PrepVector
UPSCALE YOUR CAREER

WEEK 1: AI/ML PROJECTS FOR DATA PROFESSIONALS

PrepVector
UPSCALE YOUR CAREER

Welcome!

Welcome to the AI/ML Project course!

Professional Background

Manisha Arora

I am a seasoned Data Science professional with 10+ years of experience leading data science teams and driving business growth through data-driven decision making. I am passionate about democratizing data science and enabling others level up in their careers.

CAREER HIGHLIGHTS

- **Google**
  Data Science Lead - Ads Business & Marketing
- **PrepVector**
  Founder
- **Axtria - Ingenious Insights**
  Manager, Data Science & Machine Learning
- **Novartis**
  Senior Analyst, Global Business Services

ACADEMIC BACKGROUND

- **MIT**
  Instructor - ML & AI
- **University of Texas**
  Instructor - ML & AI
- **University of Cincinnati**
  Masters, Business Analytics
- **NIT, Kurukshetra**
  Bachelors, Electrical Engineering

TABLE OF CONTENT

PrepVector
UPSCALE YOUR CAREER

1. COURSE MODULES

About the Course & Modules

COURSE MODULES

WEEK 1

Overview of ML, Github Setup, and Coding Best Practices

WEEK 2

[Project: Uber ETA Prediction] Data Exploration, Feature Engineering, ML model, Model Deployment

WEEK 3

[Project: Demand Forecasting] Visualization, ML Modeling, Insight Generation, Model Deployment

WEEK 4

[Project: Speech Translation] AI Exploration, Speech Transcription, AI ethics and challenges

WEEK 5

Build Github Portfolio, Organize projects, Showcase your Work

SCHEDULE & EXPECTATIONS

SCHEDULE

Live Sessions
Sundays, 11.00AM - 1.00PM EST

[Optional] Office Hours
Thursdays, 8.30-9.30PM EST

Mentors:
Siddarth: Principal DS Manager, Microsoft
Siva: Lead DS, Microsoft

Course Expectations

1. This course does not teach basics of Python programming, but there are resources on the portal to brush up on **pandas data manipulation**.

2. This is a **hands-on course**. You are expected to spend time throughout the week to work on projects.

3. **Participate** during the sessions, engage in discussions, and ask questions.

4. Connect with, help out, and learn from each other. I encourage a **collaborative environment** as we work together over the next 5 weeks.

PrepVector
UPSCALE YOUR CAREER

2. MACHINE LEARNING OVERVIEW

ML101 Walkthrough

PrepVector
UPSCALE YOUR CAREER

ML Process Overview

PrepVector
UPSCALE YOUR CAREER

ML Process Lifecycle

Machine Learning automates the process of pattern-discovery by finding meaningful insights from real-world or generated data. It is used to build systems that can learn hidden patterns from data to aid in making intelligent decisions.

Work with the stakeholders to gather more context about the business problem, the “why” behind it, the current state or any existing solutions, and combine that with industry knowledge. Align on milestones, metrics, and timelines.

Gather data needed to answer the problem statement. Align on data timeframe and short/long term predictions. Sanity check the data, understand trends, visualize the data. Perform outlier and missing value imputation. Design necessary features.

Depending on the question and nature of model, choose training method.
Start with a simple model and iterate.
Train the model on training data and use various algorithms to recognize patterns in the data.

Align on model evaluation metric. Test out the model on validation data.
Analyze fit and tune the model based on performance.

Use inputs to derive insights and predictions. Align on model deployment infrastructure, refresh frequency, and set up monitoring alerts.

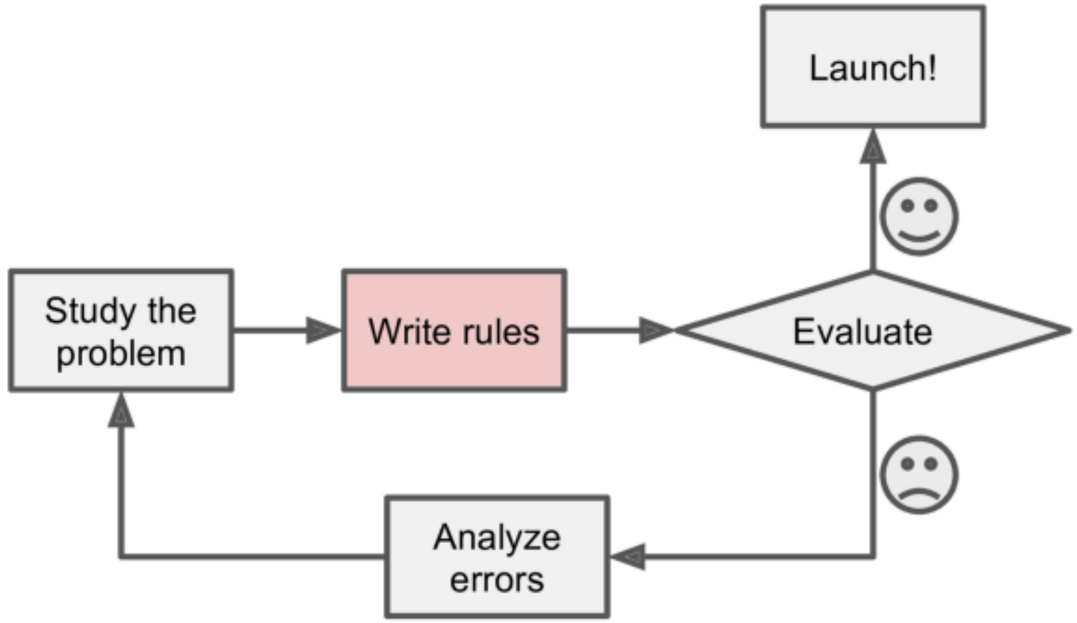
PrepVector
UPSCALE YOUR CAREER

Don’t be afraid to use heuristics

Heuristic is a simple and quickly implemented solution to a problem.
Example – If you need to rank documents, rank them alphabetically.
Example – If you want to detect spam, filter out businesses who have sent out spam before.

Start with Rule-based approach:

1. Understand the problem
2. Consult experts and refer to historical knowledge to design “rules”
3. Implement these rules as an algorithm
4. Evaluate and launch



Issues with Rule-based approach:

1. Can be labor-intensive
2. Cannot generalize to unanticipated input combinations
3. Don’t handle uncertainty / missing data

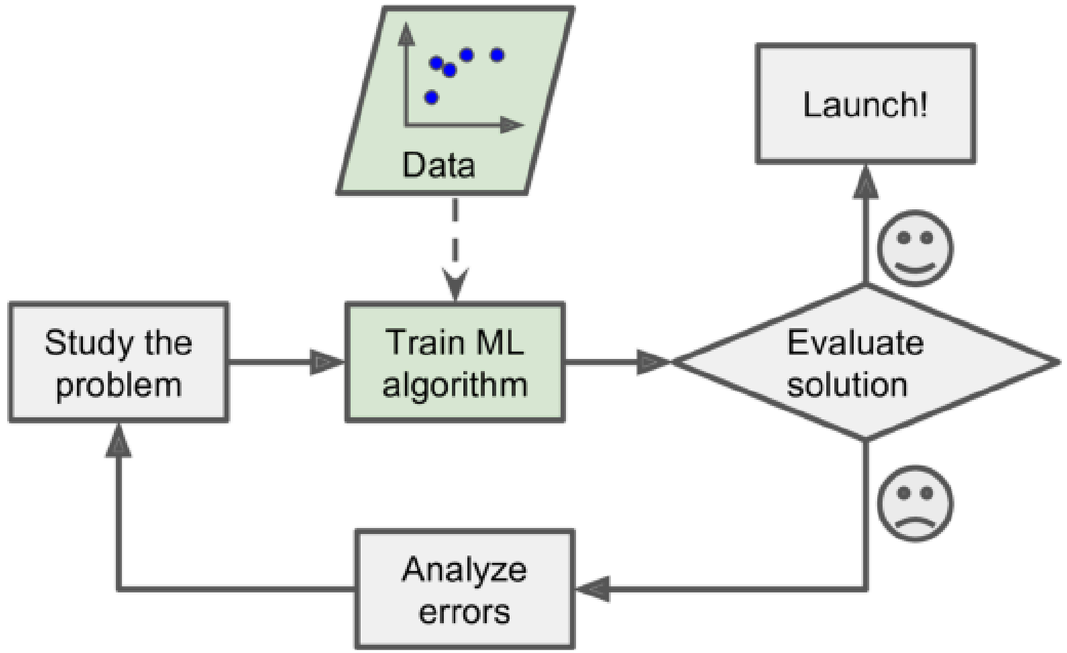
PrepVector
UPSCALE YOUR CAREER

Choose ML if heuristics get complicated

Machine Learning involves prediction of Y (output) given X (inputs).
ML helps find the best prediction function that can solve this problem.

Machine Learning based approach:

1. Learn from the training data
2. Predicts Y given a series of input features X
3. Is used to design larger systems or insights difficult to spot with simpler approaches



General Form of ML equation:
Quantitative response Y
Set of p predictors X1, X2, X3, … Xp

Y = f(X) + error
Where f is a fixed but unknown function of X1, X2, …, Xp
Error is independent of X and has mean zero

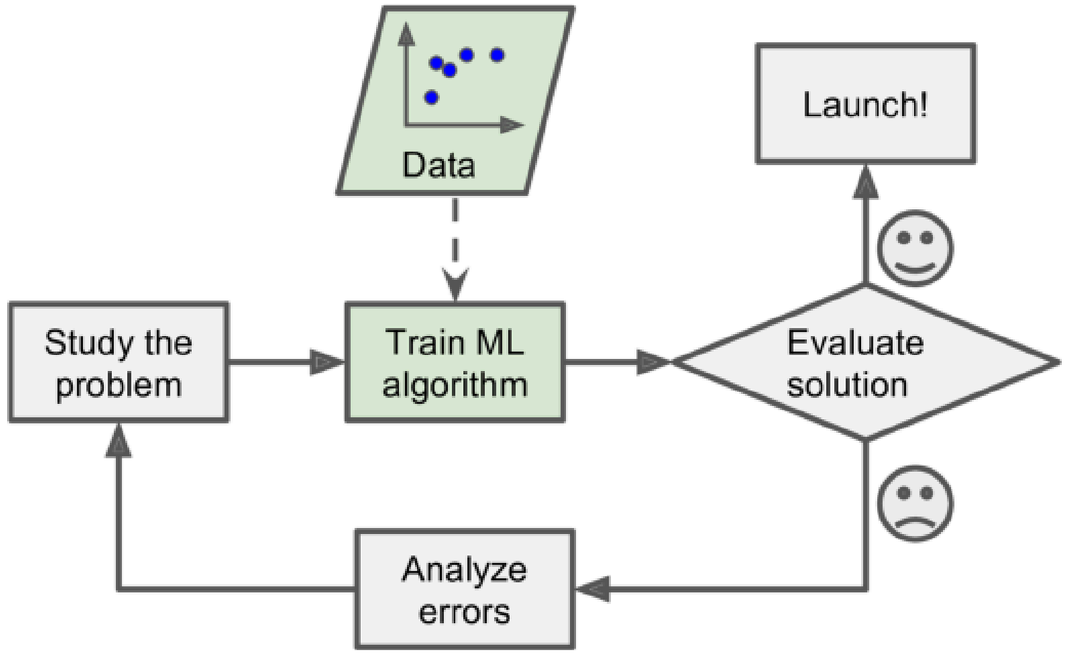
We estimate f(hat):
Y(hat) = fhat(X)
Yhat is a prediction for Y.
Fhat is not a perfect estimate of f, and hence, has errors - irreducible and reducible.

PrepVector
UPSCALE YOUR CAREER

Design systems keeping metrics in mind

Points to note for machine learning systems:

1. Understand if you need user permission to use this data
2. Align with key stakeholders on what we are building, why we are building, and how we plan to track it.
3. ML systems are iterative. Design systems that are dynamic in nature.
4. Infrastructure challenges are real. Test infra independent on the current model
5. Align on metrics and keep metric instrumentation in mind – not just the current state but also the future state!
6. Choose metrics that are easy to observe and attribute. Don't use machine learning to figure out "happiness", "satisfaction" or answer questions like "is the product improving user well-being" Instead use metrics like "downloads", "clicks" etc.
7. Tradeoff between accuracy and interpretability



PrepVector
UPSCALE YOUR CAREER

Data Preparation

PrepVector
UPSCALE YOUR CAREER

Data Preparation

Understanding and sanity checking data:

1. Align on what data to use
2. Explore data – understand shape, descriptions, data types, time trends, duplicates etc.
3. Analyze aggregate and segment level data to derive insights
4. Deal with missing data – detect nulls. Should nulls be present? If not, drop nulls or fill nulls. Check % of data being manipulated.
5. Deal with outliers – detect outliers. Should outliers be present? If not, remove outliers or cap outliers. Check % of data being manipulated.
6. Check class imbalance
7. Check correlations

Feature Engineering:

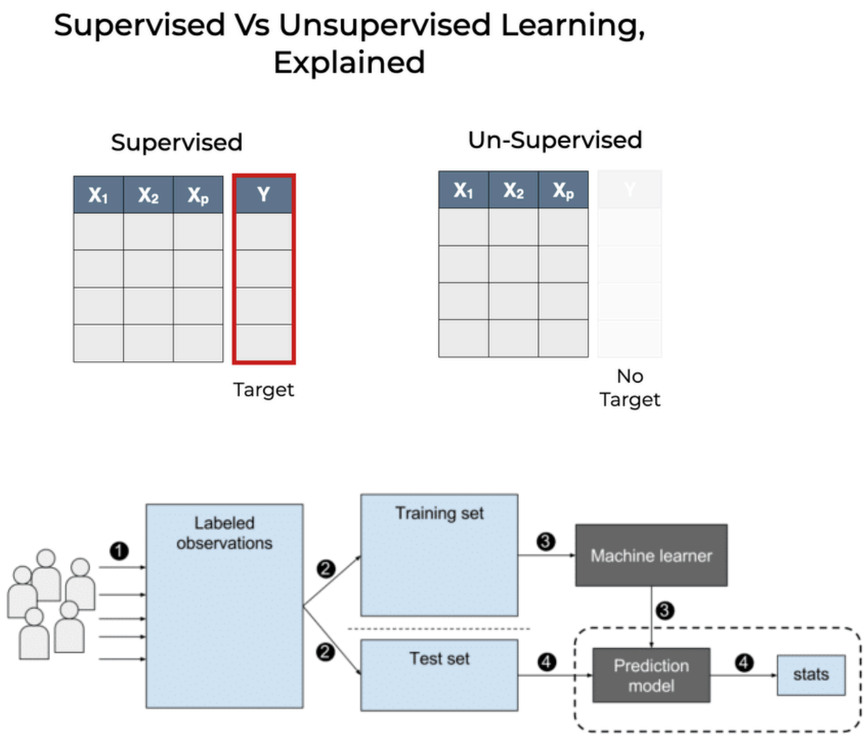
8. Scale data
9. Fix class imbalance – assign weights or balance the classes through oversampling / undersampling/ SMOTE
10. Create any new features from the original variables
11. Encode categorical variables
12. Split the data into training, validation, and testing

PrepVector
UPSCALE YOUR CAREER

Linear Models

PrepVector
UPSCALE YOUR CAREER

Types of learning



Supervised vs Unsupervised Learning:

Supervised Learning: Having a full set of labeled data while training an algorithm.
Fully labeled means that each example in the training dataset is tagged with the answer the algorithm should come up with on its own.

Clean, perfectly labeled datasets aren't easy to come by. And sometimes, researchers are asking the algorithm questions they don't know the answer to. That's where unsupervised learning comes in.

In unsupervised learning, the training dataset is a collection of examples without a specific desired outcome or correct answer. The model then attempts to automatically find structure in the data by extracting useful features and analyzing its structure.

Regression vs Classification:

Quantitative response: Regression problem
Qualitative response: Classification problem

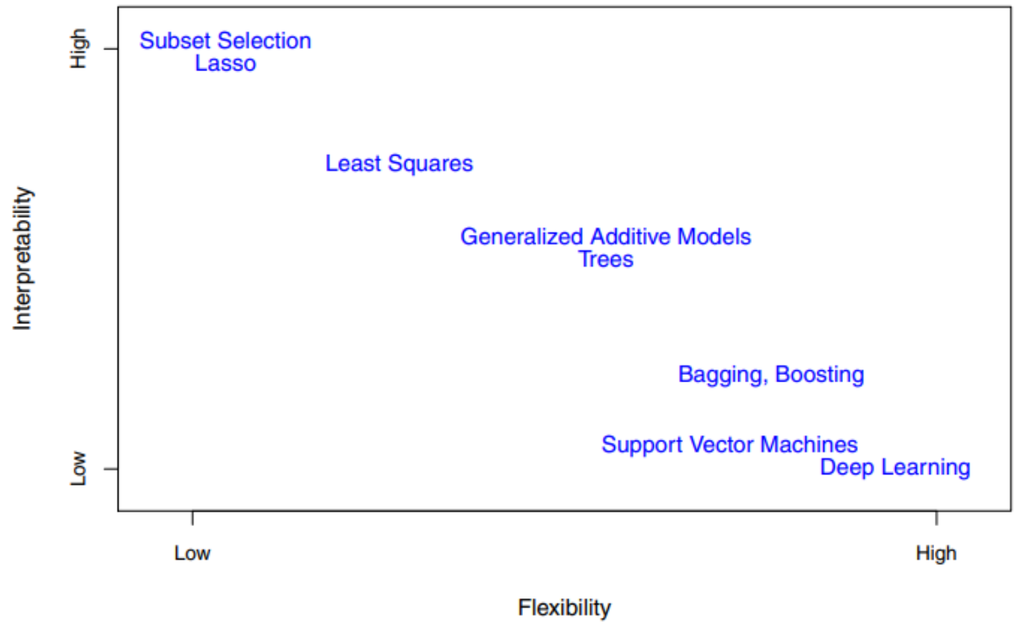
PrepVector
UPSCALE YOUR CAREER

Tradeoff between interpretability and flexibility

Linear models are the best model for inference and hence, a preferred choice in a lot of real-life projects where insights are used to drive business decision-making.
They are easy to understand the relationship between Y and X1, X2,...

GAMs extend linear models to allow for certain non-linear relationships, and hence are more flexible than linear model.

Non-linear models like bagging, boosting, and neural networks are highly flexible but very hard to interpret.



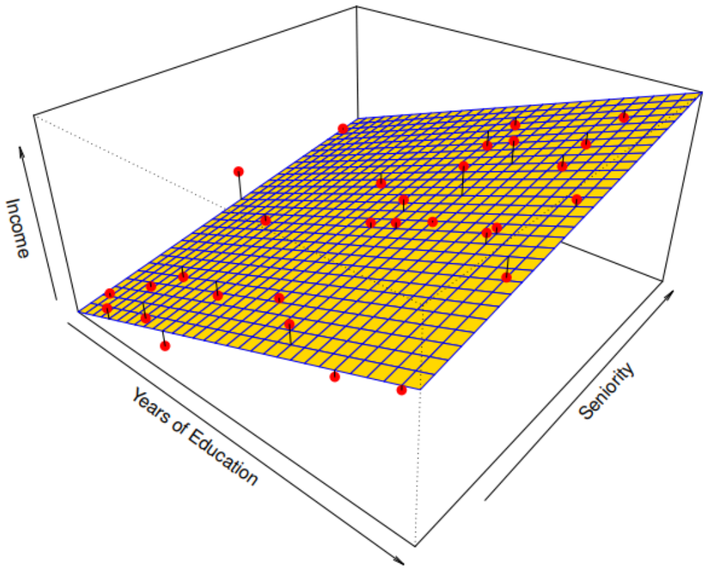
Representation of tradeoff between flexibility and interpretability.
More flexibility reflects a much wider range of possible shapes to estimate f.

PrepVector
UPSCALE YOUR CAREER

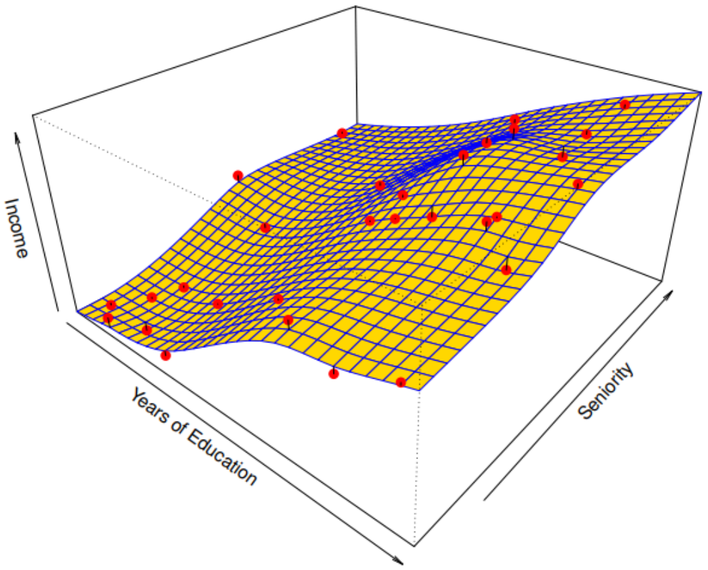
Linear Regression

A linear model fit by least squares to the Income data. This is a parametric approach because it makes assumptions about the underlying functional form or shape of f.

$$\texttt{income} \approx \beta_0 + \beta_1 \times \texttt{education} + \beta_2 \times \texttt{seniority}.$$



A linear model fit by least squares to the Income data



A non-linear model fit to the Income data

PrepVector
UPSCALE YOUR CAREER

Evaluation Metrics For Linear Regression

- Mean Squared Error: It is the average of the squared difference between the predicted and actual value. It has a convex shape and is easier to optimize. It penalizes large errors. $MSE = \frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i - y_i)^2$

- R-Squared: It measures the strength of the relationship between your model and the dependent variable. $R^2 = 1 - \frac{RSS}{TSS}$

- Adjusted R-Squared: Measures variation explained by only the independent variables that actually affect the dependent variable. $R^2_{adjusted} = [\frac{(1-R^2)(n-1)}{n-k-1}]$ n – number of data points; k – number of variables in your model

- Root Mean Squared Error (RMSE): This is the square root of the average of the squared residuals (difference of the predicted and actual value). RMSE measures the scatter of these residuals. $RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i - y_i)^2}$

- Mean Absolute Percentage Error (MAPE): Is one of the most commonly used KPIs to measure forecast accuracy. It's the sum of the individual absolute errors divided by the demand. It is the average of the percentage errors.

$$MAPE = \frac{1}{n}\sum\frac{|e_t|}{d_t}$$

PrepVector
UPSCALE YOUR CAREER

Logistic Regression

$$\Pr(\texttt{default} = \texttt{Yes}|\texttt{balance}). \qquad p(X) = \frac{e^{\beta_0+\beta_1 X}}{1+e^{\beta_0+\beta_1 X}}.$$

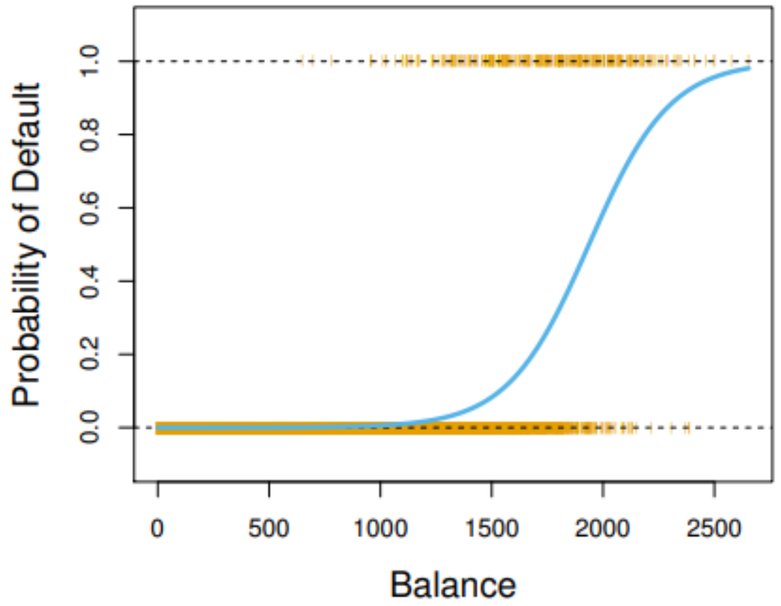
Logistic regression is well-suited for classifying two or more classes.
Here, we are computing a probability of defaulting given a balance.

Odds = p / (1-p)
Odds can lie between 0 and infinity

Log(odds) = B0 + B1X1 + …
Log(p/1-p) = B0 + B1X1 + …

Interpretation:
Increasing X1 by one unit changes log odds by B1. Or it multiplies the odds by e(B1)**

If B1 >0, then increasing X will increase p(X)
If B1 < 0, then increasing X will decrease p(X)



A non-linear model fit to the Income data

PrepVector
UPSCALE YOUR CAREER

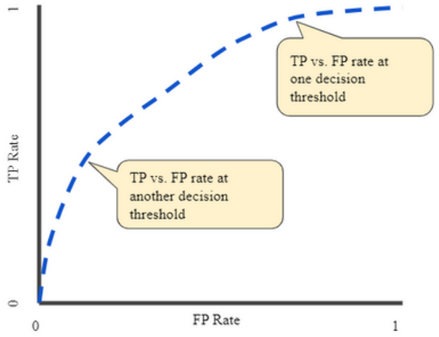
Evaluation Metrics for Logistic Regression

	Predicted Condition	
Actual Condition	True Positive (TP)	True Negative (TN)
	False Positive (FP) Type 1 Error	False Negative (FN) Type 2 Error

- **Precision**: TP / (TP + FP). Percentage of your results which are relevant
- **Recall**: TP / (TP + FN). Percentage of total relevant results correctly classified by your algorithm
- **F1 Score**: Harmonic mean of precision and recall. This is a simpler metric which takes into account for both precision and recall. You can aim to maximize this metric to make your model better

$$\text{F1 Score} = 2 * \frac{\text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}}$$

- **Accuracy**: TP + TN / Total = TP + TN / (TP + TN + FP + FN). This measures the total correction classifications of your model as a percentage of all classifications.
- **Receiver Operating Characteristic Curve (ROC):** Is a graph showing performance of a classification model at various classification thresholds
- **Area Under the ROC Curve (AUC)**: Measures the entire 2-dimensional area underneath the ROC curve (integral calculus from (0,0) to (1,1)



PrepVector
UPSCALE YOUR CAREER

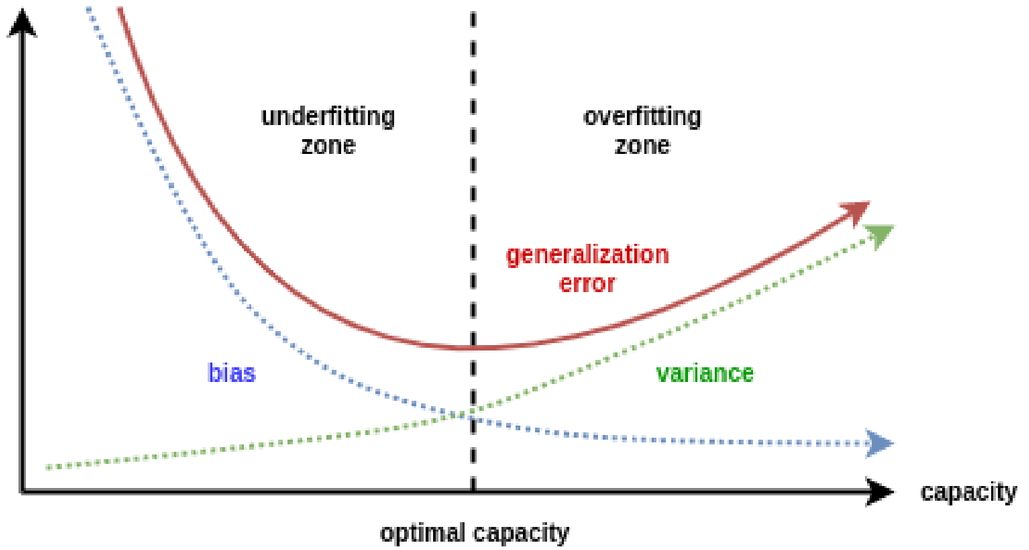
Shrinkage Models

PrepVector
UPSCALE YOUR CAREER

Bias Variance Tradeoff

Bias = Difference between avg prediction of the model and actual value being predicted.
Model with high bias oversimplifies the model and hence, leads to high error on both training and test data.

Variance = variability of model prediction for a given data point
Model with high variance also models the noise in the training data and hence, doesn't generalize well on the test data.



$$Err(x) = E\left[(Y - \hat{f}\,(x))^2\right]$$

$$Err(x) = \left(E[\hat{f}\,(x)] - f(x)\right)^2 + E\left[\left(\hat{f}\,(x) - E[\hat{f}\,(x)]\right)^2\right] + \sigma_e^2$$

$$Err(x) = \text{Bias}^2 + \text{Variance} + \text{Irreducible Error}$$

PrepVector
UPSCALE YOUR CAREER

Shrinkage Methods (1 of 2)

Ridge: $$\text{RSS} = \sum_{i=1}^{n}\left(y_i - \beta_0 - \sum_{j=1}^{p}\beta_j x_{ij}\right)^2.$$

$$\sum_{i=1}^{n}\left(y_i - \beta_0 - \sum_{j=1}^{p}\beta_j x_{ij}\right)^2 + \lambda\sum_{j=1}^{p}\beta_j^2 = \text{RSS} + \lambda\sum_{j=1}^{p}\beta_j^2,$$

In other words:

$$\underset{\beta}{\text{minimize}}\left\{\sum_{i=1}^{n}\left(y_i - \beta_0 - \sum_{j=1}^{p}\beta_j x_{ij}\right)^2\right\} \quad \text{subject to} \quad \sum_{j=1}^{p}\beta_j^2 \le s,$$

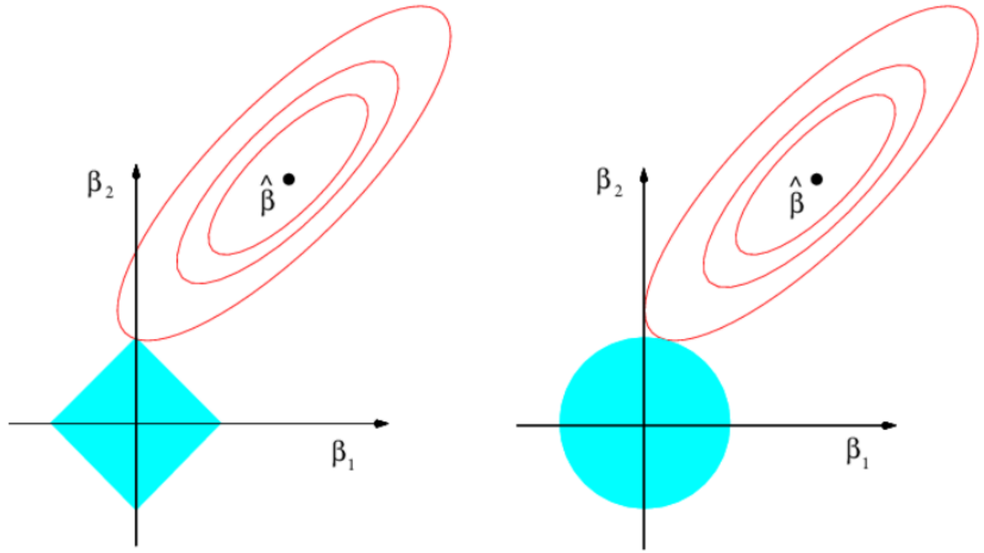
Lasso:

Lasso coefficients minimize the quantity:

$$\sum_{i=1}^{n}\left(y_i - \beta_0 - \sum_{j=1}^{p}\beta_j x_{ij}\right)^2 + \lambda\sum_{j=1}^{p}|\beta_j| = \text{RSS} + \lambda\sum_{j=1}^{p}|\beta_j|.$$

In other words,

$$\underset{\beta}{\text{minimize}}\left\{\sum_{i=1}^{n}\left(y_i - \beta_0 - \sum_{j=1}^{p}\beta_j x_{ij}\right)^2\right\} \quad \text{subject to} \quad \sum_{j=1}^{p}|\beta_j| \le s$$



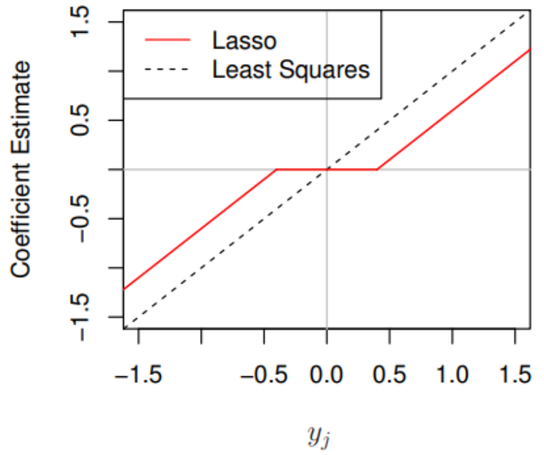
Red ellipse is the contour of RSS. Blue region is the constraint function for Lasso (left) and Ridge (right).

PrepVector
UPSCALE YOUR CAREER

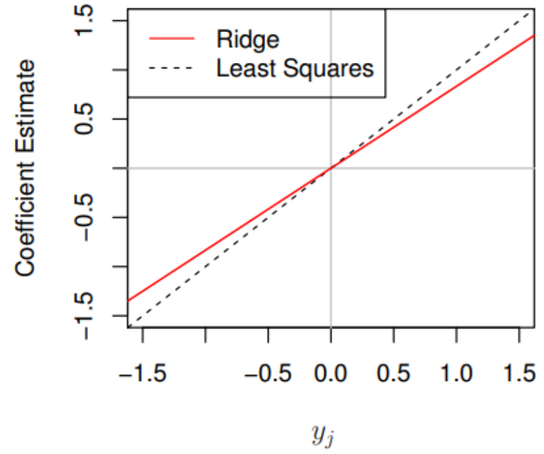
Shrinkage Methods (2 of 2)

When least square estimates have very high variance, shrinkage methods can yield a reduction in variance at the expense of a small increase in bias.
Both Lasso and Ridge shrink the coefficient estimates towards zero.

Lasso (L1 penalty): Forces some coefficients to be exactly zero, hence performing variable selection. Hence, model involves only a subset of variables and is easier to interpret.
It performs better where a small number of predictors have substantial coefficients and the remaining ones are either zero or very small.



Ridge (L2 penalty): Shrinks the coefficients towards zero but not exactly zero. Hence, it contains all the variables fed into the model.
It performs better when the response is a function of many predictors, with coefficients roughly the same size.



However, number of predictors related to the response is not known beforehand. Hence, cross-validation is used to determine which approach is better on a particular dataset.

PrepVector
UPSCALE YOUR CAREER

Tree Based Models

PrepVector
UPSCALE YOUR CAREER

Tree based Methods

Trees are based on a top-down, greedy approach known as recursive binary splitting:

1. We divide the predictor soace (set of possible values for X1, X2, ..., Xp) into distinct overlapping regions.
2. For every observation that falls in a particular region, we make the same prediction:
   - mean of all response values for the training observations in that region, or
   - majority vote of all response values for the training observations in that region

   Goal is to find regions that minimize RSS given by:

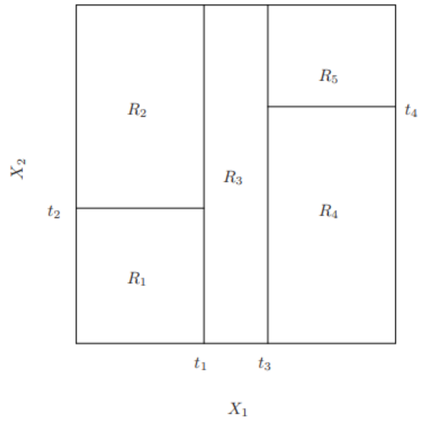
$$\sum_{j=1}^{J} \sum_{i \in R_j} (y_i - \hat{y}_{R_j})^2,$$

   Trees tend to overfit and need to be pruned based on cost-complexity pruning (weakest link pruning)

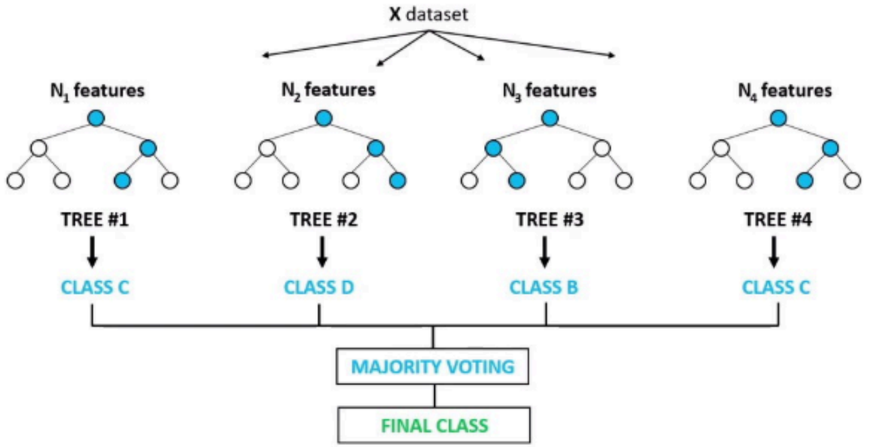
   Evaluation metrics:

- Gini index
- Entropy

$$G = \sum_{k=1}^{K} \hat{p}_{mk}(1 - \hat{p}_{mk}), \qquad D = -\sum_{k=1}^{K} \hat{p}_{mk} \log \hat{p}_{mk}.$$



Random Forest Classifier



PrepVector
UPSCALE YOUR CAREER

Resources

PrepVector
UPSCALE YOUR CAREER

Resources

Introduction to Statistical Learning Book:
https://hastie.su.domains/ISLR2/ISLRv2_website.pdf

Microsoft ML for Beginners course:
https://github.com/microsoft/ML-For-Beginners

Rules of Machine Learning by Google:
https://developers.google.com/machine-learning/guides/rules-of-ml

Google Machine Learning course:
https://developers.google.com/machine-learning/crash-course

Black Box Machine Learning ppt:
https://davidrosenberg.github.io/mlcourse/Archive/2017Fall/Lectures/01.black-box-ML.pdf

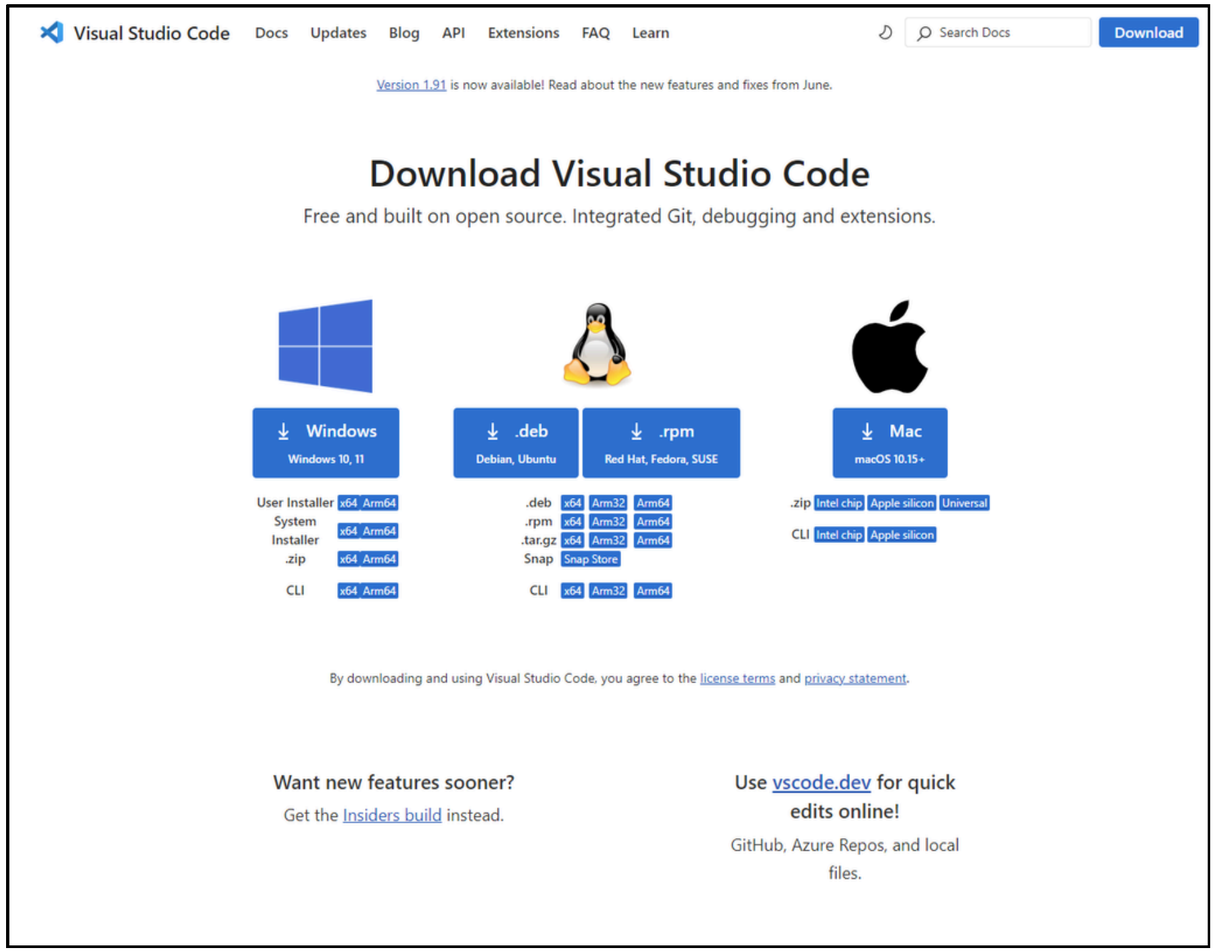
Data Driven Science & Engineering book:
http://databookuw.com/databook.pdf

PrepVector
UPSCALE YOUR CAREER

3. INSTALLATIONS: VSCODE, GIT & GITHUB SETUP

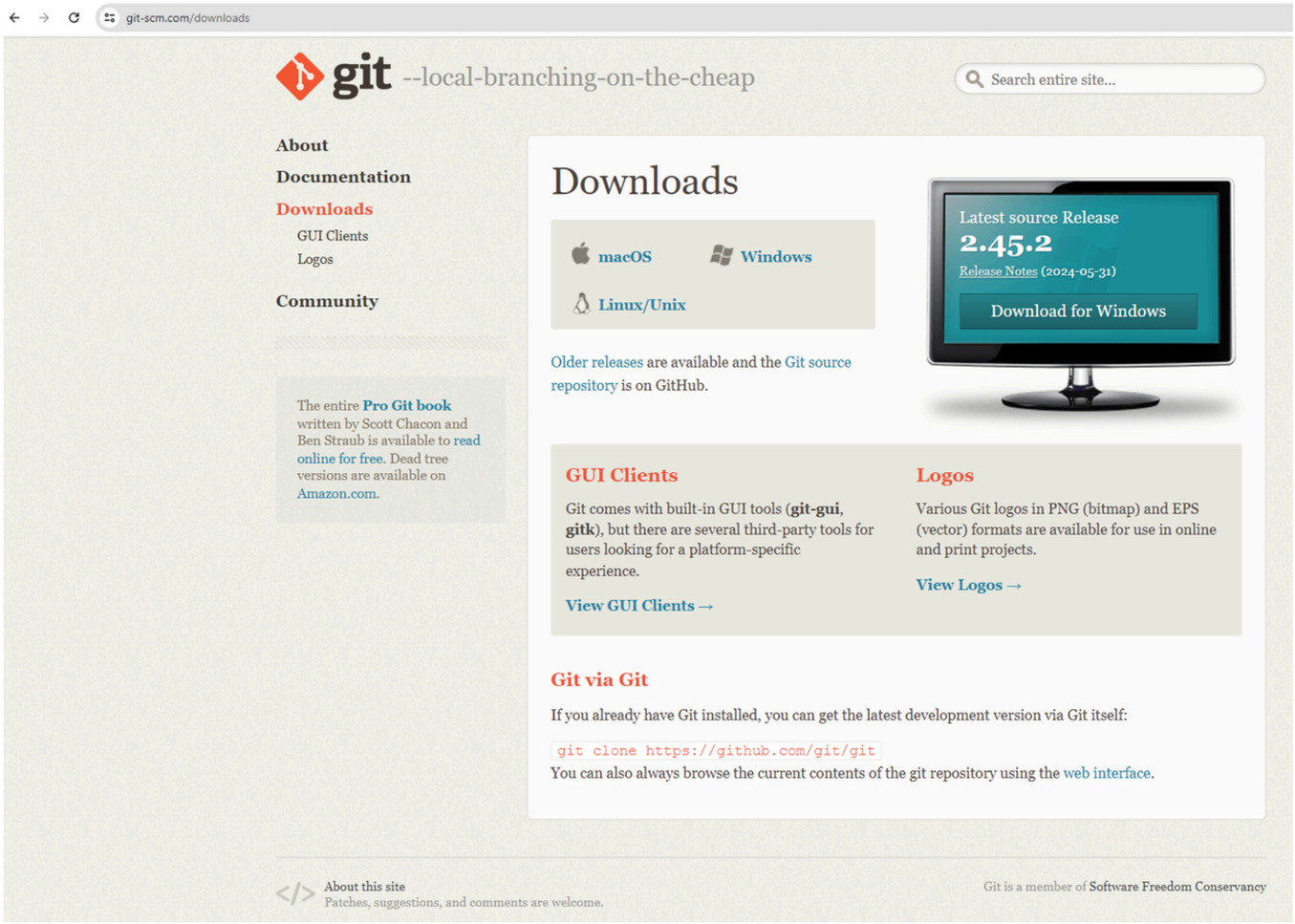
DOWNLOAD VSCODE

Visit https://code.visualstudio.com/download and download the version based on your OS



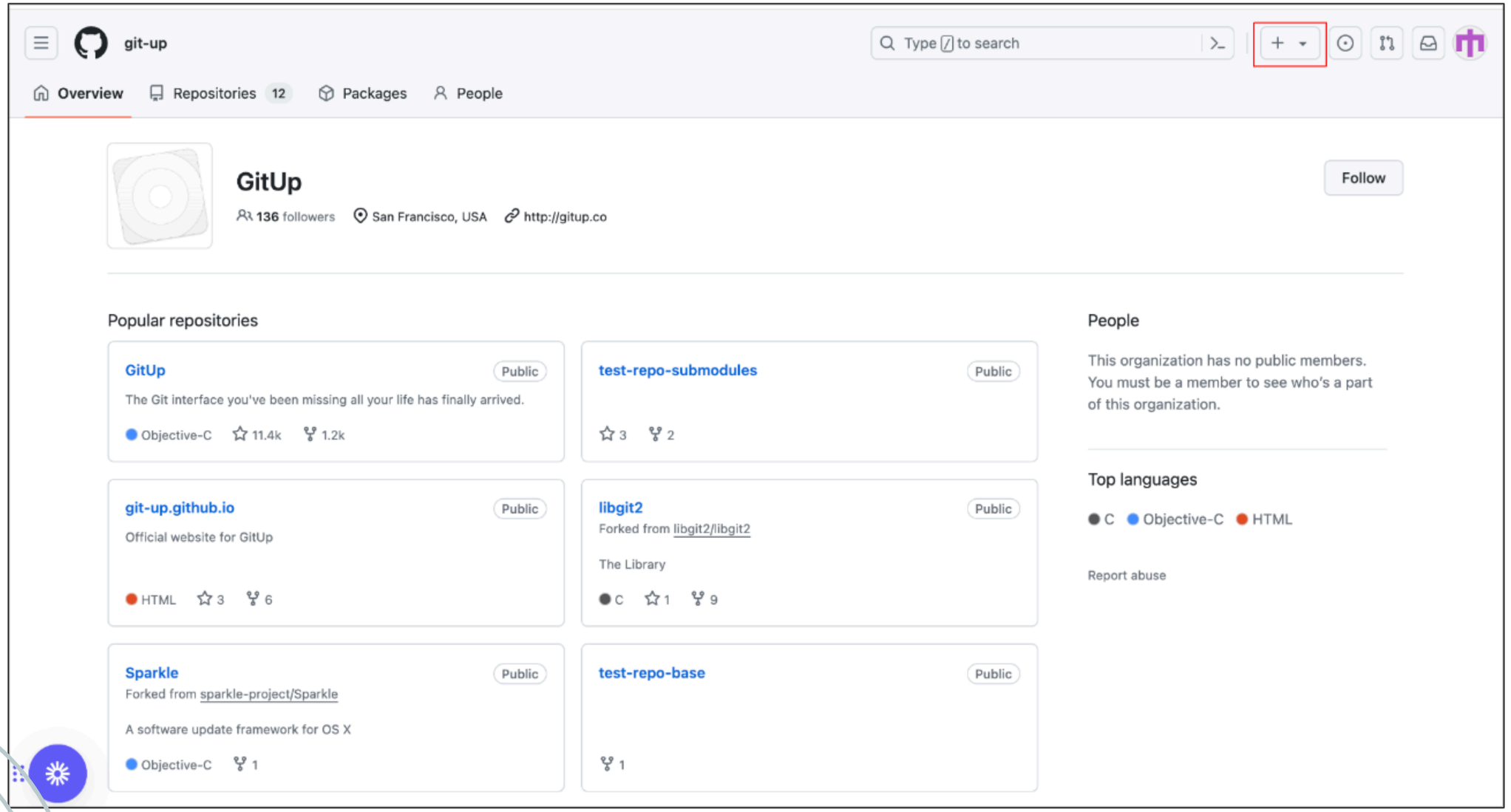
DOWNLOAD GIT

Visit git-scnm.com/downloads and download the version based on your OS



CREATE A REPO

In the top right corner click ‘create repository’.



Create a repository and add files

Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? Import a repository.

Required fields are marked with an asterisk (*).

Owner * sujithra14 /

Repository name * Name your repository

Great repository names are short and memorable. Need inspiration? How about silver-telegram ?

Description (optional)

- **Public** — Anyone on the internet can see this repository. You choose who can commit.
- **Private** — You choose who can see and commit to this repository.

Set your accessibility mode

Initialize this repository with:

- [x] **Add a README file** — Check the box

This is where you can write a long description for your project. Learn more about READMEs.

Add .gitignore

.gitignore template: None

Choose which files not to track from a list of templates. Learn more about ignoring files.

Choose a license

License: None

A license tells others what they can and can't do with your code. Learn more about licenses.

You are creating a public repository in your personal account.

Click here → **Create repository**

Pull requests | Actions | Projects | Wiki | Security | Insights | Settings

DataAnalysis Public — Pin | Unwatch 1

main | 1 Branch | 0 Tags | Go to file t | Add file | Code

- Create new file
- Upload files

sujithra14 Create UnicornCompanyData	bcc9	
Data analysis Practice (1).ipynb	Add files via upload	2 years ago
UnicornCompanyData	Create UnicornCompanyData	last year

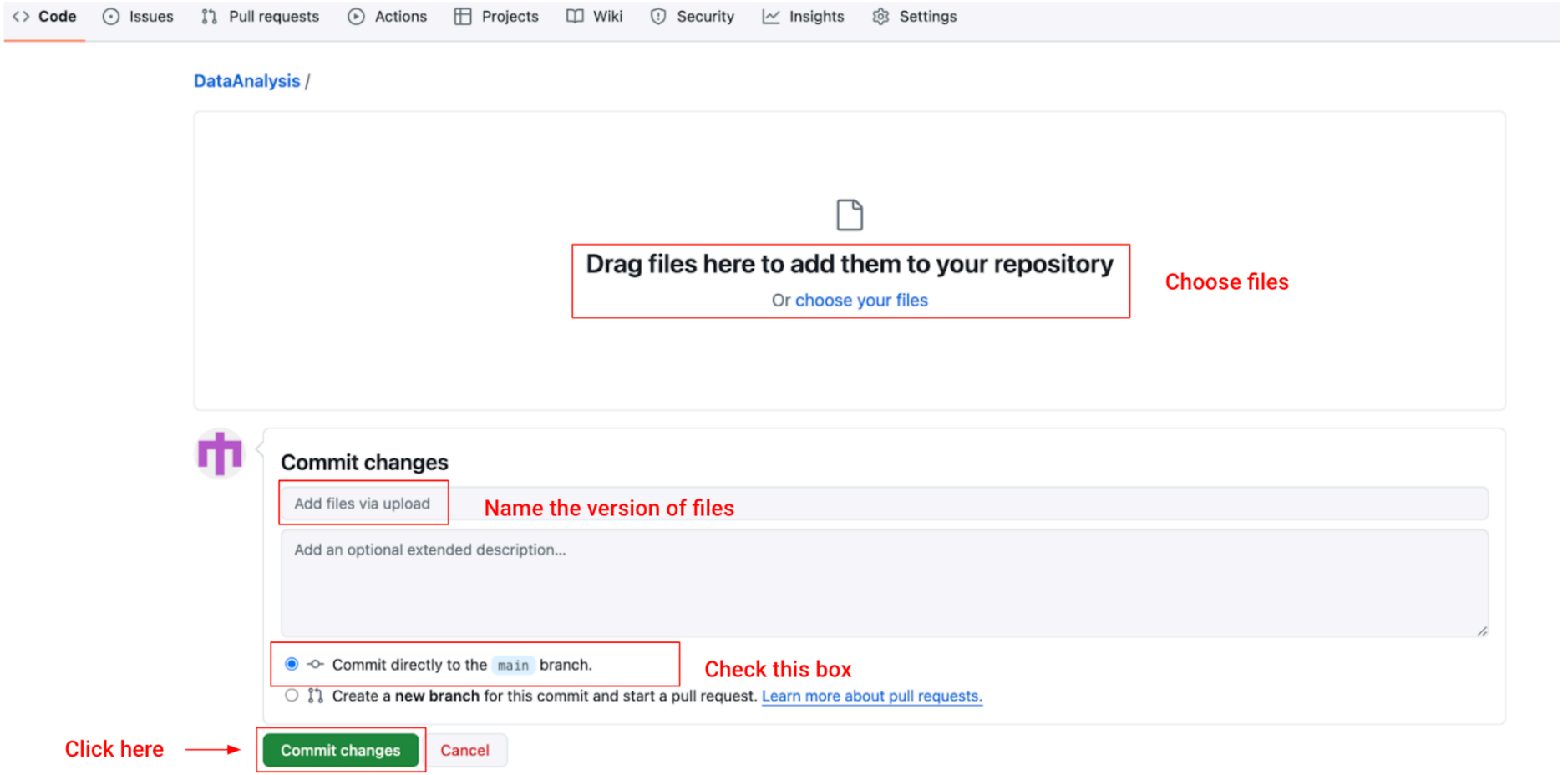
README

Add a README

Help people interested in this repository understand your project by adding a README.

Add a README

Add a name to the version and commit the changes



PrepVector
UPSCALE YOUR CAREER

4. ORGANIZING DS WORKFLOW

DATA SCIENCE WORKFLOW

Data Science projects are often iterative in nature, so defining a structure to your projects is key.

Data Scientists engaged with rapid experimentation, visualization, and data analysis. Without a formal structure in place, returning to past analysis or projects can be challenging.

Following a structured approach to projects will benefit:

- your future self, when you need to go back to your past work
- other team members, who will review your code or work on further iterations of your work
- new people on the team, who will be ramping up on the work your team delivers

DATA SCIENCE WORKFLOW

How to organize projects in Github: sample workflow below.

```
project
    ├──data
    │    ├──external    <- Third party data
    │    ├──interim     <- Transformed intermediate data, not ready for modelling
    │    ├──processed   <- Prepared data, ready for modelling
    │    └──raw         <- Immutable original data
    ├──models           <- Serialized model
    ├──notebooks        <- Jupyter notebooks for exploration, communication and protyping
    └──src              <- Folder containing project source code
        ├──data         <- Folder containing scripts to download/generate data
        ├──features     <- Folder containing scripts to transform data for modelling
        └──model        <- Folder containing scripts to train and predict
```

PrepVector
UPSCALE YOUR CAREER

5. PROJECT 1: PROBLEM WALKTHROUGH

UBER ETA PREDICTION

Background:
Assume Uber is a food delivery company that was launched in 2022. The users can select a restaurant to order any of the following food items: snacks, drinks, meals, buffet. The delivery partner normally uses a bicycle, electric scooter, scooter, or motorcycle to deliver the order.

Goal:
Create an internal tool to estimate the time to deliver the food to the user, based on a set of given inputs. This will be used by other teams for enhancing driver experience, route optimization, capacity planning etc.

Outputs:
Build a machine learning model to predict the time taken to deliver the food. Deploy the application using Streamlit Community Cloud with an easy to use UI, where the time to deliver is calculated based on some user inputs. Ensure that the code is clean, well organized, and document your findings (so your future self and other team members thank you!).

Data description here.

THANK YOU