**Team Name**

Forecast TX

**Timeline**

Spring 2025 – summer 2025

**Students**

* Kristal Phommalay – CSE
* Noe Sanchez – CSE
* Kevin Simbakwira – CSE
* Kenil Patel – CSE
* Mason Berry – SE

**Sponsor**

State Farm

**Abstract**

Sponsored by State Farm, this project leverages machine learning to predict severe weather events. Specifically, thunderstorms and hail across Texas, with forecasts extending five years into the future. The predictive models are trained using over 70 years of historical data from ERA5 reanalysis and NOAA storm reports, capturing long-term weather trends and storm behavior.

The results are presented in a user-friendly, Firebase-hosted web application featuring interactive maps and statistical visualizations. Users can explore predicted storm frequencies, locations, and intensities across different timeframes. The backend is supported by Google Cloud Platform services, including Cloud Storage, PostgreSQL databases, and virtual machines to enable scalable data processing and real-time API delivery.

**Background**

Designed to support State Farm’s operational planning, this system aims to assist in disaster preparedness, risk assessment, and other insurance-related decision-making tasks by providing a data-driven outlook on future severe weather activity in Texas.

**Project Requirements**

**Predict Future Weather Events**  
The system must be capable of predicting the **frequency and severity** of severe weather events (hail, wind, flooding, etc.) over the **next 5–10 years** in a selected Texas region.

**Focus on a Specific Peril and Location**  
The team must select **one peril type** (e.g., wind, hail, or rain) and an **area in Texas** where this peril is frequent for the modeling effort.

**Use Long-Term Historical Data**  
The system must use weather data covering **at least 10 years**, ideally from sources like **NOAA** (1950–2024) or similar, ensuring data depth and validity.

**Store and Organize Data Efficiently**  
The system must have the **ability to store raw and processed data** in a structured and scalable way (e.g., cloud storage, PostgreSQL).

**Retrieve and Query Data Easily**  
The system must allow for **efficient data retrieval** to support modeling, analysis, and front-end interaction (e.g., filtering, querying via API or SQL).

**Visualize Predictions Clearly**  
The system must present predictions in a **clear, interactive, and consumable format**—such as graphs, maps, and dashboards—for stakeholder use.

**Filter Data by Key Criteria**  
Users must be able to **filter data by year, peril type, and location** to support customized analysis and decision-making.

**Deliver a Web-Based Interface**  
A **user-friendly web application** must be built for users to **interact with model outputs**, perform filtering, and explore data visually.

**Run on a Scalable Cloud Platform**  
The system must be **hosted on cloud infrastructure** (e.g., Google Cloud, AWS, Azure) to ensure scalability and reliability.

**Support Model Accuracy and Validation**  
The machine learning model must be **trained on valid data**, **tested on a separate dataset**, and evaluated using techniques like **cross-validation** to ensure prediction quality.

**Design Constraints**

**Accessibility**

To support equitable use, the front-end UI includes color palettes suitable for colorblind users. Visual contrast, keyboard navigation, and responsiveness are included to comply with basic accessibility standards, ensuring usability for a broader audience.

**Cost/Economic**

As a student-led academic project, the system is constrained by limited financial resources. All infrastructure is cloud-based, and free-tier or sponsored tools like Firebase Hosting, Google Cloud Platform credits, and open-source libraries were used to avoid paid licensing or enterprise tools.

**Functionality**

The system must support both basic and advanced users. This constraint necessitated a flexible UI with layered access permissions and robust backend support for spatial querying, dynamic filtering, and ML model interaction.

**Interoperability**

The ETL and ML pipelines are built using widely adopted file formats (e.g., GRIB, CSV, Parquet) and standards-compliant APIs. This enables integration with external tools used by State Farm or research partners without major rework.

**Schedule**

All development milestones were tied to a fixed academic calendar, which limited time for experimentation and required parallel development of subsystems. Deliverables such as the UI prototype, ETL demonstration, and machine learning output were time-boxed.

**Maintainability**

Code modularity and documentation were prioritized. The use of open-source frameworks like React and Python ensures long-term support and makes the system easier to maintain by future developers or collaborators.

**Usability**

A major goal was to build an intuitive interface with map-based visualizations, dynamic filtering, and responsive controls. Stakeholder feedback was incorporated to simplify workflows, especially for non-technical users at State Farm.

**Engineering Standards**

**IEEE Software Documentation Standards**

The team followed structured documentation practices inspired by IEEE 830 (Software Requirements) and IEEE 1016 (Design Descriptions), producing System Requirement Specifications (SRS), Architectural Design Specifications (ADS), and Detailed Design Specifications (DDS).

**NIST Security Guidelines**

Security features—including user authentication, encrypted data storage, and access control via Firebase and GCP IAM—align with general NIST cybersecurity and cloud computing best practices.

**ISO/IEC 27001 (Information Security)**

While not formally certified, the system design shows awareness of ISO/IEC 27001 principles, such as minimizing data exposure through secure access layers and protecting sensitive model output.

**RESTful API Standards**

All communication between the frontend and backend adheres to REST architectural principles. Endpoints return data in structured JSON or CSV formats, with clear URI naming, HTTP methods, and status codes.

**Web Content Accessibility Guidelines (WCAG)**

The front-end design takes into account WCAG 2.1 Level AA recommendations for visual clarity, color contrast, and content readability, improving accessibility for users with visual impairments.

**Open Source Package Compliance**

All software dependencies are used under approved licenses (MIT, Apache 2.0, etc.), ensuring legal reuse and adherence to ethical development practices.

**System Overview**

Forecast TX is a modular, cloud-native system developed to predict the likelihood and severity of severe weather events across Texas over the next 5 to 10 years. Designed in partnership with State Farm, the system supports risk mitigation and long-term planning for insurance operations and public safety.

The architecture consists of four primary layers:

**1. Data Storage Layer**

This layer uses Google Cloud Platform to manage all structured and unstructured data. It includes:

* **Input Bucket**: Stores historical weather data (from NOAA and ERA5).
* **Output Bucket**: Holds model predictions in formats like CSV and Parquet.
* **PostgreSQL Database**: Contains spatially indexed records via PostGIS for querying and integration with the user interface.

**2. Data ETL Layer**

The Extract, Transform, Load (ETL) layer prepares raw climate data for modeling. It:

* Fetches large-scale climate datasets via APIs (e.g., CDSAPI).
* Transforms GRIB files into clean tabular formats.
* Loads processed data into GCP storage for downstream use.  
  Scripts are optimized for multi-threading and flexible enough to run on local machines or VMs.

**3. Machine Learning Layer**

At the core of Forecast TX, this layer trains and runs prediction models such as LSTM, SARIMA, and linear regression. It:

* Trains on over 70 years of data.
* Generates probabilistic forecasts for specific event types like hail, wind, or heavy rainfall.
* Stores predictions and model artifacts for use in UI and analytics.

**4. User Interface Layer**

A web-based front end built with React.js and hosted on Firebase provides:

* **Map Visualizations**: Geospatial plots of predicted events.
* **Interactive Filters**: Let users explore by time, region, or event type.

Together, these components form a seamless pipeline—from raw climate data ingestion to actionable insights—empowering State Farm to make informed decisions about disaster preparedness and risk exposure.

**Results**

Our project successfully predicts useful weather data and showcases how machine learning can be leveraged to provide valuable data to State Farm. Our web interface delivers our model data to the user in an easy to explore format which can assist State Farm internal users in decision making across various use cases.

**Video Demo**

**Future Work**

The major areas that could be improved with additional work would be creating additional interactive user interfaces and improving machine learning model accuracy.

By establishing a model improvement process, the ML model can be improved with re-training and testing, adjusting input feature weights and biases to produce a more accurate prediction. Establishing a clear pipeline to apply this iterative process would allow for time and resource effective model re-training to improve data accuracy and veracity as well as explore predictions of additional valued weather indicators.

**Project Files**

Project Charter  
System Requirements Specification  
Architectural Design Specification  
Detailed Design Specification  
Poster  
Closeout Materials

**References**

1. Google, Inc., “Google Cloud Platform Documentation,” *Google Cloud Platform*, 2024. [Online]. Available: <https://cloud.google.com/docs>
2. Overleaf, “Collaborative Writing and Publishing Tool for LaTeX,” *Overleaf*, 2024. [Online]. Available: <https://www.overleaf.com>
3. Agafonkin, V., “Leaflet: An Open-Source JavaScript Library for Interactive Maps,” *Leaflet.js*, 2024. [Online]. Available: [https://leafletjs.com](https://leafletjs.com/)
4. Hersbach, H., et al., “ERA5 hourly data on single levels from 1940 to present,” *Copernicus Climate Change Service (C3S) Climate Data Store (CDS)*, 2023. [Online]. Available: [https://cds.climate.copernicus.eu](https://cds.climate.copernicus.eu/)
5. NOAA National Centers for Environmental Information, “Storm Events Database,” *National Oceanic and Atmospheric Administration (NOAA)*, 2024. [Online]. Available: <https://www.ncdc.noaa.gov/stormevents/textsearchhelp.jsp>
6. Google, Inc., “Firebase Hosting Documentation,” *Firebase by Google*, 2024. [Online]. Available: <https://firebase.google.com/docs/hosting>