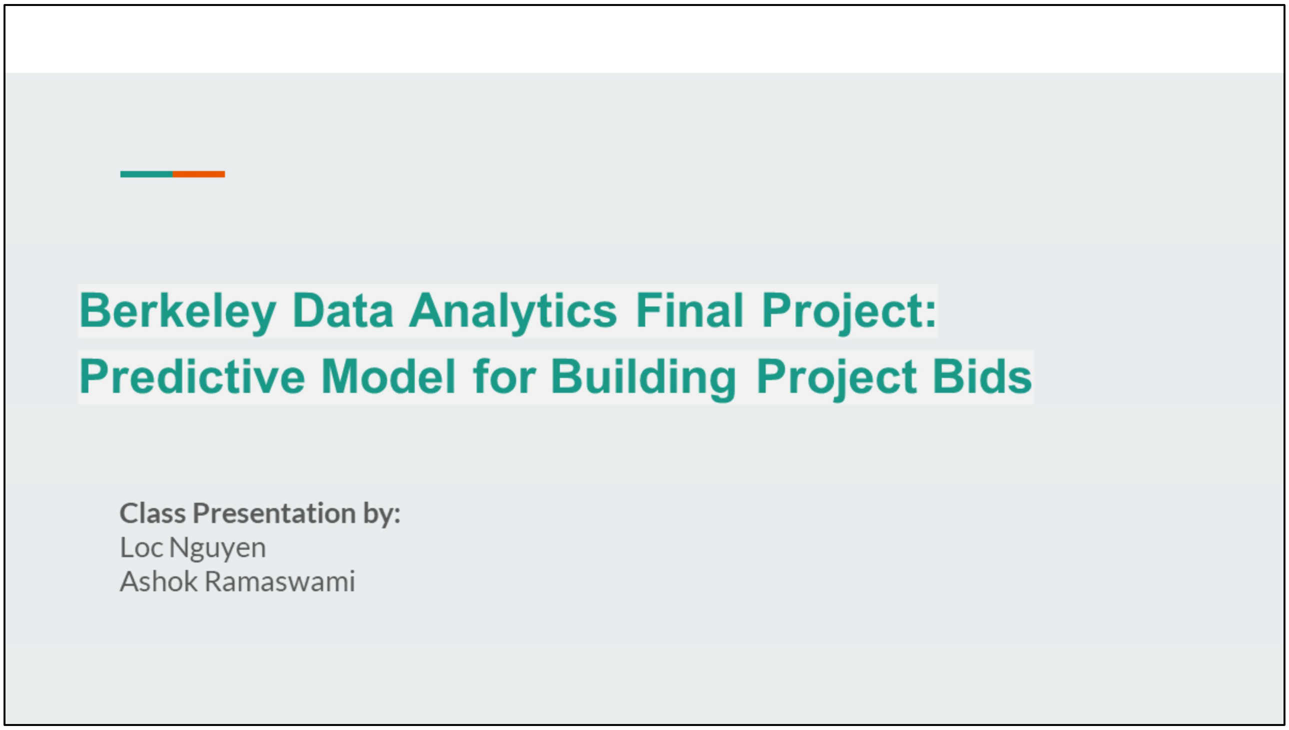
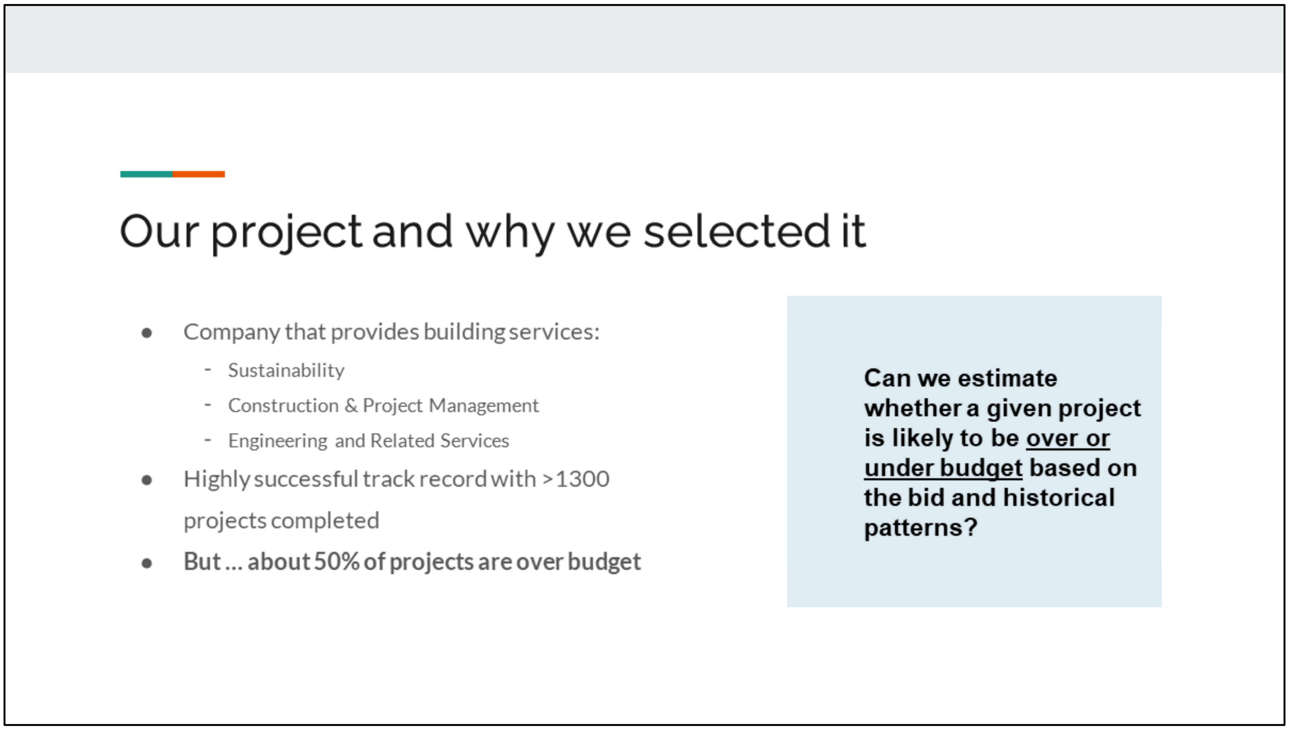


Berkeley Data Analytics Final Project: Predictive Model for Building Project Bids

Class Presentation by:
Loc Nguyen
Ashok Ramaswami



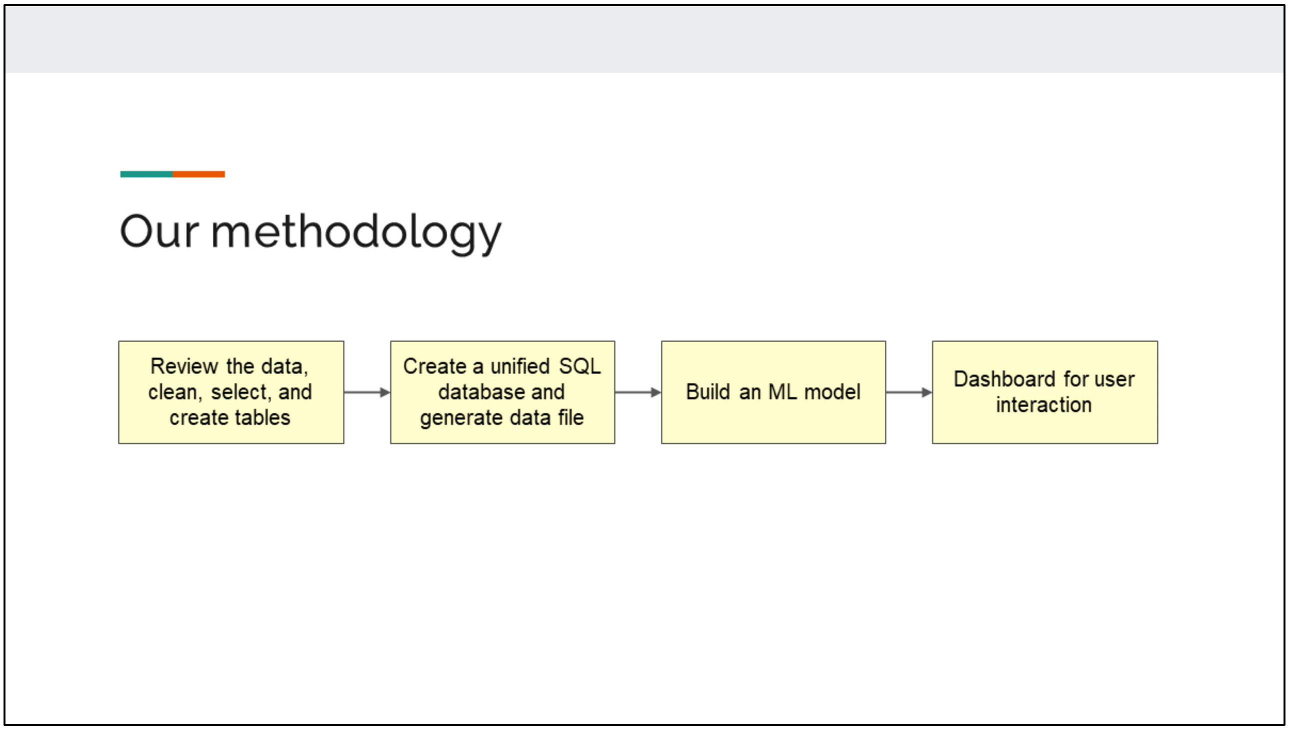
Our project and why we selected it

- Company that provides building services:
  - Sustainability
  - Construction & Project Management
  - Engineering and Related Services
- Highly successful track record with >1300 projects completed
- **But … about 50% of projects are over budget**

Can we estimate whether a given project is likely to be <u>over or under budget</u> based on the bid and historical patterns?

Speaker notes - Ashok:

- We're working with real world data from a company that provides a variety of services for buildings
- The company has a very successful track record and reputation with over 1300 completed projects
- An aspect of the work that the company would like to improve is their bid evaluation and pricing
- This is because about 50% of their projects have been over budget
- The question we'd like to address is whether an ML model can predict if a given project is likely to be over or under budget given the type, scope, and other details of a project
- Gaining some insight on this could help the company bid more accurately and manage better the budget for projects that fit certain patterns
- Our goal: Build a model that predicts the likelihood of a project being under or over budget



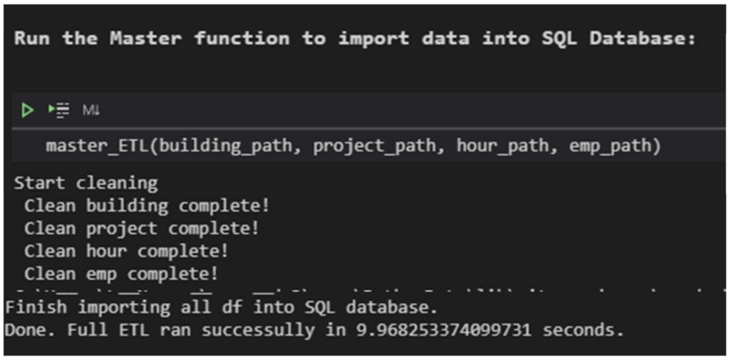
Our methodology

Review the data, clean, select, and create tables → Create a unified SQL database and generate data file → Build an ML model → Dashboard for user interaction

Speaker notes - Ashok:
Our methodology consists of four steps, broadly. We'll expand on this in subsequent slides.

- 1. Clean up data: using python with pandas and numpy libraries. Then use sqlalchemy to load the data into SQL Server.
- 2. Create a database: the ETL returns 4 tables to load on our server, then we use SQL to merge them before extracting the cleaned data.
- 3. Processing the cleaned data, then build and train a ML model.
- 4. Build a deliverable dashboard for the users to apply that model

ETL

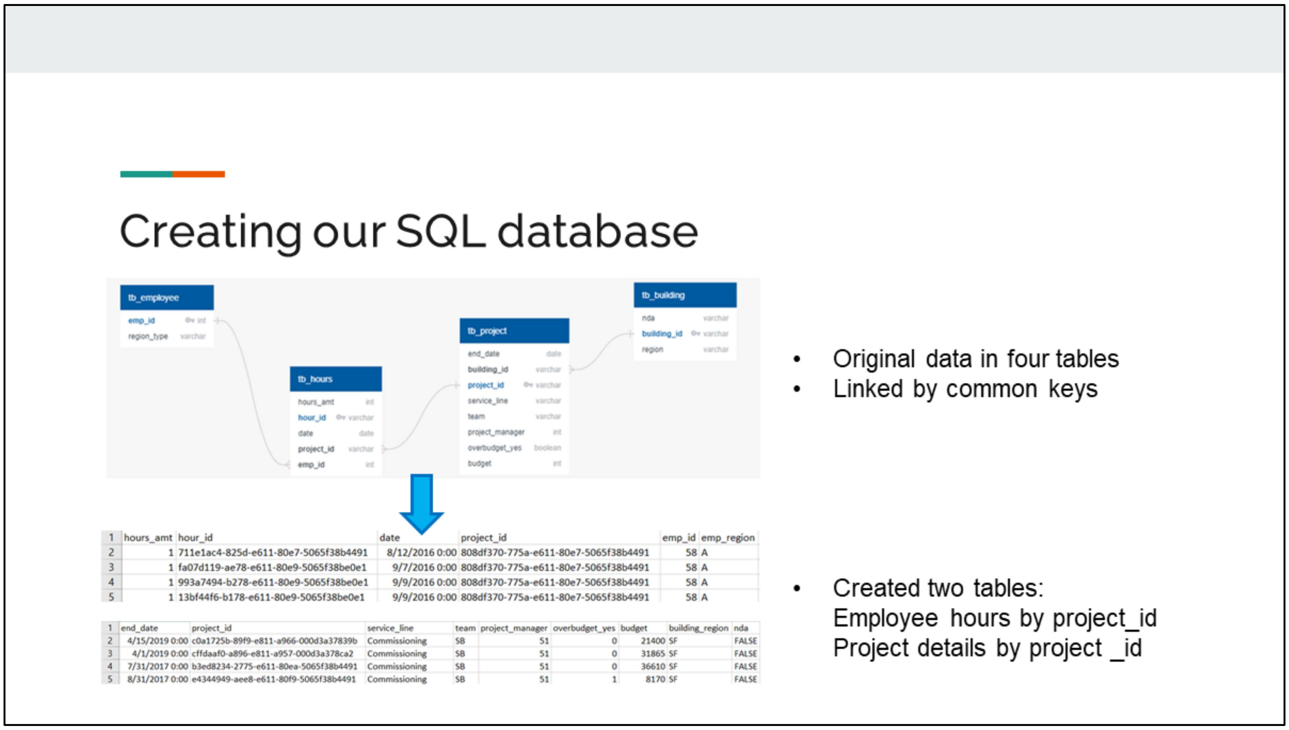


Run the Master function to import data into SQL Database:

```
master_ETL(building_path, project_path, hour_path, emp_path)
```

```
Start cleaning
 Clean building complete!
 Clean project complete!
 Clean hour complete!
 Clean emp complete!
Finish importing all df into SQL database.
Done. Full ETL ran successully in 9.968253374099731 seconds.
```

ETL - Loc:
Step 1 was to load all the raw data, cleaned them up (remove or fill na), kept just the columns that we needed, convert the datatypes, and rename them into something meaningful. Then we built a few functions to have all of them auto-loaded the next time we have a new raw dataset.
The snippet showed it took about 10 seconds to finish the ETL with 4 tables.



Creating our SQL database

tb_employee
- emp_id int
- region_type varchar

tb_hours
- hours_amt int
- hour_id varchar
- date date
- project_id varchar
- emp_id int

tb_project
- end_date date
- building_id varchar
- project_id varchar
- service_line varchar
- team varchar
- project_manager int
- overbudget_yes boolean
- budget int

tb_building
- nda varchar
- building_id varchar
- region varchar

1	hours_amt	hour_id	date	project_id	emp_id	emp_region
2	1	711e1ac4-825d-e611-80e7-5065f38b4491	8/12/2016 0:00	808df370-775a-e611-80e7-5065f38b4491	58	A
3	1	fa07d119-ae78-e611-80e9-5065f38be0e1	9/7/2016 0:00	808df370-775a-e611-80e7-5065f38b4491	58	A
4	1	993a7494-b278-e611-80e9-5065f38be0e1	9/9/2016 0:00	808df370-775a-e611-80e7-5065f38b4491	58	A
5	1	13bf44f6-b178-e611-80e9-5065f38be0e1	9/9/2016 0:00	808df370-775a-e611-80e7-5065f38b4491	58	A

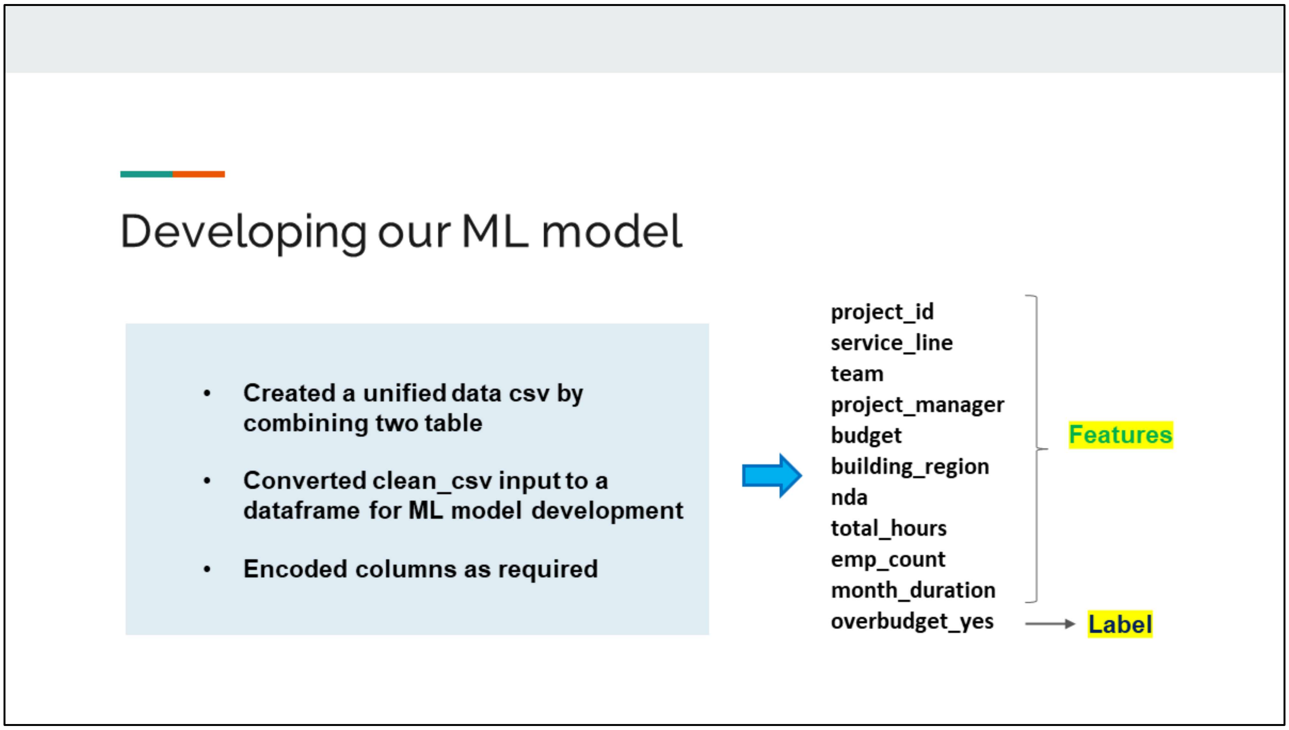
1	end_date	project_id	service_line	team	project_manager	overbudget_yes	budget	building_region	nda
2	4/15/2019 0:00	c0a1725b-89f9-e811-a966-000d3a37839b	Commissioning	SB	51	0	21400	SF	FALSE
3	4/1/2019 0:00	cffdaaf0-a896-e811-a957-000d3a378ca2	Commissioning	SB	51	0	31865	SF	FALSE
4	7/31/2017 0:00	b3ed8234-2775-e611-80ea-5065f38b4491	Commissioning	SB	51	0	36610	SF	FALSE
5	8/31/2017 0:00	e4344949-aee8-e611-80f9-5065f38b4491	Commissioning	SB	51	1	8170	SF	FALSE

- Original data in four tables
- Linked by common keys

- Created two tables:
  Employee hours by project_id
  Project details by project _id

Speaker notes - Loc:

- After the ETL process, we move on to the 2nd step to work on our server. Our original data is in 4 tables: Building, project, hours, and employees.
- We use SQL to:
  - Merged the project table with the building table in order to extract the building information for each project,
  - Merged the hours table with employees table to get the employee region for each hour entry.
  - Completed data exploration to remove features that may overfit the model. For e.g: after merging the hours table, we found out that almost everyone was in the same region. So the employee region will not be a good feature anymore. Thus, we removed it in our cleaned df.



Developing our ML model

- Created a unified data csv by combining two table
- Converted clean_csv input to a dataframe for ML model development
- Encoded columns as required

project_id
service_line
team
project_manager
budget
building_region
nda
total_hours
emp_count
month_duration
overbudget_yes

Features

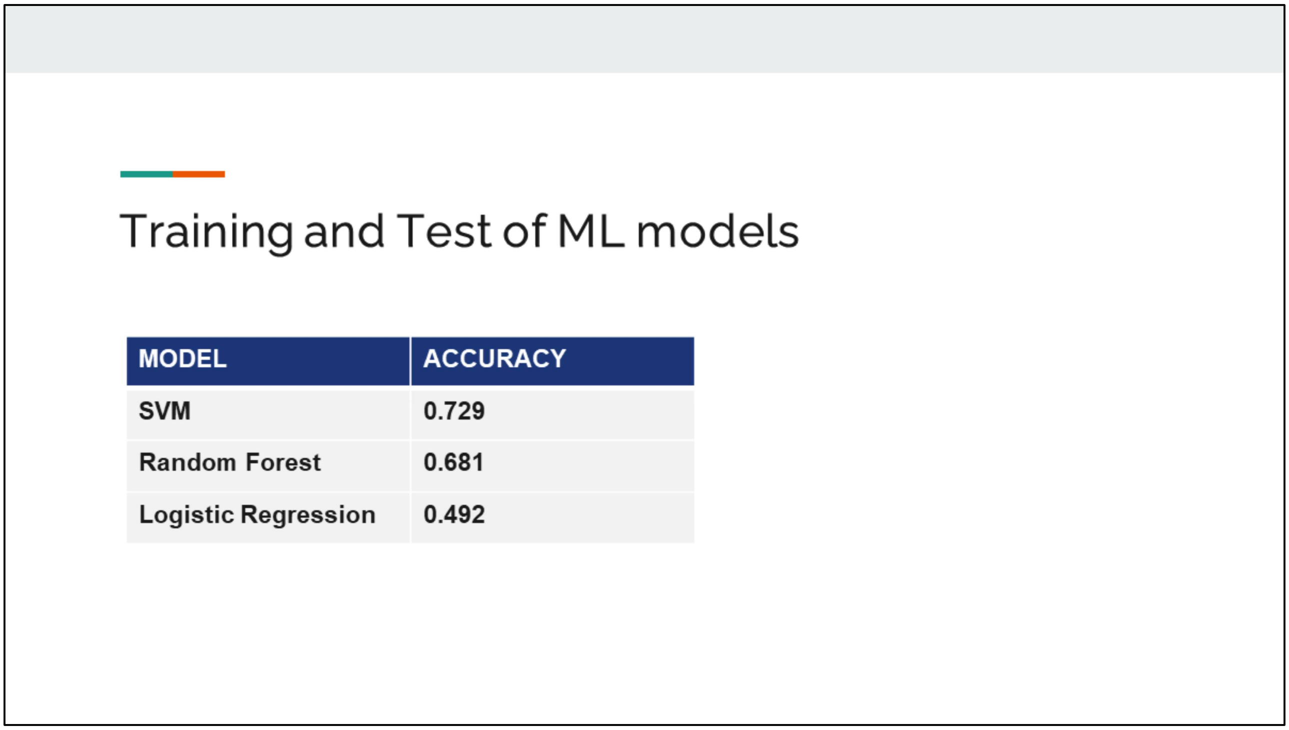
Label

Speaker notes - Ashok:

- Our data was fairly clean but still needed some preprocessing. This included:

  Summing the number of hours from the hour table to the project
  Calculating the project month duration using the first date/last date of the hours table and the project end date
  Counting the number of employees working on each project.

  Following this cleanup process, we prepared the dataset for one-hot-coding, scaling, and generating a model. We have 10 features for the ML model.

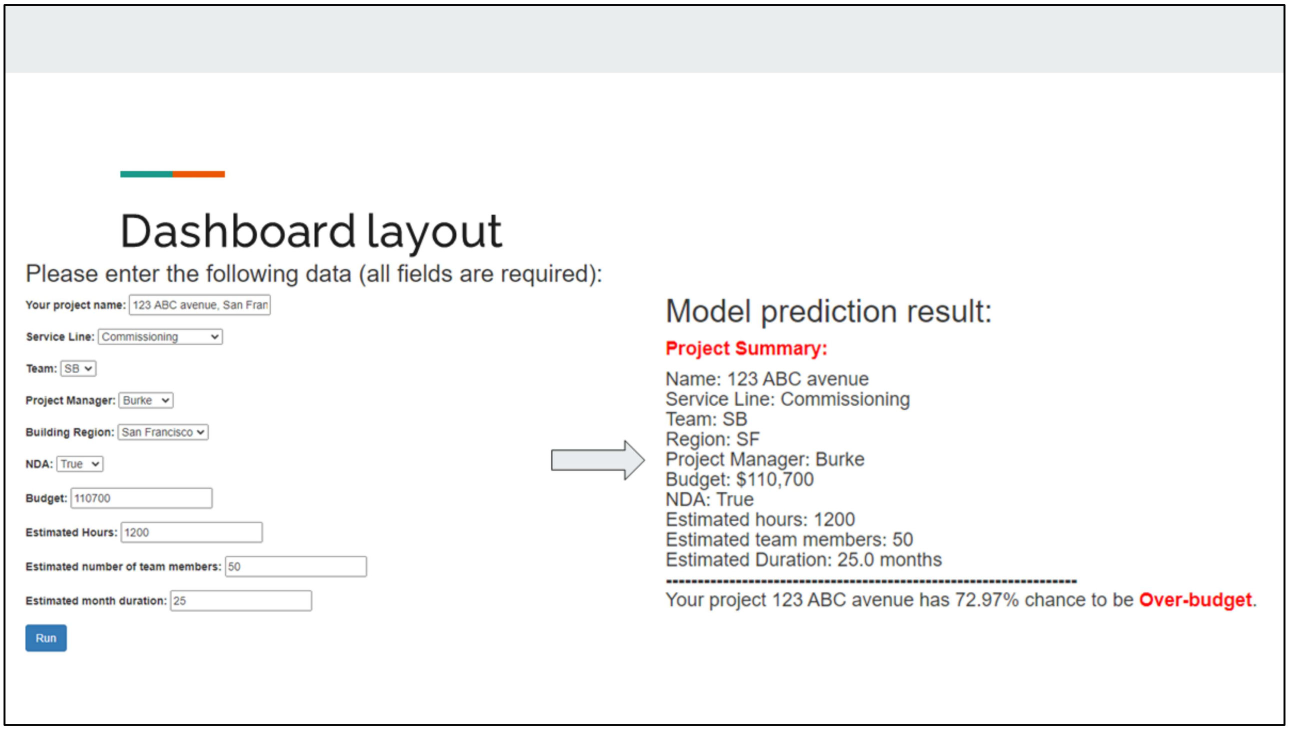


Training and Test of ML models

MODEL	ACCURACY
SVM	0.729
Random Forest	0.681
Logistic Regression	0.492

Speaker notes - Ashok:
We created a dataset based on the feature engineering that we described in earlier slides
- Specifically we created ML models with three approaches:
    - SVM
    - Random Forest
    - Logistic Regression
- We applied the standard form for separating training and test data
- We also created a version of the ML model using binning for one of the features
- We then reviewed the model accuracy for each approach

As an additional step, we ran a cross validation to evaluate model accuracy under different test samples



Dashboard layout

Please enter the following data (all fields are required):

Your project name: 123 ABC avenue, San Fran

Service Line: Commissioning

Team: SB

Project Manager: Burke

Building Region: San Francisco

NDA: True

Budget: 110700

Estimated Hours: 1200

Estimated number of team members: 50

Estimated month duration: 25

Model prediction result:

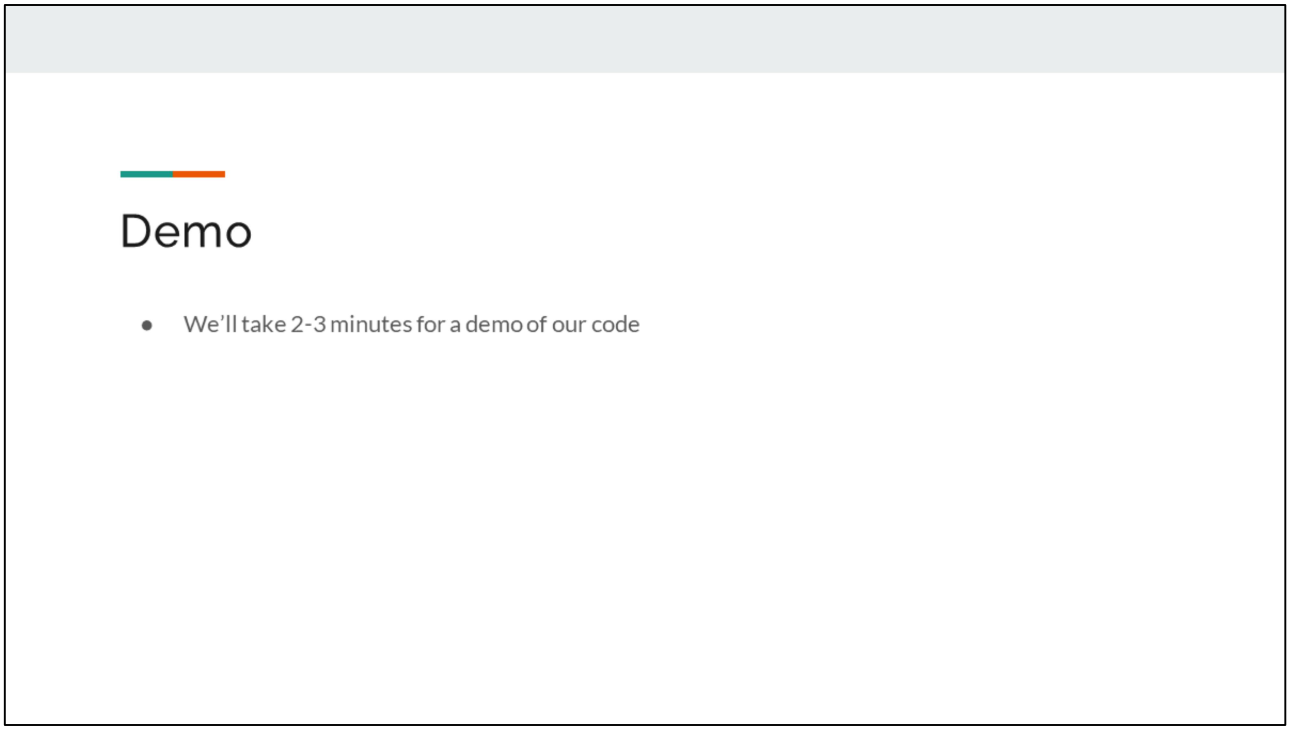
Project Summary:

Name: 123 ABC avenue
Service Line: Commissioning
Team: SB
Region: SF
Project Manager: Burke
Budget: $110,700
NDA: True
Estimated hours: 1200
Estimated team members: 50
Estimated Duration: 25.0 months

Your project 123 ABC avenue has 72.97% chance to be **Over-budget**.

Speaker notes - Loc:

- Our goal is to create a simple dashboard that would take some user input parameters (such as project budget, duration, number of employees, estimated project manager, estimated hours, etc.) and predict the target
- To implement the dashboard we:
  Download the ML model using pickle library
  Developed a Flask app with python to take and process user input
  Developed UI code to display user input fields and display a prediction

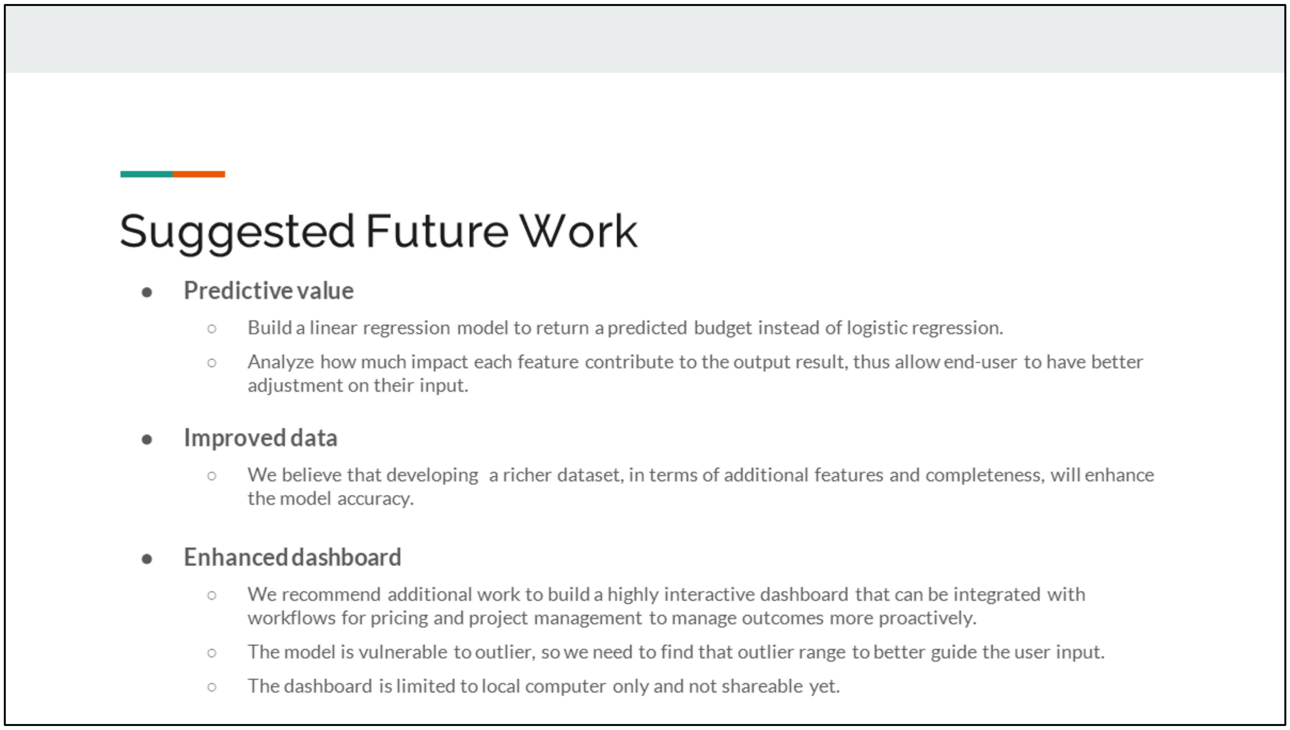


Demo

- We'll take 2-3 minutes for a demo of our code

Loc:

- Over-budget: 'Commissioning', 'SB', Burke, 110700, 'SF', True, 1200.0, 50, 25
- Under-budget: 'WELL', 'SB', Ashok, 56221, 'SF', True, 200, 15, 6



Suggested Future Work

- **Predictive value**
  - Build a linear regression model to return a predicted budget instead of logistic regression.
  - Analyze how much impact each feature contribute to the output result, thus allow end-user to have better adjustment on their input.
- **Improved data**
  - We believe that developing a richer dataset, in terms of additional features and completeness, will enhance the model accuracy.
- **Enhanced dashboard**
  - We recommend additional work to build a highly interactive dashboard that can be integrated with workflows for pricing and project management to manage outcomes more proactively.
  - The model is vulnerable to outlier, so we need to find that outlier range to better guide the user input.
  - The dashboard is limited to local computer only and not shareable yet.

Loc:

1. Predict how much should we bid for a particular project, similar to a pricing tool.
   Consider adding/reducing other features to further increase the accuracy, but careful not to overfit the model.
2. The more data, the better our model.
   Make sure to collect only clean data.
3. - The model is vulnerable to outlier (such as budget as low as $1, or 1000+ employees, but still return “Under-budget”).
   - The end-user is usually non-tech people, so this dashboard has many rooms to improve. Such as: include some pie chart/report, save/export multiple predictions/scenarios, provide some suggestion on what should the user do to have the expected outcome (including limiting the range of input to avoid outliers).
   - The flask app is not shareable yet.