If answer from? If in state code | Football games have you been to? If you're American, what country are you from? If answer from? If in state code | How many are you taking this semester? If you're American, what country are you from? If answer from? If in state code | Football games have you been to? If you're American, what country are you from? If answer from? If in state code |
|---------------------|--------------------|---------------------|------------------------|--|--|--|--|---|--|--|--|
| 2023-09-01 12:00:00 | John Doe | Computer Science | Graduate | US | 1 year | 1 pet | 1 credit | 1 roommate | 1 game | 1 game | 1 game |

At the bottom, it shows 'Disk 68.47 GB available' and tabs for 'Variables' and 'Terminal'.



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, there's a sidebar titled 'Files' with a 'Mount' section. A red arrow points to the three-dot menu next to 'Mount'. The menu options include 'Download', 'Rename file', 'Delete file', 'Copy path', 'Refresh', and 'Add to Gemini'. Below the sidebar, the main workspace shows a code cell for 'Week 1 - Pandas Practice'. It contains code to import pandas and seaborn, and to mount Google Drive. A note explains that drive.mount('/content/drive') provides a path to a Google Drive address. Another code cell shows df.read_csv('MDST Week 1 - Pandas Practice.csv'). A third code cell displays df.head(), showing a sample of a survey dataset with columns like 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?, How many football games have you been to?, and How many Michigan games have you been to?. At the bottom, it says '(Approximate answers are ok.)'. The bottom navigation bar includes 'Variables', 'Terminal', and 'Python 3'.

```
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')

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

df.head()
```

If you're American, what state are you from? If not, leave blank. (Answer in state code)

| 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? | How many football games have you been to? | How many Michigan games have you been to? |
|--------------------------|--------------------|---------------------|------------------------------|--|--|----------------------------|--|---------------------------------|---|---|
| 2023-09-18 14:45:23.456Z | John Doe | Computer Science | 2025 | United States | United States | 1 | 15 | 1 | 1 | 1 |



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 a GitHub user profile for "WillMcKanna". The profile picture is a circular photo of a smiling person wearing an apron. The user has 6 repositories, 1 star, and no packages. A search bar at the top allows searching for repositories by name, type, or language. Below the search bar, three repositories are listed:

- eecs281-proj1** [Private] - Personal repository for version control of EECS 281 - Project 1 at the University of Michigan in the Winter 2026 semester. Last updated 16 hours ago.
- F25-criminal-risk-analysis** [Public] - Michigan Data Science Team F25. Last updated on Nov 25, 2025.
- eecs280-proj3** [Private] - Project 3 for EECS280 Study Abroad. Last updated on Jul 22, 2025.

At the bottom left, there's an "Edit profile" button. On the right side, a sidebar menu is open, showing options like Profile, Set status, and Repositories. The "Repositories" option is highlighted with a red circle. Other options include Stars, Gists, Organizations, Enterprises, Sponsors, Settings, Copilot settings, Feature preview, Appearance, Accessibility, Try Enterprise, and 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”

The screenshot shows the GitHub interface for creating a new repository. At the top, there's a navigation bar with icons for search, issues, pull requests, and user profile. Below it is the main title "Create a new repository". A sub-instruction says "Repositories contain a project's files and version history. Have a project elsewhere? Import a repository." and notes that required fields are marked with an asterisk (*).

The form is divided into two sections:

- General**:
 - Owner: WillMcKenna
 - Repository name: MDST-W26--project_name (highlighted with a red box)
 - Description: Great repository names are short and memorable. How about [reimagined-winer](#)?
 - Visibility: Public (selected)
- Configuration**:
 - Choose visibility: Public (selected)
 - Add README: Off (switch is off)
 - Add .gitignore: No .gitignore (selected)
 - Add license: No license (selected)

At the bottom right of the configuration section, the "Create repository" button is highlighted with a red circle.

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”

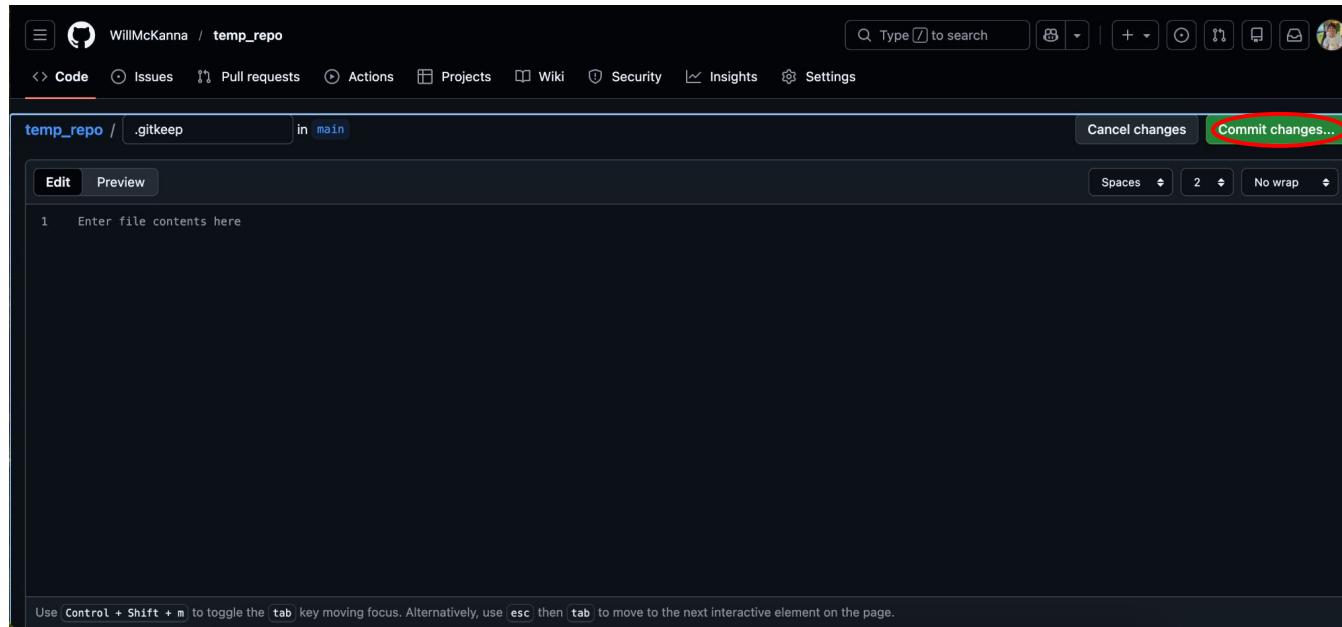
The screenshot shows a GitHub repository page for 'temp_repo' owned by 'WillMcKanna'. The page includes sections for 'Start coding with Codespaces', 'Add collaborators to this repository', and a 'Quick setup' section. The 'Quick setup' section contains links for 'Set up in Desktop', 'HTTPS', 'SSH', and a URL 'https://github.com/WillMcKanna/temp_repo.git'. A red circle highlights the link 'Get started by creating a new file or uploading an existing file.' Below this, there's a command-line guide for creating a new repository.

Start coding with Codespaces
Add collaborators to this repository
Quick setup — if you've done this kind of thing before
Get started by [creating a new file or uploading an existing file.](#) We recommend every repository include a [README](#), [LICENSE](#), and [.gitignore](#).
...or create a new repository on the command line

```
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 a GitHub repository page for 'temp_repo'. The top navigation bar includes links for Code, Issues, Pull requests, Actions, Projects, Wiki, Security, Insights, and Settings. Below the navigation is a search bar and various repository stats: main branch, 1 Branch, 0 Tags, 0 forks, 0 stars, and 0 commits. A red circle highlights the 'Add file' dropdown menu in the top right corner of the main content area. The main content area displays a commit from 'WillMcKanna' titled 'Create .gitkeep' with a timestamp of 'a168bb7 · now'. Below the commit is a section for the 'README' file, which currently contains a 'Create a README' button. To the right of the repository stats, there are sections for 'About', 'Releases', and 'Packages', each with their respective status and links.

WillMcKanna / temp_repo

Code Issues Pull requests Actions Projects Wiki Security Insights Settings

temp_repo Public

main 1 Branch 0 Tags

Go to file Add file

WillMcKanna Create .gitkeep a168bb7 · now 1 Commit

.gitkeep Create .gitkeep now

README

Add a README

Help people interested in this repository understand your project.

Add a README

About

No description, website, or topics provided.

Activity

0 stars

0 watching

0 forks

Releases

No releases published

Create a new release

Packages

No packages published

Publish your first package