AC Sales Forecasting using Meteorological factors

Author
ARYAN DAGA, ARYAN GOSAIN, ATHARV RAGHUWANSHI

Date
16 MAY 2025

Problem Statement

- AC retailers/manufacturers struggle to predict daily demand accurately
- Current methods have isolated datasets and use region-level aggregates and do not replicate real world sales.
- Current models fail to capture the non-linear, region-specific nature of weather-influenced demand.
- Fail to capture correlation of meteorological effects on purchasing

LITERATURE REVIEW

There were no publicly available research papers which showcased AC Sales and the papers forecasting sales of general items used a limited subset of weather parameters under isolated conditions

- **Regression & tree-based ML with lagged weather features** - used isolated weather only dataset, needs to be retrained as patterns change
- **Stacking ensemble (RF, XGBoost, GBDT)** - needs extensive tuning, no interpretability
- **Regression + Monte Carlo simulation for forecast scenarios** - isolated dataset, requires probabilistic forecasting
- **XGBoost vs ARIMA & exponential smoothing** - blackbox nature hence no explainability, needs more datapoints

References are at the end

LITERATURE REVIEW

	Paper	Models Used	Pros	Cons
[1]	Chan & Wahab (2024)	Regression & Tree-based ML with Lagged Weather Features	• Improved accuracy (+47% for products, +56% for categories) • Captures short-term & lagged weather effects • Feature importance improves interpretability	• Less effective for non-weather-driven products • Requires accurate weather forecasts • Retraining needed as patterns change
[2]	Lv et al. (2023)	Stacking Ensemble (RF, XGBoost, GBDT)	• High gains for seasonal items (41–86% MSE drop) • Outperforms single models • Captures non-linear weather-sales relationships	• Ineffective for non-seasonal products • Stack is harder to interpret • Higher tuning/training complexity
[3]	Verstraete et al. (2019)	Regression + Monte Carlo Simulation for Forecast Scenarios	• Works for short- & long-term planning • Handles weather uncertainty via simulation • Automatically picks best short-term model	• Two-stage system adds complexity • Computationally expensive • Needs probabilistic forecasts
[4]	Haselbeck et al. (2022)	XGBoost vs ARIMA & Exp. Smoothing	• ML beat classical methods in 14/15 cases • Captures seasonality, regime shifts (e.g., COVID) • Weather/holiday boosts accuracy	• Tree models are black-boxes • Needs more data and tuning • Relies on external inputs like weather and events
[5]	Badorf & Hoberg (2020)	Random Coefficient Regression (Hierarchical Linear Model)	• Captures store-specific sensitivity • Models non-linear weather effects • Accurate for 1–7 day forecasts	• Complexity scales with number of stores • Poor performance on longer horizons • Assumes stability of past behavior
[6]	Liu & Ichise (2017)	Deep Learning (LSTM + Autoencoder)	• Excellent for weather-driven food sales • 19.3% better than ML baselines • Captures sequential and nonlinear effects	• Hard to interpret (black box) • Requires large data & compute • Not always better for non-weather products
[7]	Retail Case Studies	Commercial ML platforms + weather integration	• Real-time localization • Boosts revenue & reduces waste • Useful for inventory, promotion, staffing	• Dependent on weather accuracy • Not useful for general/staple goods • External services add cost and reduce control

References are at the end

GAP

- General forecasting models are unexplainable, i.e a "blackbox"
- Models trained on historical patterns can’t quickly recalibrate to sudden heatwaves or novel climate shifts.
- All current models have isolated very specifically crafted datasets and is not real world data.

Project Objectives

- Preprocess the data effectively and understand how usable it is.
- Build models to forecast AC sales for **9 different Indian cities**.
- Ensure **explainability** in modeling approaches.
- Understand sales patterns & data limitations.

so we got a **real world dataset**

FOR SALES OF AIR CONDITIONERS AND WEATHER PARAMETERS

DATASETS

```
columns_to_drop  = ['Capacity Rating', 'Energy Rating', 'S
df.drop(columns=columns_to_drop, inplace=True)
df.head()
```

	Invoice Date	Invoice Quantity	Sales Zone	Sales Location
0	4/5/2024	1	East	PTA
1	4/5/2024	150	South	HBI
2	4/5/2024	90	South	KCI
3	4/5/2024	84	South	KCI
4	4/6/2024	120	North	JPR

- Bhopal.csv
- Durgapur.csv
- Ghaziabad.csv
- Gurgaon.csv
- Hubli.csv
- Jaipur.csv
- Jammu.csv
- Kochi.csv
- Lucknow.csv
- Madurai.csv
- New Delhi.csv
- Patna.csv

```
# Dir
weath

# Pat
csv_f

dataf
for f
    #
    c
    d
    d
    d
```

Sales Invoice data

Sales from 2021-2024, as and when it occurred for 12 cities but then 3 were unusable due to empty cells, so we ended up with 9 cities

Weather Data

Kaggle Dataset with 20 GB of Weather Data for 800+ Indian cities dated 2020-2024

Combined dataset mapped using city tags

Final Dataset had 29 features including temperature, humidity, location, precipitation etc.

```
city_mapping = {
    "BHP": "Bhopal", "DGP": "Durgapur", "DLI": "New Delhi", "GRN": "Gurgaon",
    "GZB": "Ghaziabad", "HBI": "Hubli", "JMU": "Jammu", "JPR": "Jaipur",
    "KCI": "Kochi", "LKW": "Lucknow", "MDI": "Madurai", "PNE": "Pune",
    "PTA": "Patna"
}
```

Invoice Quantity

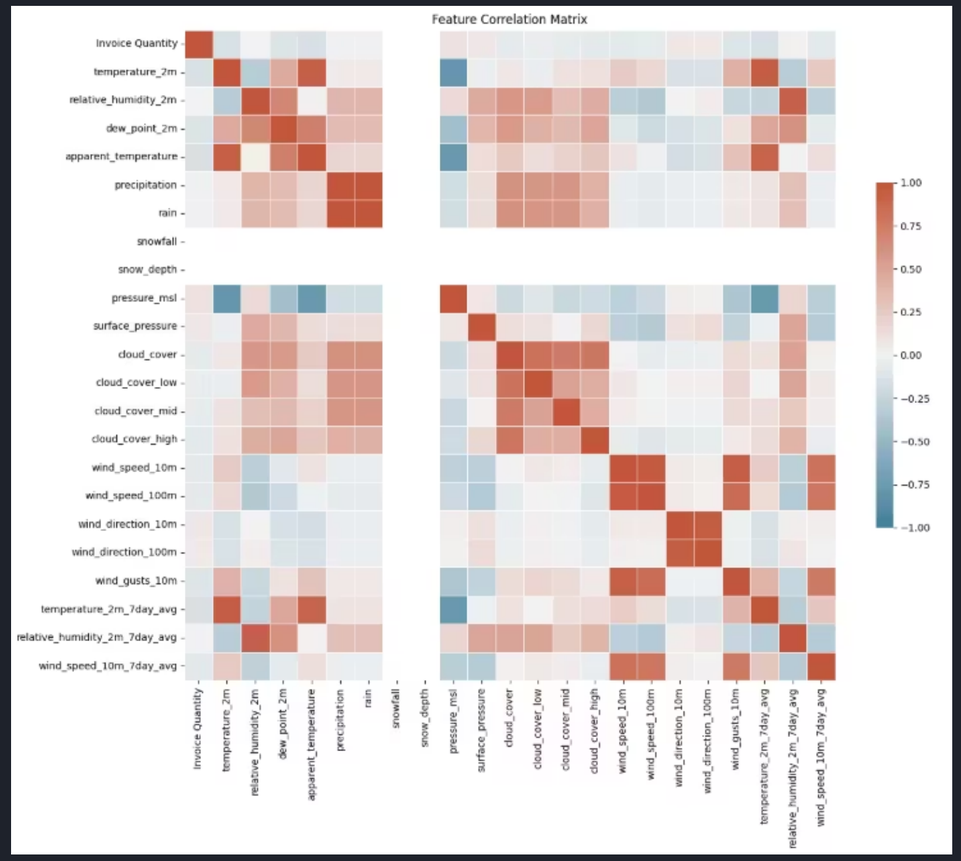
WAS OUR TARGET VARIABLE

Preprocessing

```
PC1,PC2,PC3,PC4,PC5,PC6,PC7,PC8,PC9,PC10,City
-2.1439409543229777,3.425357846948046,-1.5131893660010594,2.2889127718253475,-2.4715656217027027,1.921854587905054,0.3852053629210869,0.43354799257961846,0.2588880145020217,1
0.0633672008338589,0.806981301402826,-1.2835242919290992,0.9539971845392415,-0.0885687100172247,-0.37819498409862273,0.3886702618048805,1.1139032176037316,-0.42558504155648
-3.0141260247910364,1.563752258013005,-0.662057040529376,2.2597393059054327,-2.2777877496865013,3.5711268079075817,0.6298279215950704,0.6244688237480315,0.09466931729844093,1
-2.950265223936795,1.8418760666334728,-0.771079380832419,2.1324555369696703,-2.0836707994249677,0.8683174281871185,-0.3758247874753337,0.8601174336648121,0.12126204722073428,
-1.9320827370653029,1.899014591192375,-1.5598921473824658,2.1291397733475765,-2.353200119409353,3.410320682845593,1.3801881664097182,0.8366071484980189,-0.12475391293698708
-1.7841109194921276,2.201132745695477,-1.7150277488999148,1.9915588064588536,-2.0369246446897393,0.038530572357499106,0.10850441605980501,1.0899355207293748,-0.0952797540606
-1.046488652830957,3.43884264526528,-1.35831693644556,0.9685660330261592,-0.0482169710164674,0.04096344971776996,-0.9968328026234113,0.04063972374720495,0.35801338292013707,
-0.996100006850671,3.390823420546226,-1.295328939672079,0.9772761719531944,0.06370608903362558,0.051442496452223756,-1.013975043370874,-0.005794866594930664,0.34376818394260
```

Initially we took all 29 features and ran PCA on the entire Dataset. We got 10PC's but we realized that a lot of features are correlated.

Preprocessing

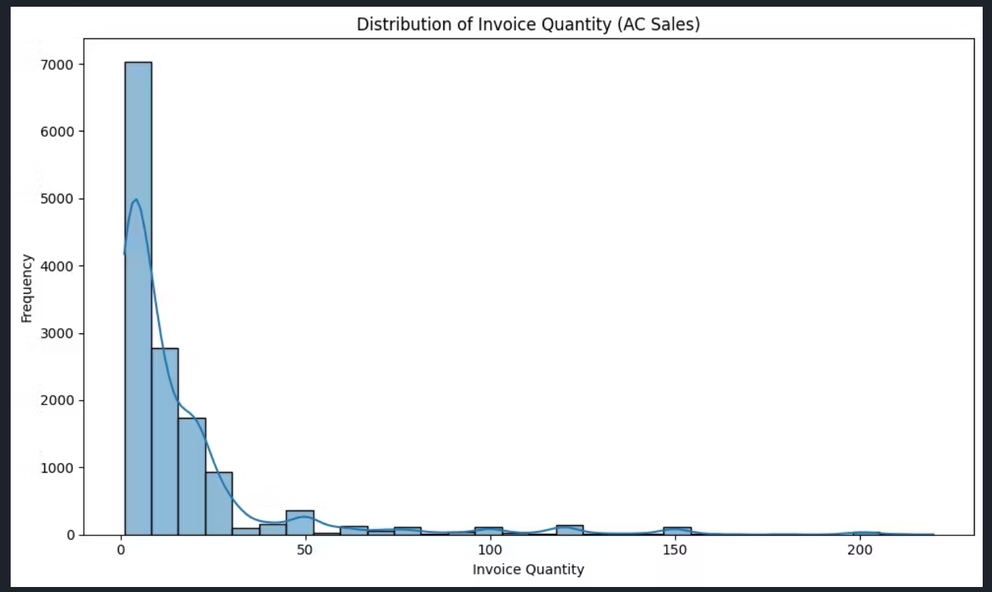


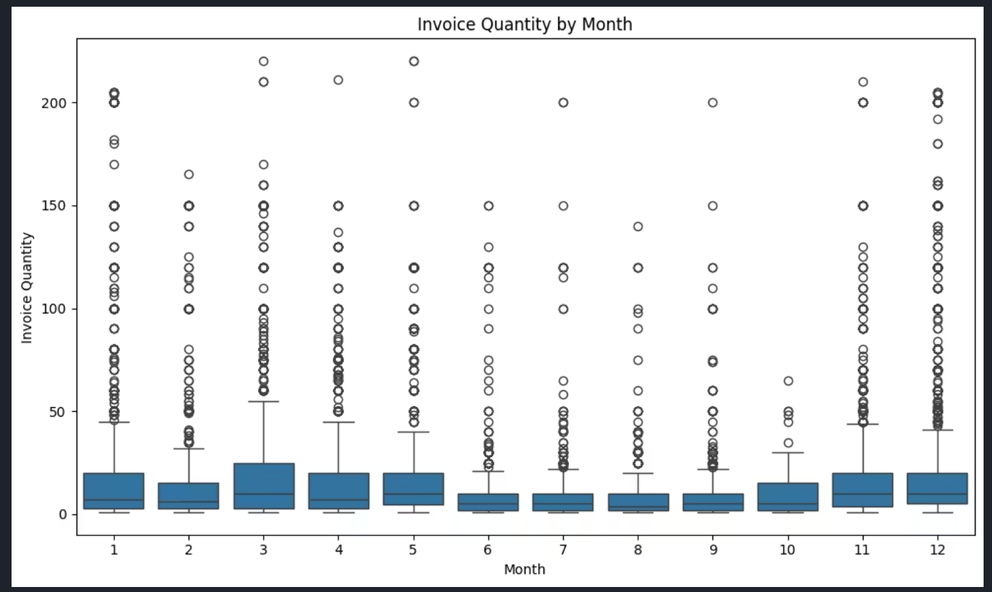
Now looking at all the heavily correlated features

we removed some and ran PCA again

Our dataset was extremely skewed so we created bins- invoice quantity >30 and <30.

OUR MODELS USED THE <30 BIN





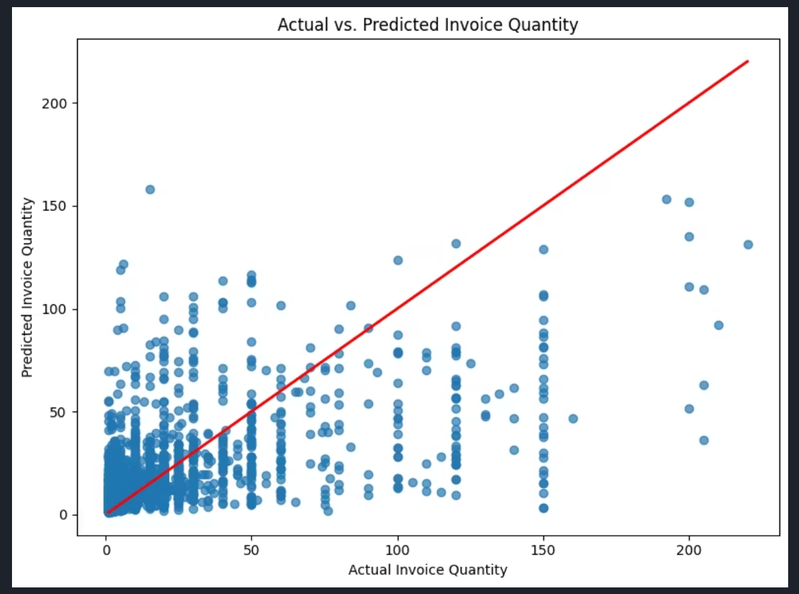
Random Forest

WHAT IS RANDOM FOREST?

a supervised machine learning algorithm that combines multiple decision trees to make predictions.

```
Columns in data: ['Invoice Quantity', 'City', 'temperature
Random Forest Regression Model Performance:
R² Score: 0.3119543209460588
RMSE: 23.0310499736602
Predictions saved to rf_predictions_processed_data.csv
```

Dataset without PCA




```
Random Forest Regression Model Performance:
R² Score: 0.9929114195130949
RMSE: 2.337678266570578
Predictions saved to rf_predictions.csv
```

Dataset with PCA

```
Training Data Columns: ['PC1', 'PC2', 'PC3', 'PC4', 'PC5', 'PC6', 'P
Test Data Columns: ['PC1', 'PC2', 'PC3', 'PC4', 'PC5', 'PC6', 'PC7',
Random Forest Regression Model Performance on PCA-transformed data:
R² Score: 0.014125161373387596
RMSE: 27.56868131663778
Predictions saved to rf_predictions_from_pca_train_test.csv
```

Dataset with separate PCA for training and testing

</>

On training and testing our models on the PCA Dataset we got really high R^2 scores

SO WE TRIED TO TROUBLESHOOT WHY THIS HAPPENED

CHANGED TRAIN-TEST SPLIT TO 50-50

R^2 on PCA dataset with about 9000 training datapoints and 9000 testing datapoints = 0.966

CHANGED TRAIN-TEST SPLIT TO 20-80

R^2 on PCA dataset with about 3600 training datapoints and 15000 testing datapoints = 0.911

CHANGED TRAIN-TEST SPLIT TO 1-99

R^2 on PCA dataset with about 180 training datapoints and 17800 testing datapoints = 0.86

Even with TA's help we found no errors in the code leading to this but PCA gave no explainability to the model either so we started trying other methods.

</>

We made models for each city, thinking maybe isolated models was the way to go.

IT DID NOT WORK

RESULTS OF GURGAON

```
[1]

...   RMSE: 21.23
      R^2 Score: 0.2256
```

So we took other models →

Cyclic Boosting

WHAT IS CYCLIC BOOSTING?

Cyclic Boosting is a fast machine learning method for structured data. It explains each prediction clearly by breaking it down into feature-wise effects, making it both accurate and interpretable.

Cyclic Boosting on the non-pca dataset

```
First 5 predictions: [ 5.13981081 16.38702779  5.77062828 50.52225002 12.40310958]
RMSE on test data: 25.17453509363759
MAE (forecast error) on test data: 13.608945878576469
R^2 on test data: 0.1779225895733586
```

Cyclic Boosting on the PCA Data

```
First 5 predictions: [-2.20940304  1.18976492  2.04112583  1.31006032  3.48828839]
First 5 actual values: [-1.52394449  0.66334116  2.31102806  5.88201141  3.01592176]
RMSE on test data: 0.9978927257803214
R^2 on test data: 0.8203579713900189
MAE on test data: 0.7189452054625557
```

It runs well on PCA(again...) but underperforms on the normal dataset

So we tried running it on a sequential dataset

expecting better results, but alas...

This was run on a weekly aggregate

We also tried using a monthly aggregate,

but there was little to no impact on performance

```
RMSE: 323.19
MAE: 201.54
R²: 0.0185
```

SINCE CYCLIC BOOSTING WASN'T GIVING US ANY REAL RESULTS, WE TRY USING GRADIENT BOOSTING AND LSTM MODELS →

Baseline LightGBM

TRAINED A LIGHTGBM REGRESSION MODEL

Used only basic inputs:

- Raw weather data (e.g., temperature, humidity)
- Date-based features (day, month, weekday)
- Basic sales lag signals: previous day’s sales, 7-day moving average

No advanced tuning, stacking, or city-specific models

```
RMSE: 20.41
 R² Score: 0.27
```

model's **predicted sales deviate from actual sales by around 20 units** on average & model explains only **27% of the variance** in AC sales

Baseline LightGBM captures basic trends but lacks accuracy. Useful as a benchmark for future improvements.

Tuned LightGBM

TRAINED A LIGHTGBM REGRESSION MODEL

Built on baseline by adding:

- Deeper lag features (Sales_3_Days_Ago, 14Day_MA_Sales)
- Interaction features (temp × humidity, wind ÷ pressure)
- Group-wise features (Sales_Location_Mean)

Tuned model parameters for better generalization

RMSE: 19.25
R^2: 0.35

Captured compound weather effects, Better Temporal Context

The model captured deeper temporal patterns and subtle weather-location interactions, improving accuracy over the baseline.

Weighted Ensemble

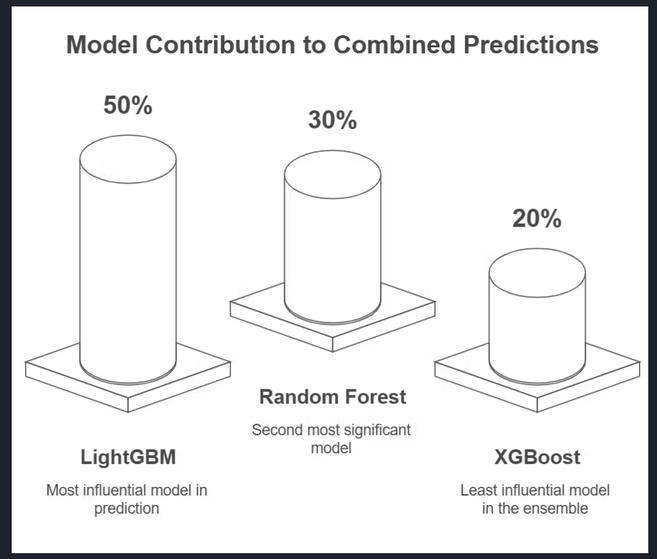
COMBINED PREDICTIONS FROM 3 BASE MODELS

Used 3 Model:

- 50% LightGBM
- 30% Random Forest
- 20% XGBoost

Used full feature set with:

- Lag variables (1-day, 3-day)
- Rolling averages (7-day, 14-day)
- Interaction terms (temp × humidity, wind ÷ pressure)
- Location-wise sales means



RMSE: 19.00
R^2: 0.37

Why this worked??

- Aggregates diverse model perspectives
- Reduces overfitting by avoiding a learned meta-model

Stacked Ensemble

TRAINED 3 BASE MODELS: LIGHTGBM, RANDOM FOREST, XGBOOST

- Used 5-fold cross-validation to get out-of-fold predictions
- Trained a **meta-model** (XGBoost) on these predictions
- On test data, predictions from the base models are passed to the meta-model for final output

WHAT IS A META MODEL?

- A model that learns how to best **combine outputs of base models**
- Instead of fixed weights, it learns patterns in the base models' predictions

```
Stacked RMSE: 19.70
Stacked R2: 0.32
```

Why it underperformed?

- All base models were tree-based and learned similar patterns
- Their predictions were highly correlated
- Increased complexity led to **overfitting on the training predictions**

In our case, stacking added complexity without improving accuracy. Weighted ensemble remained simpler and more robust

Stacked Ensemble : IMPROVED (Slightly)

TRAINED 3 BASE MODELS

- Used 5-fold cross-validation to get out-of-fold predictions
- **Trained a meta-model (XGBoost) on these predictions**
- On test data, predictions from the base models are passed to the meta-model for final output

Trained a meta-model (Ridge) on these predictions

where alpha is set at 1.0

WHAT IS A META MODEL?

- A model that learns how to best **combine outputs of base models**
- Instead of fixed weights, it learns patterns in the base models' predictions

WHY RIDGE?

- Tree-based meta-models tend to overfit when base models are similar
- Ridge adds generalization by penalizing extreme weights

Result?!

```
Stacked RMSE: 19.24
Stacked R²: 0.35
```

Reduced overfitting seen in the previous stacked setup

Stacked Ensemble : IMPROVED EVEN MORE (Slightly)

TRAINED 3 BASE MODELS

- Used 5-fold cross-validation to get out-of-fold predictions
- **Trained a meta-model (XGBoost) on these predictions**
- On test data, predictions from the base models are passed to the meta-model for final output

Trained a meta-model (Ridge) on these predictions

where alpha is set at 1.0

Trained a meta-model (RidgeCV) on these predictions

RidgeCV simply finds the best value for alpha

WHAT IS A META MODEL?

- A model that learns how to best **combine outputs of base models**
- Instead of fixed weights, it learns patterns in the base models' predictions

Result?!

```
Stacked RMSE: 19.24
Stacked R²: 0.35
```

```
Stacked RMSE: 19.23
Stacked R²: 0.35
```

Did not help at all

Per-City Ensemble

WHAT WE DID

- Trained separate models for each city to capture local sales-weather patterns
  - Trained a separate 3-model ensemble for each city
  - Augmented predictions with engineered features

    (Prev_Day_Sales, 7Day_MA_Sales, 7Day_STD_Sales, temp × humidity)
  - Final predictions were made using a meta-model

WHY THIS APPROACH

- Cities differ in weather, demand, and seasonality
- Localized modeling avoided global noise

RESULT

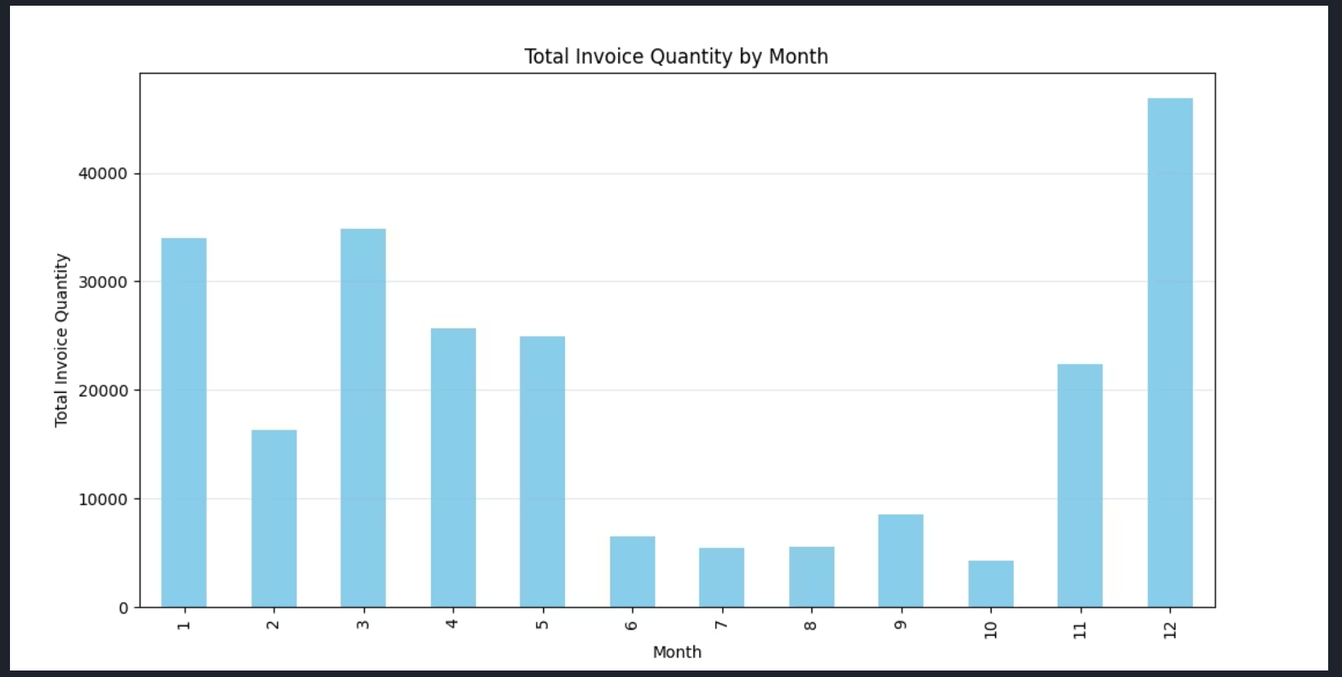
Final Per-City MLP Ensemble Results:

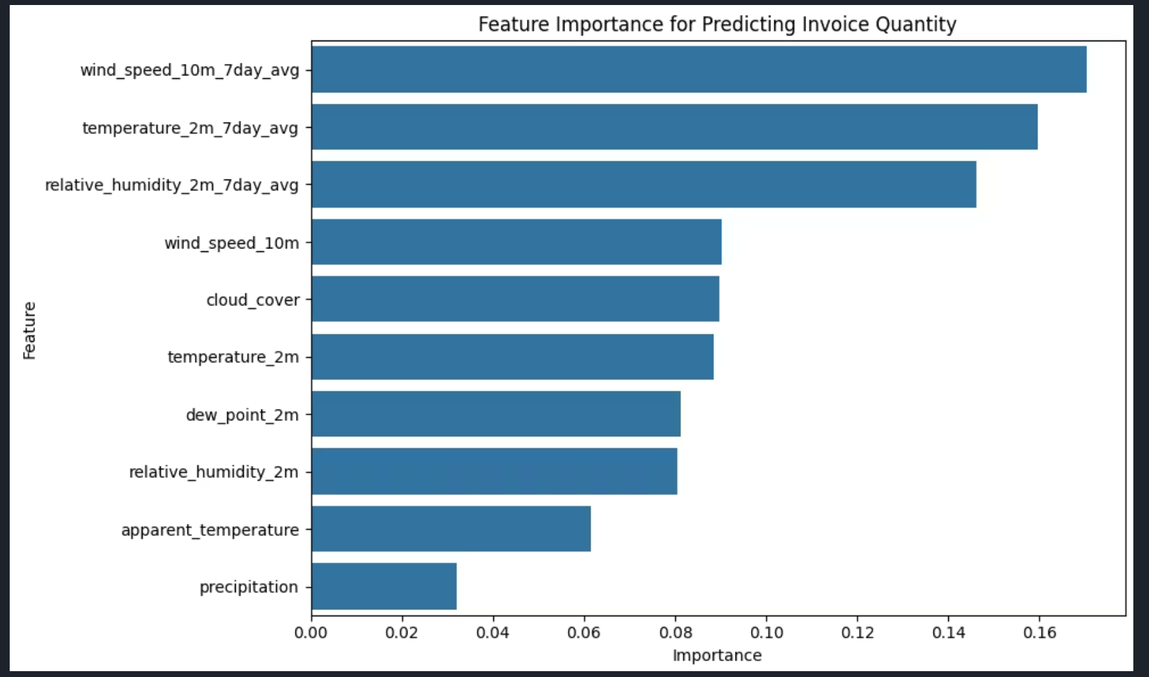
	City	RMSE	R2
0	Madurai	10.841909	0.225000
10	Jammu	31.884983	0.206484
2	Durgapur	31.132792	0.162921
1	Gurgaon	25.487109	0.135372
5	Kochi	13.066214	0.003147
9	Jaipur	26.056241	-0.012634
8	Pune	11.152082	-0.121838
6	Lucknow	19.771674	-0.846766
4	Ghaziabad	37.933565	-0.906888
7	Patna	12.900400	-1.284978
3	Hubli	10.547800	-1.991039

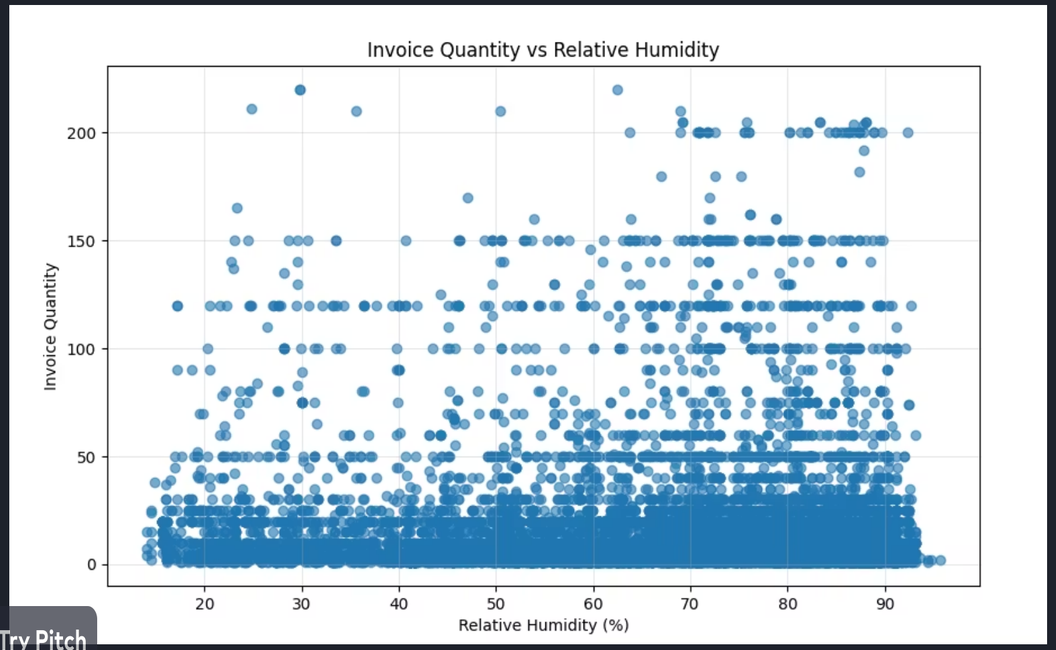
- Many cities (like Hubli, Patna) had **limited data points**
- Some cities had **very low variance** in sales (almost flat lines)
- In such cases, even a good prediction leads to **poor R2**
- Lack of External Context

Nothing was working.

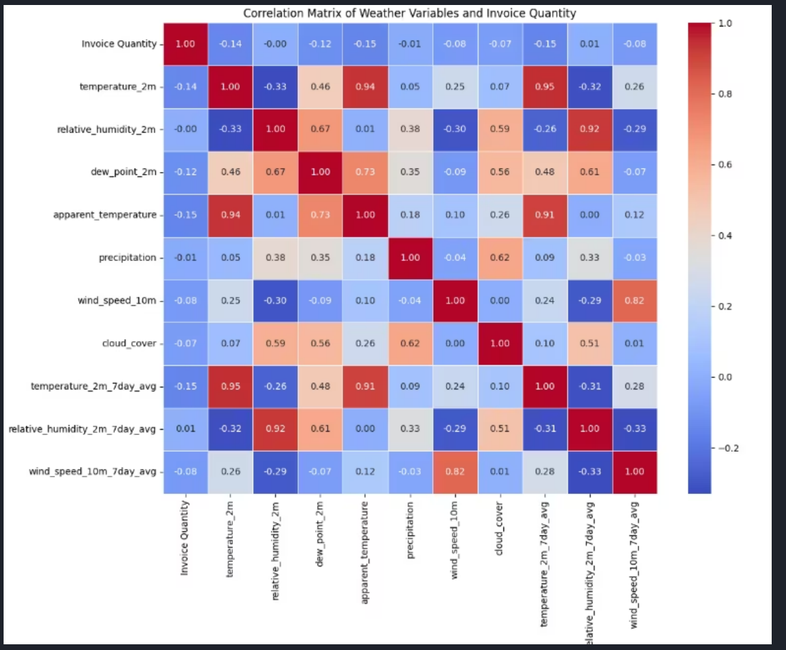
SO WE TOOK A DEEPER LOOK INTO OUR DATASET, TO FIND MORE DISAPPOINTMENT.








```
Invoice Quantity,City,temperature_2m,relat
8,Durgapur,27.603833,68.47537824999999,20.
20,Durgapur,34.34133308333333,24.777364208
2,Durgapur,33.21008308333334,53.0602175416
2,Durgapur,33.21008308333334,53.0602175416
2,Durgapur,33.21008308333334,53.0602175416
25,Durgapur,33.21008308333334,53.060217541
6,Durgapur,28.749666166666668,79.567707916
10,Durgapur,28.524666291666662,79.21439029
25,Durgapur,28.524666291666662,79.21439029
25,Durgapur,28.524666291666662,79.21439029
5,Durgapur,28.524666291666662,79.214390291
25,Durgapur,28.524666291666662,79.21439029
25,Durgapur,28.524666291666662,79.21439029
5,Durgapur,27.037166416666665,82.038786291
6,Durgapur,27.037166416666665,82.038786291
4,Durgapur,27.037166416666665,82.038786291
4,Durgapur,27.037166416666665,82.038786291
10,Durgapur,27.037166416666665,82.03878629
```



No Promotion periods mentioned

HOW WOULD THIS AFFECT SALES?

People are incentivized to buy goods when they are cheaper.

Dataset does not account for when the AC goes on sale, it only shows purchase history.

The Sales Margin feature (column) was incorrect as it showed 15% sales margin throughout the dataset.

ZERO-SALE Periods

WHAT ARE ZERO SALE PERIODS?

Time periods where no sales have been listed

The data did not actually showcase customer orders, it actually showcased the sale. Now when the retailer is out of a product such as this Air Conditioner, he makes no sale of said AC. Stock out cannot be a binary predictor.

The Dataset did not showcase OTIF (On Time In Full) either- i.e.- if the retailer is out of brand x he shows the customer an AC of brand y, making a sale on brand y whereas the customer had truly come for a sale of brand x.

2 PATHS TO CHOSE FORM

Remove periods with no sales

This would go off the sequence and would render our time series useless, hence it could not be done

Take aggregate of the total sales and replace

But this would create false sales, so this couldn't be done either.

Monthly sequences

THE ONLY RELEVANT DATASET WE WERE USING NOW WAS MONTHLY AGGREGATES

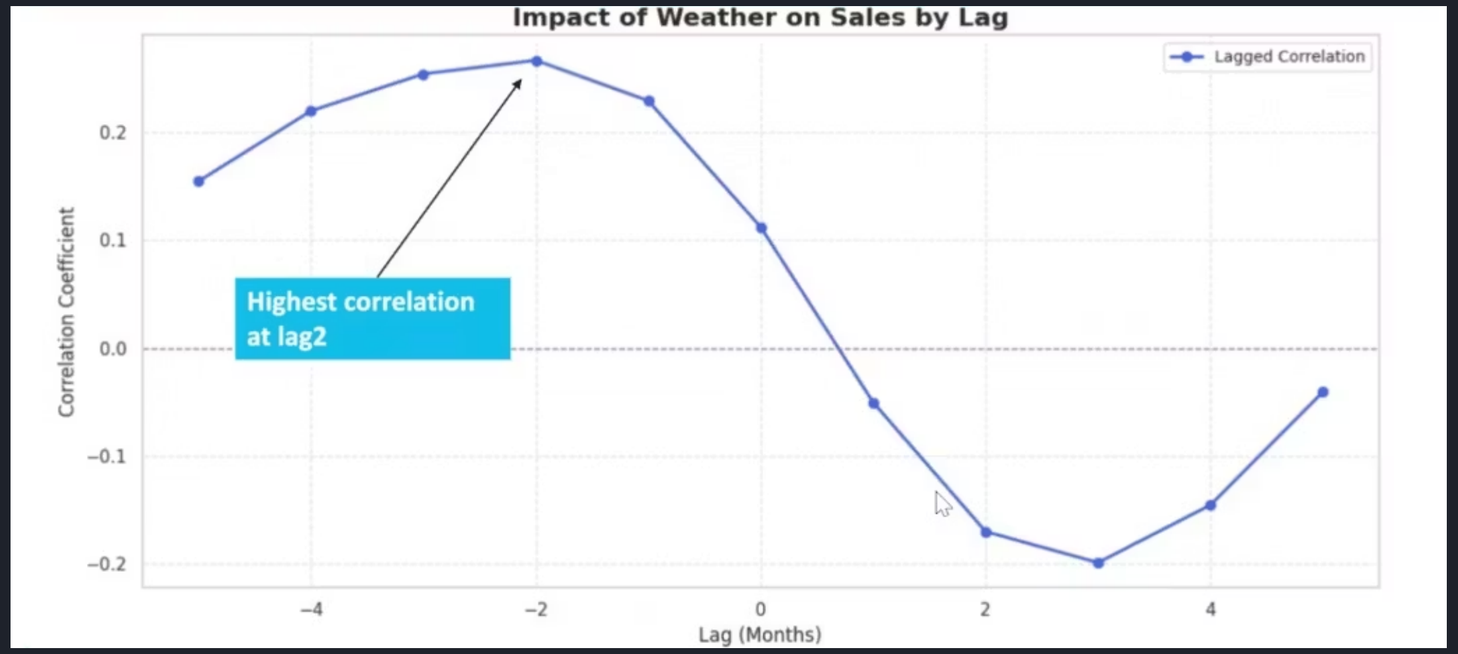
- Daily data was too noisy and there were too many missing cells.
- To account for Zero Sale periods we aggregated the sale by looking at the average sale of other days in the same month

  Example →

  Jan 12-16 has no sales, so we sum up sales of other days in jan and divide by number of days sales were there (26). Now the procured value is assigned to Jan 12-16.
- So taking monthly average with each datapoint being the aggregate sale of that month.

Accounting for Lag

WHEN MAPPING TEMPERATURE PEAKS VS SALES / MONTH WE FOUND A 2 MONTH LAG. SO WHEN TEMPERATURE PEAKED IN MAY –> PEOPLE BOUGHT ACS MAJORLY IN MARCH. (ANTICIPATION)



</>

Final results

MODELS USED AT THE END:

RANDOM FOREST

CYCLIC BOOSTING

LSTM

Random Forest

This Dataset has

- Monthly Aggregates
- Accounts for No Sale Period
- Reduced Features
- Does Not account for OTIF
- Does Not account for Promotion Periods with sale spikes

```
RMSE: 8.031524356602l
R^2 Score: 0.599438375432848
```

Random Forest trained on 36*9 Datapoints (1 for each month) and tested on 18*9 Datapoints

LSTM

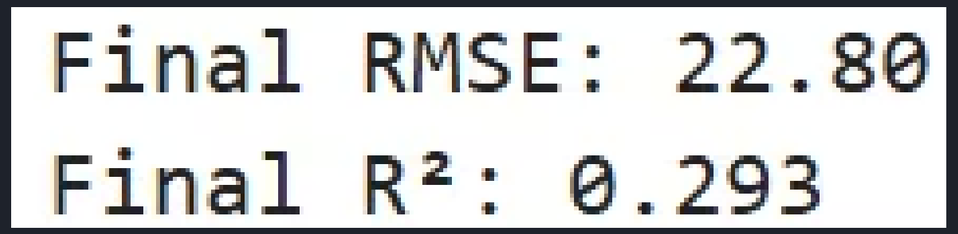
WHAT WE DID

- Modeled sales as a **time series**, using **past 7/14/30 days of weather + sales data** to predict future sales
- Preprocessed using:
  - Min-Max Scaling
  - Rolling Averages: 7-day, 30-day
  - Lagged feature: Prev_Sales
- Created sequences of length 7/14/30 (tried all)

WHY THIS APPROACH

- 2 stacked LSTM layers with 128 and 64 units
- Dropout layers for regularization
- Dense layer for final prediction
- Early stopping used to avoid overfitting

Performance



- LSTM model moderately captured sequential patterns in sales and weather data
- performance suggests more tuning/features/aspects (economic) needed

Cyclic Boosting

```
RMSE on test data: 6.835707923656834
R² on test data: 0.41035797139001 89
```

Thank you

References

[1] https://www.econbiz.de/Record/a-machine-learning-framework-for-predicting-weather-impact-on-retail-sales-chan/10014516549#:~:text=the%20use%20of%20weather%20information,the%20importance%20and%20influence%20of

[2] https://www.scirp.org/journal/paperinformation?paperid=122879

[3] https://ideas.repec.org/a/eee/joreco/v48y2019icp169-177.html#:~:text=information%20is%20much%20shorter%20than,that%20the%20company%20is%20facing

[4] https://doaj.org/article/3aba0c7a747a48ef887c5cac464cb82d#:~:text=compared%20the%20performance%20of%20nine,sudden%20increase%20in%20demand%20of

[5] https://ideas.repec.org/a/eee/joreco/v52y2020ics0969698919303236.html#:~:text=In%20this%20study%2C%20we%20examine,can%20be%20as%20high%20as

[6] https://www.researchgate.net/publication/318155185_Food_Sales_Prediction_with_Meteorological_Data_-_A_Case_Study_of_a_Japanese_Chain_Supermarket#:~:text=much%20they%20are%20willing%20to,3

[7] https://www.inc.com/reuters/retailers-turn-to-weather-strategies-as-climate-changes/90998860

Pitch

Want to make a presentation like this one?

Start with a fully customizable template, create a beautiful deck in minutes, then easily share it with anyone.

Create a presentation (It's free)