CS 422 Project 1 Time Series Forecasting & Benchmarks Team 1 "zeakz"

Aiden Duval, Zane Globus-O'Harra, Erin Stone, Kareem Taha, Zachary Weisenbloom

Team Management and Meetings

Product Owner – Aiden

Frontend - Erin

Backend - Zane, Zach and Kareem

Meetings (online and in person)

Mondays 6-8pm

Thursdays 4-6pm

What we considered:

What we did:

Frontend:

Flask, React, Express



Backend:

Flask, MongoDB, mySQL



Frontend:

HTML, JS, CSS Integration into Django



Backend:

SQLite through Django



N						pr-23			10-Apr-23					17-Apr-23								Apr-2						May-						
	4.								T W																				F S			Т		
Task	assigned to	start	end	5	6	7 8	9	10	11 12	13	14 1	15 16	17	18	19 2	20 2	21 2	2 23	24	25	26	27	28	29 30	1	2	3	4	5	6	7 8	3 9		
Backend & functionality																																		
decide on framework	Team	4/5/2023	4/13/2023																															
set up a DB	Zane & Zach & kareem		4/20/2023																															
download data for testing	Zane & Zach	4/20/2023	5/1/2023																															
functionality to save user account information	Zane & Zach	4/13/2023	4/27/2023																															
functionality for login screen	Zane & Zach & Team	4/13/2023	4/27/2023																															
Store/download .csv/database files	Zane & Zach & Team		5/1/2023																															
functionality to compare test/solution data accuracy			5/1/2023																															
create method for contributor to upload TS data & fo	Zane & Zach & Team	4/20/2023	5/1/2023																							L								
create method for participant to download forecasting	Zane & Zach & Team	4/20/2023	5/1/2023																															
create method for participant to upload solution data	Zane & Zach & Team	4/20/2023	5/1/2023																															
TACAL BY TO BY ON THE CO.																																		
Frontend/UI design																																		
create UI design	Erin		4/20/2023																															
Create login page	Aiden & Erin & Team		4/27/2023																															
create contributor page	Aiden & Erin & Team		4/27/2023																															
create participant page	Aiden & Erin & Team		4/27/2023																															
present statistical comparisons (and graphs?) to parti	Aiden & Erin & Team	4/20/2023	5/1/2023																															
create page for contributor to upload TS data & forec	Aiden & Erin & Team	4/20/2023	5/1/2023																															
create page for participant to download forecasting t	Aiden & Erin & Team	4/20/2023	5/1/2023																															
create page for participant to upload solution data to	Aiden & Erin & Team	4/20/2023	5/1/2023																															
create admin page	Aiden & Erin & Team	4/20/2023	5/1/2023																															
Desgin and Maintenance																																		
SRS Document	Zane & Team	4/5/2023	4/24/2023																															
SDS Document	Team	4/5/2023	4/24/2023																															
design SW architecture	Team	4/5/2023	4/24/2023																															
N. C.							$\overline{}$																											
3																																		
- 21																																		
Organization & Outputs																																		
distribute tasks	Team	4/5/2022	4/17/2023																							1								
set up git repository	Zane		4/5/2023												-							-				1								
set up git repository set up trello	Aiden		4/5/2023																															
							-															-				1								
set up google drive	Aiden		4/5/2023				-									-									-	-								
set up discord	Erin		4/5/2023				-									4					-													
Team Meetings	Team		rring																															
Progress Meetings	Team		rring																															
Project presentations			o 2023-05-05																															
project due	Team	3-05-05 to 2	023-05-09 ??																		- 1							1						

Organization

Our initial approach – Good decisions

Using SQLite for our database



- Relational database scheme
- Converting uploaded sets into downloadable zip files
- Using a Python-based web framework



Our initial approach – Flawed decisions

Frontend vs. Backend

Consequences:

- Lack of clarity about what the other end has accomplished
- Lack of clarity from backend about how frontend works (HTML, JS, etc)
- Lack of clarity from frontend about how backend works (Django models, views)
- Inability to integrate frontend with backend due to the limitations/mechanics of the Django framework and JavaScript

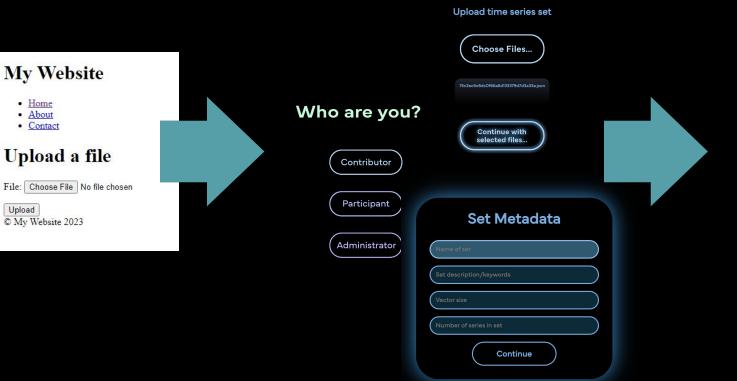
Our solution to that mistake

Frontend vs. Backend

 We ended up having to redo almost all of the frontend development to integrate it with the backend.

 Parts of the frontend implementation had to be scrapped in order to preserve the backend functionality of uploading and using files with the database.

Evolution of Design



Forecasting Task



Upload time series sets





Accuracy Metrics

Accuracy Score % correct

R-correlationRelationship strength

MAE (Mean Absolute Error)

Avg difference

MAPE (Mean Absolute %Error)

Avg % difference

SMAPE (Symmetric MAPE) Avg % difference (from symmetric μ)

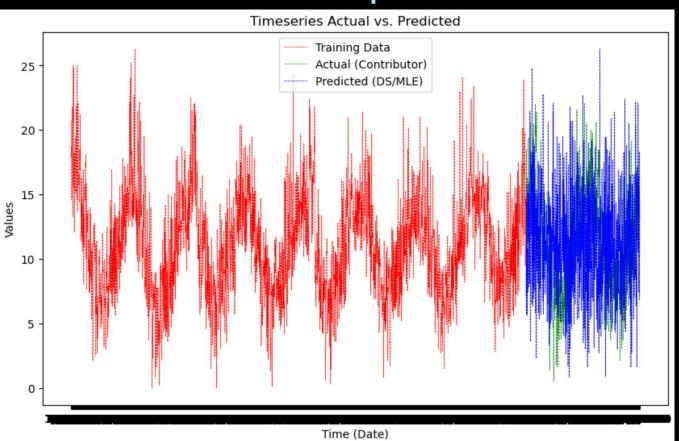
MSE (Mean Squared Error)

Avg (difference²)

RMSE (Root MSE)

√(Avg (difference²))

Model Comparison



Team Experience

- Gained a lot of knowledge about Django/software development
- Difficulty Organizing
 - How do we break down large tasks and split them among the team?
 - What responsibilities will each team member have time to take on?
 - How can we apply everyone's unique skills to the task at hand?
 - What to do next?
- Unforeseen technical problems
 - Linking HTML to Django models to database
 - Issues with version control and GitHub (merging)
- Time management
 - Linking frontend HTML to Django models to database

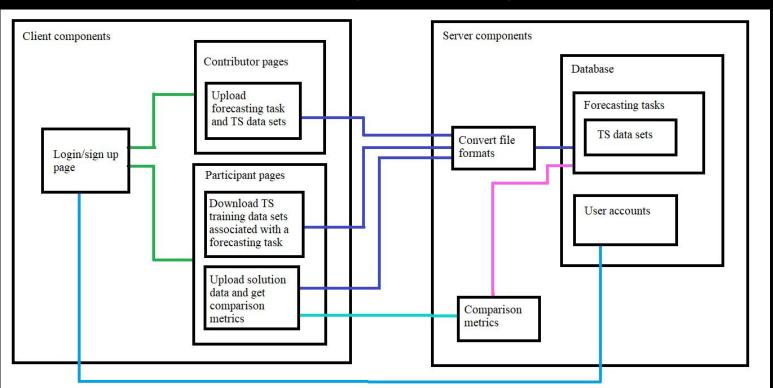
Questions We Asked Our Client

How will time series data be displayed and compared to our models?

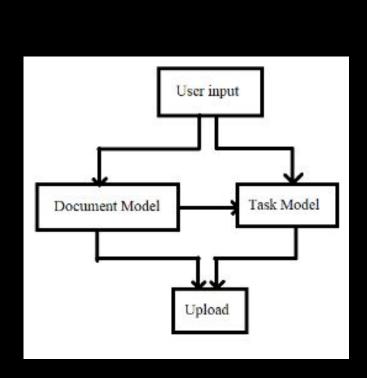
What does each user need to have in terms of functionality and data access?

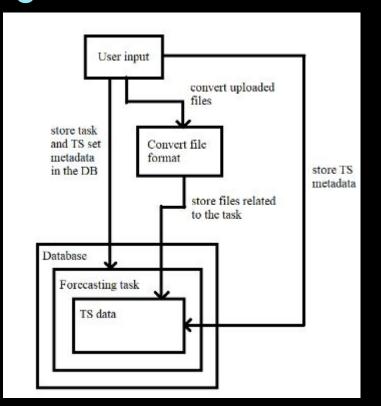
How should data be forecasting internally based on the situation of files uploaded/downloaded?

Design & Integration

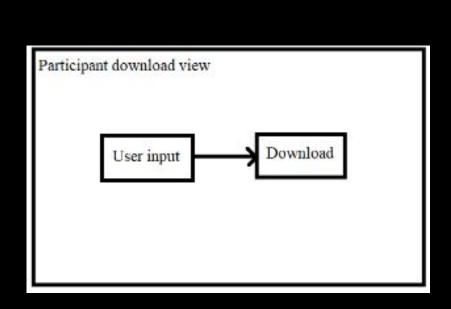


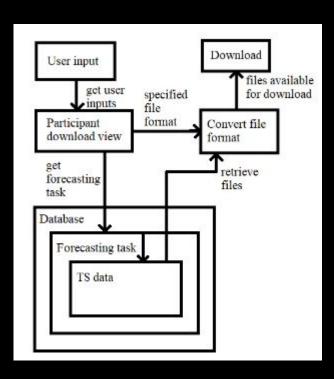
Contributor Page



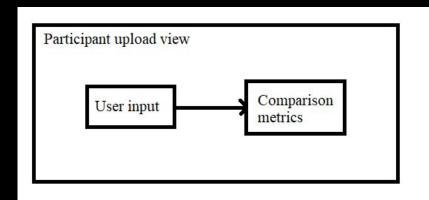


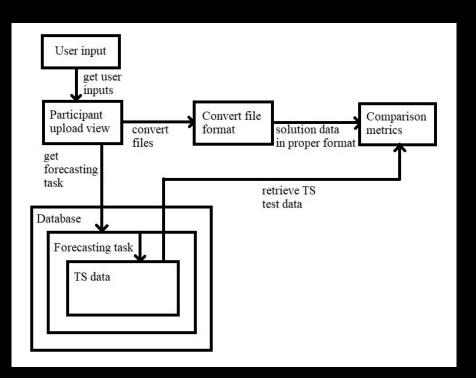
Participant download Page





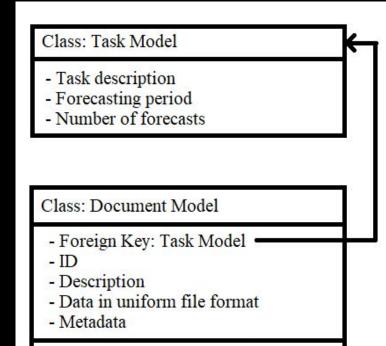
Participant upload page





Class: Account Model - User name - Email - Password Class: Task Model - Task description - Forecasting period - Number of forecasts Class: Document Model - Foreign Key: Task Model • - ID - Description - Data in uniform file format - Metadata - Convert file format module methods

Relational database scheme



- Convert file format module methods

Testing

- Unittests
 - Backend
 - Accuracy metrics
 - Set, Timeseries, Data objects
- Jupyter
 - o I/O
 - Database conversion (JSON -> Pandas)
 - Graph functionality (matplotlib -> Fronter
- User testing
 - Dreamweaver real-time web design

Testing everything is difficult

```
Test 1: Accuracy-Score
# of predictions correct
Accuracy Score: 0.0085 proportion of predictions were correct
Accuracy Score: 0.8493% of predictions were correct
_____
Test 2: Correlation
How similar 2 variables are
r-value: 0.0068
Test 3: MAE (Mean-Absolute Error)
Average prediction error
Mean-Absolute Error (MAE): 4.6390
Test 4: MAPE (Mean-Absolute Percentage Error)
Average prediction % \error
Mean-Absolute Percentage Error (MAPE): 30476413916726.9805
Test 5: SMAPE (Symmetric Mean-Absolute Percentage Error)
Average prediction % \error
Symmetric Mean-Absolute Percentage Error (MAPE): 4.5801
Test 6: MSE (Mean-Squared Error)
Average squared prediction error
Mean-Squared Error (MSE): 32.5554
......
Test 7: RMSE (Root Mean-Squared Error)
Average root squared prediction error
Root Mean-Squared Error (RMSE): 4.5905
Ran 7 tests in 0.179s
OK
[Done] exited with code=0 in 15.371 seconds
```