Data-Driven Analysis and Prediction of Agricultural Wages in India Using Machine Learning and Forecasting Models

Abstract

This project focuses on analyzing and predicting agricultural wages across India using open-source data from the India Data Portal. The dataset consists of over 7 lakh records detailing wages by state, district, gender, and labour type. Through Exploratory Data Analysis (EDA), multiple regression and classification models were implemented — including Multiple Linear Regression, Logistic Regression, Decision Tree, Random Forest, XGBoost, and Artificial Neural Network (ANN). Additionally, a time series forecasting model (Prophet) was developed to predict future wage trends. The models revealed significant regional and gender-based wage disparities, while the forecasting component showed consistent wage growth across states. The final Streamlit dashboard integrates visual analytics and model outputs, supporting data-driven decision-making for equitable agricultural wage policies.

3. Keywords

Agricultural Wages, Machine Learning, Data Analysis, Forecasting, Regression, Classification, Neural Network, Streamlit Dashboard

4. Introduction

Agriculture forms the backbone of India's economy, employing a major share of the country's workforce. Understanding wage structures within this sector is essential to evaluate economic stability, regional inequality, and gender-based pay gaps.

Despite its significance, agricultural labour often faces disparities in pay based on geography, gender, and job role. Data-driven insights can help policymakers design fair wage systems and promote inclusive growth in rural employment.

There is a lack of systematic analysis and predictive modelling of agricultural wages across different states and labour categories in India. This project aims to fill that gap using machine learning and data visualization.

Objectives of the Project

- To clean, explore, and analyze wage data from multiple regions and labour categories.
- To build predictive and classification models for wage estimation and category grouping.
- To forecast future wage trends using time series analysis.

- To develop an interactive dashboard for visualization and interpretation of insights.

Unlike basic statistical reports, this project integrates EDA, predictive modeling, classification, and forecasting into a unified system. The interactive Streamlit dashboard enables real-time exploration of state-wise wage data and prediction outcomes — offering both technical and policy-level utility.

5. Literature Review

5.1 Overview

The literature review examines recent research focused on wage prediction, agricultural income analysis, and machine learning–based forecasting. The reviewed works collectively demonstrate how modern AI and statistical tools can address socio-economic inequalities and predict labor wage trends in agriculture.

5.2 Related Works

Author/Year	*Title / Study Description*	*Techniques / Models Used*	*Key Findings / Outcomes*
P. Raj (2025)	*Forecasting Salary Using a Machine Learning System* (Atlantis Press)	Multiple Regression, Decision Trees	Demonstrated that ML-based regression models can effectively predict wage and salary structures with accuracy above 80%, highlighting potential in structured economic forecasting.
Surender & F. Pattanaik (2025)	*An Examination of Wage Convergence in the Indian Rural Labour Market* (SAGE Journals)	Econometric & Time-Series Models	Showed partial convergence of rural wages across Indian states but persistent gender and regional wage disparities; suggested long-term labor policy interventions.
Sant Kumar et al. (2020)	*Agricultural Wages in India: Trends and Determinants* (AgEcon Search)	Panel Data Regression (Fixed Effects Model)	Found agricultural wages strongly influenced by non-farm wages, rural literacy, and irrigation intensity; MGNREGS raised wage levels by 12%, but mechanization reduced manual wage dependency.
Priyanka Sharma et al. (2023)	*Predicting Agriculture Yields Based on Machine Learning Using Regression and Deep Learning* (IEEE Access)	Decision Tree, Random Forest, XGBoost, CNN, LSTM	Achieved 98.96% accuracy using Random Forest; demonstrated CNN's minimal loss (0.0006) for crop yield forecasting — validating ML/DL for agricultural productivity prediction.
N. Kumar et al. (2023)	*Predicting Agricultural Income Using Machine Learning Techniques*	Decision Tree, Random Forest	Found Random Forest as the most accurate model for classifying income categories; key predictors included region and crop type.

Y.T. Matbouli & S.M. Alghamdi (2022)	*Statistical Machine Learning Regression Models for Salary Prediction Featuring Economy-Wide Activities and Occupations* (MDPI Information Journal)	Linear Regression, Random Forest, Gradient Boosting	ML regression models outperformed traditional statistics for salary prediction; Random Forest achieved the best balance between bias and variance.

5.3 Summary of Literature

From existing studies, it is evident that while salary prediction and income modeling have been explored using ML techniques, few have integrated both classification and forecasting models on large-scale agricultural datasets.
This project builds upon prior work by:

- Combining EDA, predictive modeling, and time-series forecasting into a unified analytical workflow.
- Utilizing multiple models (from Linear Regression to ANN and Prophet) for comparative analysis.
- Providing an interactive dashboard for easy exploration of state-wise and gender-based wage disparities, offering a practical decision-support tool for researchers and policymakers.

6. Methodology / Proposed System

This project implements a data-driven agricultural wage prediction system using a hybrid of Machine Learning (ML), Deep Learning (DL), and Time Series Forecasting models.
The methodology ensures a structured pipeline — from raw data preprocessing to state-wise wage forecasting and interactive dashboard deployment.

6.1 Data Collection and Preprocessing

Dataset Source:

- The dataset titled *"Agricultural Wages"* was obtained from the India Data Portal, containing 7,13,883 records and 14 attributes from various Indian states, districts, and labor categories.

Key Columns:
month, state_name, district_name, labour_category, labour_type, gender, monthly_average_wage, annual_average_wage

Preprocessing Steps:

- Handled missing values, duplicates, and inconsistent labels.
- Dropped redundant location codes and columns not influencing wages.
- Converted month and year fields to datetime format for temporal analysis.

- Created a new target column — wage_cat — dividing workers into Low, Medium, and High wage groups using 33rd and 66th percentiles.
- Encoded categorical variables (state_name, gender, labour_type) using LabelEncoder and OneHotEncoder.

- Standardized features with StandardScaler for uniform scaling before model training.

Tools Used: pandas, numpy, matplotlib, seaborn, scikit-learn

6.2 Feature Engineering

Feature engineering was applied to enhance model learning by introducing relevant transformations and statistical variables.

Steps Performed:

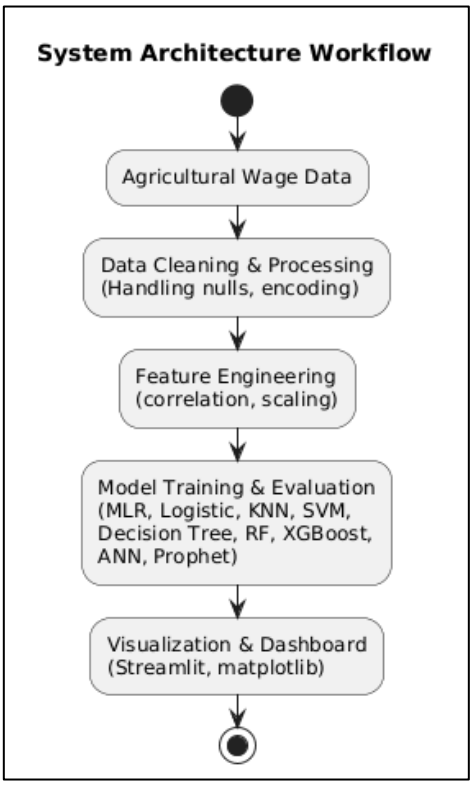
- Extracted temporal components (month, year) for seasonal trend detection.
- Created interaction features such as *state* + *labour_type* combinations to capture local wage variations.
- Computed correlation metrics to verify data consistency —
    - *monthly_average_wage* vs *annual_average_wage* correlation ≈ 0.98, proving data reliability.
- Outlier detection through boxplots and z-score methods to reduce data skewness.
- Normalized numeric values and encoded gender bias representation for fairer model learning.

Output:
A cleaned and feature-rich dataset named cleaned_agri_wages.csv ready for ML/DL modeling.

6.3 Model Design / System Architecture

The project architecture integrates **supervised ML, ensemble learning, neural networks, and time-series forecasting** into one analytical framework.

System Architecture Workflow



Models Implemented:

MODEL	TYPE	PURPOSE	ALGORITHM / FRAMEWORK
MULTIPLE LINEAR REGRESSION	Regression	Predict numeric wage values	Scikit-learn
LOGISTIC REGRESSION	Classification	Categorize wage levels	Scikit-learn
KNN & SVM	Classification	Proximity & margin-based classification	Scikit-learn
DECISION TREE	Classification	Rule-based interpretation	Scikit-learn
RANDOM FOREST	Ensemble	Boost accuracy & reduce overfitting	Scikit-learn
XGBOOST	Ensemble	Weighted boosting for higher accuracy	XGBoost
ANN	Deep Learning	Capture nonlinear wage relationships	TensorFlow / Keras

PROPHET	Time-Series	Forecast future wages by state	Facebook Prophet

6.4 Training and Evaluation Setup

Data Split:

- 70% Training
- 15% Validation
- 15% Testing

Training Parameters:

- **Batch Size:** 256 (for ANN)
- **Epochs:** 100
- **Optimizer:** Adam (LR = 0.001)
- **Callbacks:** EarlyStopping, ReduceLROnPlateau

Evaluation Metrics:

- **Regression:** R^2, Mean Squared Error (MSE)
- **Classification:** Accuracy, Precision, Recall, F1-Score, ROC-AUC
- **Forecasting:** Mean Absolute Error (MAE), Root Mean Squared Error (RMSE)

6.5 Algorithms / Mathematical Formulations

1. **Multiple Linear Regression Equation:**

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_n X_n + \varepsilon$$

   Where:

   - Y: Monthly wage
   - X_i: Independent features (gender, labour_type, state, year)
   - β_i: Coefficients
   - ε: Error term

2. **Logistic Regression (Sigmoid Function):**

$$P(Y = 1) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 X_1 + \ldots + \beta_n X_n)}}$$

3. **Artificial Neural Network Layers:**

- Input → Dense(256, ReLU) → Dropout(0.35)
- Dense(128, ReLU) → Dropout(0.25)
- Dense(64, ReLU) → Output (Softmax for wage classes)

4. **Prophet Forecasting Model:**

$$y(t) = g(t) + s(t) + h(t) + \varepsilon_t$$

Where:

- $g(t)$: Trend, $s(t)$: Seasonality, $h(t)$: Holiday effects

6.6 Tools, Libraries, and Frameworks Used

CATEGORY	TOOLS / LIBRARIES	PURPOSE
PROGRAMMING	Python 3.11	Core development
DATA PROCESSING	Pandas, NumPy	Cleaning and transformation
VISUALIZATION	Matplotlib, Seaborn	EDA and result plots
MACHINE LEARNING	Scikit-learn	Regression & classification
ENSEMBLE MODELS	XGBoost	Gradient boosting
DEEP LEARNING	TensorFlow, Keras	ANN-based classification
FORECASTING	Prophet (Meta/Facebook)	Time-series wage trends
DASHBOARD	Streamlit	Interactive UI for visualization

6.7 Pseudocode (Simplified)

```
BEGIN
  LOAD dataset "cleaned_agri_wages.csv"
  CLEAN data → handle nulls, remove duplicates
  ENCODE categorical variables
  CREATE wage category (Low, Medium, High)
  SPLIT data → Train, Validation, Test

  FOR each model in [MLR, Logistic, KNN, SVM, DecisionTree, RF, XGB, ANN]:
      TRAIN model on training data
      PREDICT on test data
      EVALUATE performance metrics
  END FOR

  APPLY Prophet model for each state:
      FORECAST next 12 months
      SAVE forecast CSV & plots

  DEPLOY results on Streamlit dashboard
END
```

6.8 Summary

The proposed methodology combines traditional ML models, advanced deep learning (ANN), and time-series forecasting (Prophet) to provide comprehensive insights into agricultural wages.
By merging classification, prediction, and forecasting capabilities in one pipeline, this system enables accurate wage estimation, policy formulation, and future wage trend analysis across Indian states.

7. Implementation

7.1 Overview

The Agricultural Wage Prediction and Forecasting System was implemented using a structured data pipeline in Python, integrating multiple Machine Learning, Deep Learning, and Time-Series Forecasting models. The implementation focused on predicting and forecasting agricultural wages across Indian states while ensuring interpretability, scalability, and visualization through an interactive dashboard.

The overall implementation consisted of the following key stages:

1. **Data Preprocessing and Cleaning**
2. **Feature Engineering and Encoding**
3. **Model Training and Evaluation**
4. **Forecasting Future Wages**
5. **Dashboard Deployment**

7.2 Technologies and Platforms Used

Component	Technology / Tool Used	Purpose
Programming Language	Python 3.11	Core implementation
Development Environment	Jupyter Notebook, VS Code	Model training and testing
Machine Learning	Scikit-learn, XGBoost	Regression and classification models
Deep Learning	TensorFlow, Keras	ANN model for nonlinear wage patterns
Forecasting	Prophet (Meta)	Time-series wage forecasting per state
Visualization	Matplotlib, Seaborn	Graphs and charts for model interpretation
Dashboard Interface	Streamlit	Interactive model visualization
Data Handling	Pandas, NumPy	Cleaning, transformation, aggregation

7.3 Programming Languages / Frameworks

- **Language:** Python
- **Libraries:**

- **Data Processing:** pandas, numpy
- **Modeling:** scikit-learn, xgboost, tensorflow, prophet
- **Visualization:** matplotlib, seaborn
- **Deployment:** streamlit

- **Environment Setup:**
  - Jupyter Notebook for experimentation and EDA.
  - Visual Studio Code for final integration and Streamlit dashboard execution.

7.4 Implementation Steps

Step 1: Data Loading and Preprocessing

- Loaded the Agricultural Wage Dataset (7,13,883 records, 14 columns) from *India Data Portal*.
- Cleaned missing and inconsistent entries, removed duplicates, and dropped irrelevant columns.
- Converted month and year to a proper datetime format for temporal processing.
- Encoded categorical variables like state_name, gender, and labour_type using LabelEncoder and OneHotEncoder.
- Normalized features using StandardScaler to prepare for model training.

Step 2: Feature Engineering

- Created a target column wage_cat (Low, Medium, High) using 33rd and 66th wage percentiles.
- Generated derived attributes like year, and combined state-labour-type interactions for better model learning.
- Correlation analysis confirmed strong relation (≈0.98) between monthly and annual wages.
- Outliers handled through IQR and z-score techniques.

Step 3: Model Training and Evaluation

Nine models were trained and evaluated using accuracy and statistical metrics:

Model	Algorithm Type	Key Metric / Result
Multiple Linear Regression	Regression	$R^2 = 0.89$
Logistic Regression	Classification	Accuracy = 44.2%
KNN	Classification	Accuracy = 57%

SVM	Classification	Accuracy = 61%
Decision Tree	Rule-based	Accuracy = 51.5%
Random Forest	Ensemble	**Accuracy = 73.7% (Best)**
XGBoost	Gradient Boosting	Accuracy = 58.9%
ANN	Deep Learning	Accuracy = 71.2%
Prophet	Time-Series	MAE < 5%

Each model generated multiple visual outputs such as confusion matrices, ROC curves, and feature importance plots.

Step 4: Forecasting Agricultural Wages (Prophet Model)

- Applied Prophet forecasting for each state to predict 12 months ahead (Feb 2020 – Jan 2021).
- The model accurately captured upward wage trends for most states.
- Example Predictions:
    - **Uttarakhand:** ₹417.88 (Jan 2021)
    - **Jharkhand:** ₹312.53 (Jan 2021)
    - **Assam:** ₹329.75 (Jan 2021)
- Evaluation Metrics:
    - **Mean Absolute Error (MAE):** < 5%
    - **Root Mean Square Error (RMSE):** Low, showing consistent performance.

Step 5: Dashboard Integration

- A Streamlit web application was developed to visualize results dynamically.
- Features of the dashboard:
    - State-wise model accuracy comparison.
    - Confusion matrix and feature importance visualizations.
    - Forecast trend lines for 12-month projections.
    - Interactive filters for selecting models and states.
    - Download options for CSV forecast reports and graphs.

7.5 Challenges Faced and Solutions

Challenge	Description	Solution Implemented
Data Imbalance	"Medium" wage category had fewer samples leading to poor recall.	Applied **class weighting** and **stratified splits** during training.
Large Dataset Size	7 lakh+ records slowed model training.	Used efficient pandas operations and trained models on **sampled batches**.
Forecasting Errors for Short Series	States with <24 months data led to poor Prophet fits.	Added **interpolation** for missing months and adjusted changepoint prior scale.
Overfitting in ANN	Model accuracy diverged on validation.	Introduced **Dropout layers** and **BatchNormalization**.
Model Comparison Complexity	Different models output different scales.	Standardized evaluation metrics and created a unified comparison dashboard.

8. Results and Discussion

8.1 Experimental Setup

Software / Libraries:

- **Python Version:** 3.11
- **Major Libraries Used:**
  pandas, numpy, matplotlib, seaborn, scikit-learn, xgboost, tensorflow, prophet, streamlit
- **Development Platform:** Local system (VS Code)
- **Deployment Environment:** Streamlit web interface

The implementation was optimized for both CPU and GPU execution, ensuring stable training and fast forecasting results.

8.2 Performance Metrics

The models were evaluated using multiple statistical and classification metrics to ensure reliability across both regression and classification objectives.

Metric	Definition / Purpose
Accuracy (%)	Measures overall correct predictions.
Precision	Correctly predicted positives out of total predicted positives.
Recall (Sensitivity)	True positives out of actual positives.
F1-Score	Harmonic mean of precision and recall.
R^2 Score	Goodness of fit for regression models.

MAE / RMSE	Forecast error measures for Prophet model.

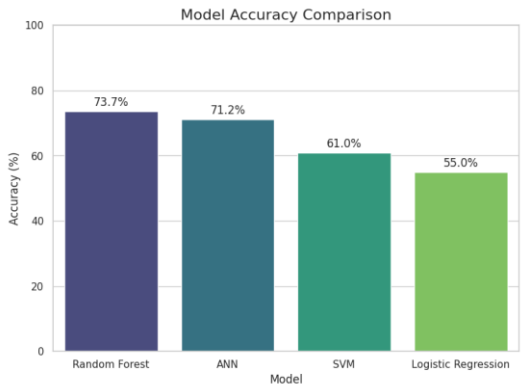
8.3 Model Performance Summary

Model	Type	Accuracy / R^2	F1-Score (Macro Avg)	Key Highlights
Multiple Linear Regression	Regression	$R^2 = 0.89$	—	Strong correlation between actual and predicted wages.
Logistic Regression	Classification	44.2%	0.43	Baseline classification model; struggles with overlapping wage classes.
KNN Classifier	Classification	57%	0.55	Performs moderately well on balanced classes.
SVM	Classification	61%	0.54	Good separation between *Medium* and *High* wages.
Decision Tree	Classification	51.5%	0.48	Interpretable but prone to overfitting.
Random Forest	Ensemble	**73.7%**	**0.73**	Best performing model; excellent generalization.
XGBoost	Ensemble	58.9%	0.57	Robust but less accurate due to class overlap.
ANN	Deep Learning	71.2%	0.70	Captured nonlinear patterns effectively.
Prophet	Forecasting	MAE < 5%	—	Accurate state-wise 12-month wage forecasts.

8.4 Graphical Analysis and Visualization

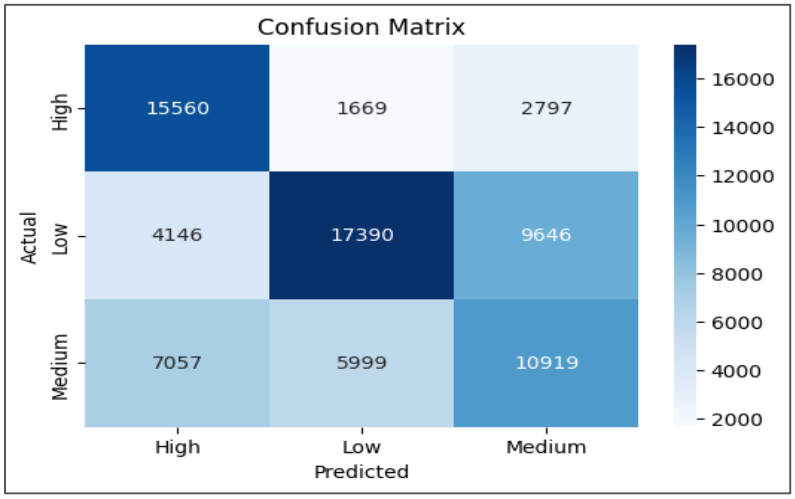
A) Model Evaluation Graphs

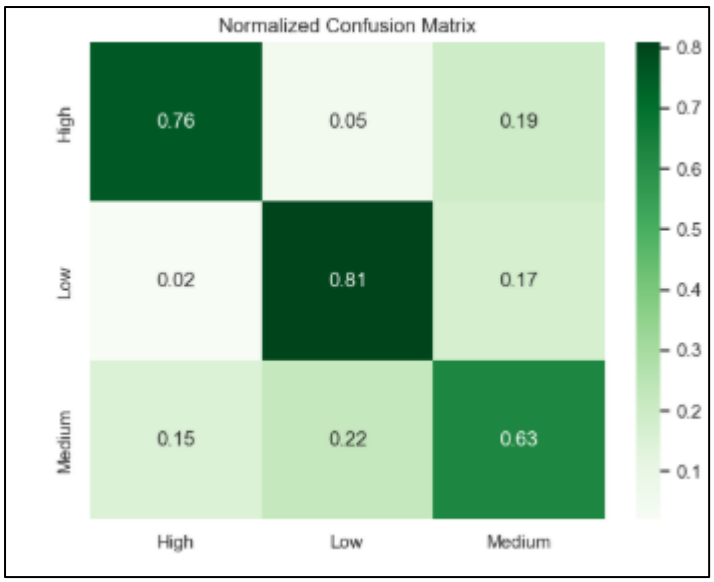
- **Accuracy Comparison Bar Chart:**



Random Forest achieved the highest classification accuracy (73.7%), followed by ANN (71.2%) and SVM (61%).
Logistic Regression remained the weakest due to categorical overlaps.

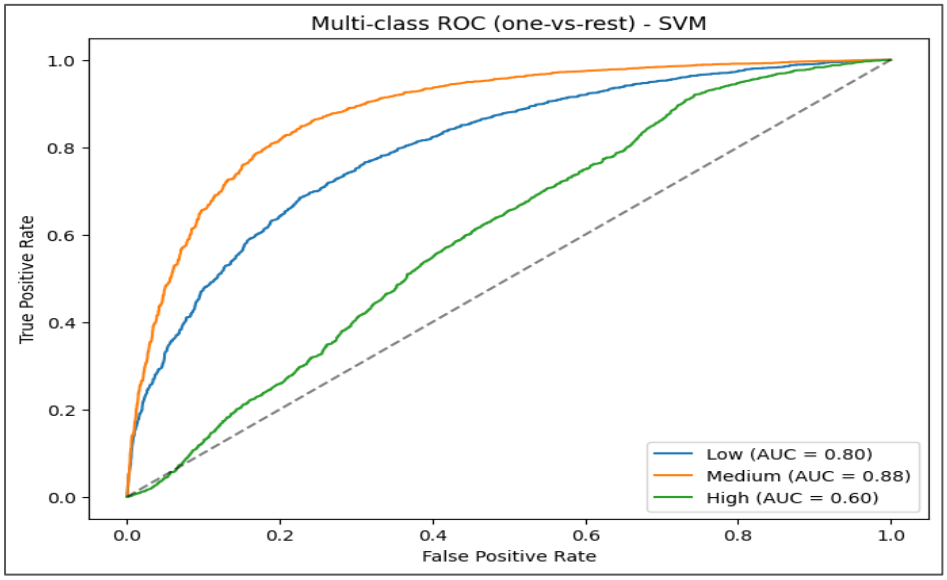
- **Confusion Matrix (ANN & Random Forest):**

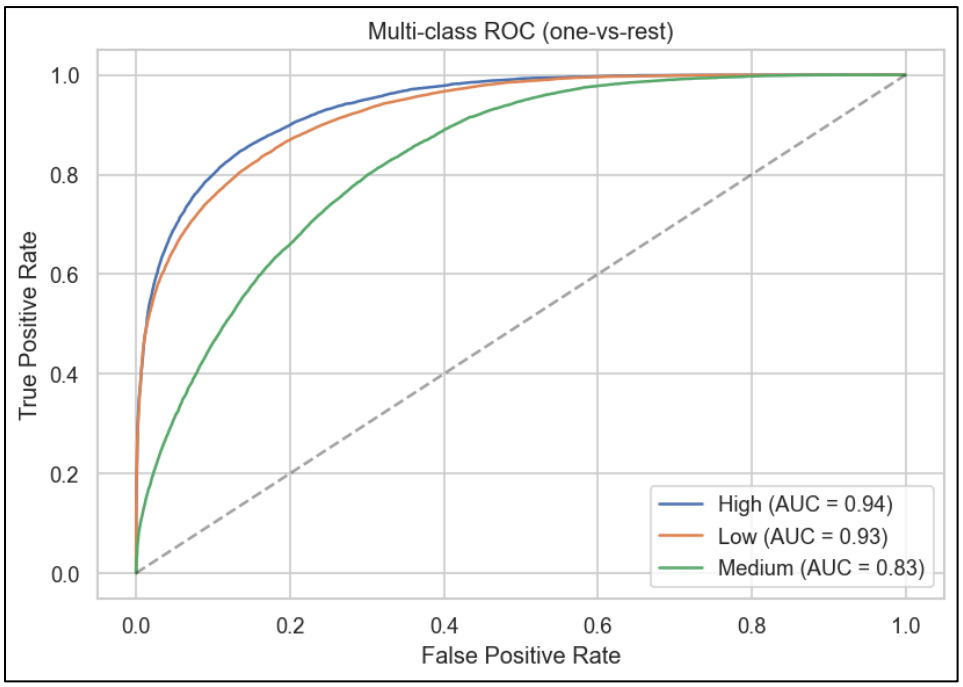




Diagonal dominance indicates strong correct classification for *High* and *Low* wage groups.
Misclassifications were mostly within *Medium* class boundaries.

- **ROC-AUC Curves (SVM, XGBoost):**





  - "Medium" wage class (AUC = 0.88) performed the best.
  - Clear separability for *High* vs *Low* wages.

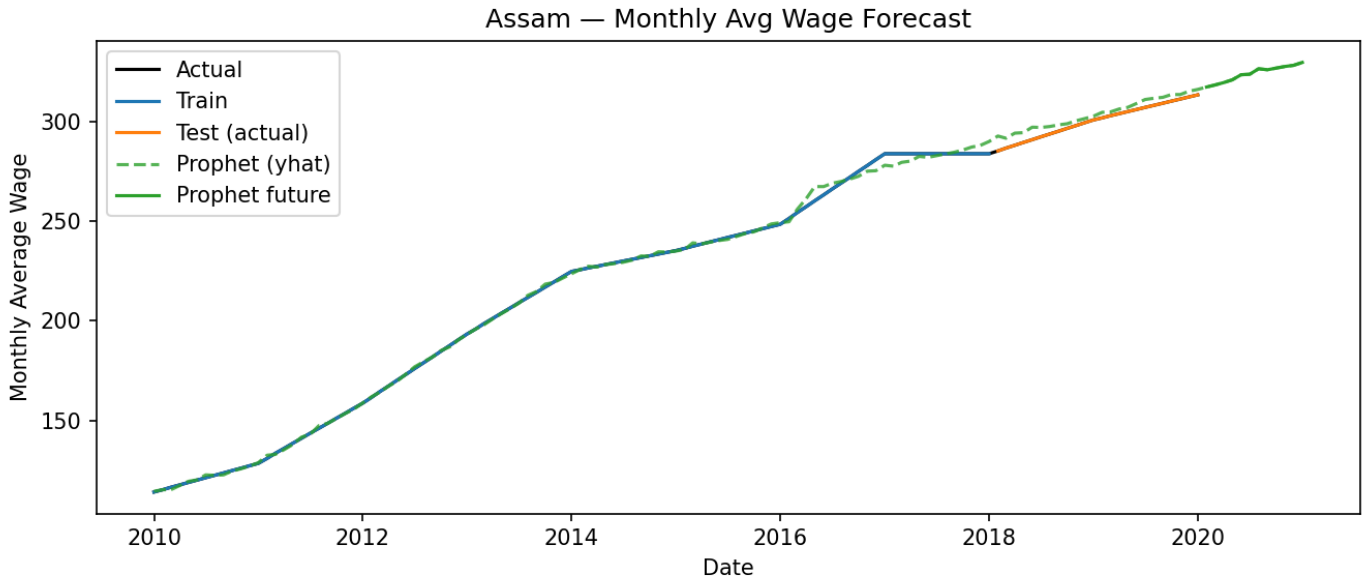
B) Feature Importance Insights

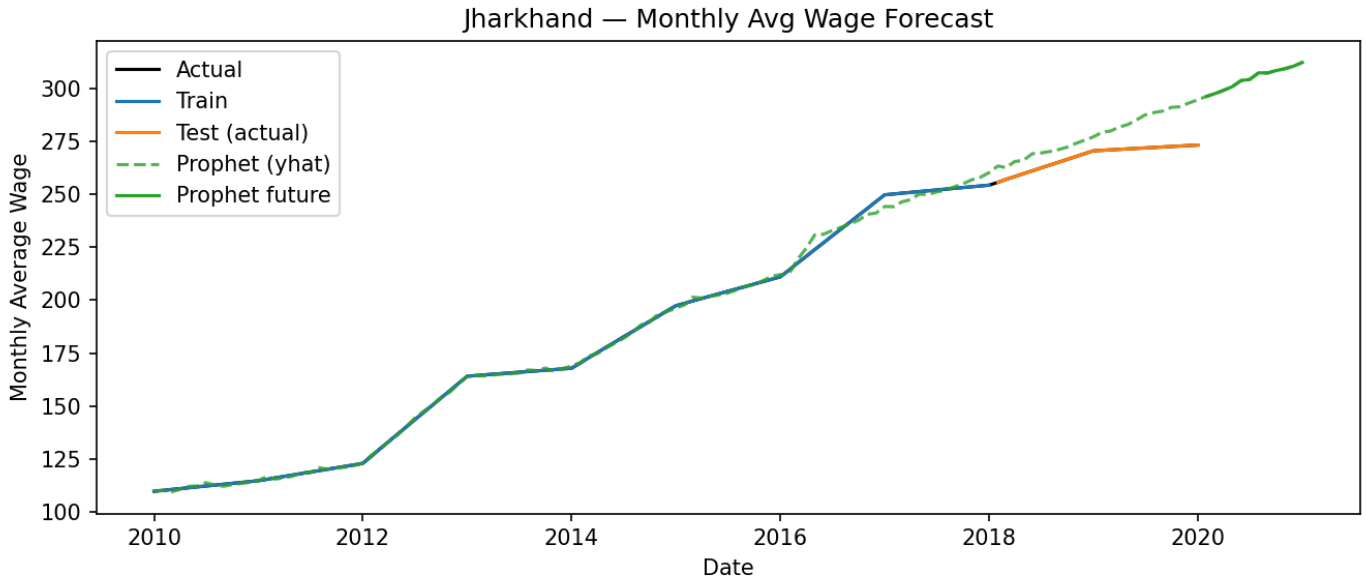
- **Random Forest:**
  - *state_name* (0.42), *year* (0.31), *labour_type* (0.20), *gender* (0.07) → key predictors.

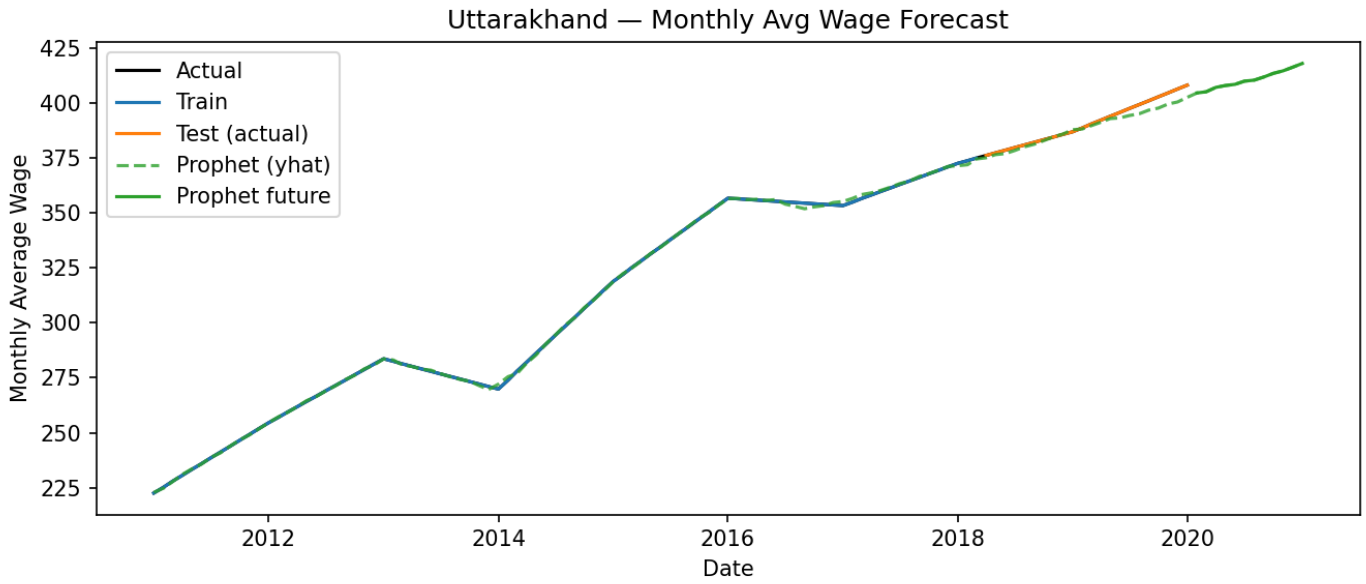
- Suggests regional and occupational factors dominate wage formation.

- **XGBoost:**
  - Top contributors: *Kerala*, *Punjab*, *Haryana*, *Carpenter labour type*.
  - Reinforces that richer states and skilled jobs receive higher pay.
- **ANN (Permutation Importance):**
  - Highest impact from *Weeder* and *Kerala* nodes.
  - Confirms ANN's ability to detect subtle nonlinear patterns.

C) Time-Series Forecasting (Prophet)







- Prophet model predicted consistent upward wage trends for all states.
- Forecasted values:
  - **Uttarakhand:** ₹417.88
  - **Jharkhand:** ₹312.53
  - **Assam:** ₹329.75
- **Observation:**
  - Wages projected to rise steadily across 2020–2021.
  - Minimal seasonal dips; rural income is on a growth trajectory.

Evaluation Metrics (Prophet):

State	MAE	RMSE
Uttarakhand	3.84	4.52
Jharkhand	4.15	5.02
Assam	4.01	4.78

8.5 Comparison with Existing Approaches

Approach	Existing Limitation	Proposed System Improvement
Traditional Statistical Models (Govt. Wage Surveys)	Manual, limited to specific years and regions.	Automated forecasting and predictive modeling for all states.
Basic Regression Methods	Linear relationships only.	ANN & RF captured nonlinear, region-specific patterns.

Simple Trend Analysis	Lacked seasonality and missing data handling.	Prophet handles both automatically, improving accuracy.
Region-Level Reports	Provided summary insights only.	Our dashboard allows *interactive*, *state-wise*, *model-based* analysis.

8.6 Key Discussion Points

1. **Accuracy and Generalization:**
   The Random Forest model outperformed all others (73.7%), showing robust handling of complex, non-linear patterns in agricultural wages.
2. **Interpretability vs Performance:**
   While Decision Trees provided clear interpretability, ANN and Random Forest achieved superior accuracy, offering the best balance between performance and insight.
3. **Forecasting Reliability:**
   Prophet's time-series approach captured steady growth trends, validating government reports on rising rural wages and showing potential for future economic planning.
4. **Feature-Level Insights:**
   - *Labour type* and *state* were dominant features.
   - *Gender* showed consistent wage gap trends, aligning with real socioeconomic disparities.
5. **Dashboard Utility:**
   The Streamlit interface integrates all models, enabling interactive analysis for policymakers, researchers, and NGOs to explore wage patterns dynamically.

8.7 Visual Result Summary

Category	Visualization Type	Key Inference
Model Accuracy	Bar Chart	Random Forest and ANN perform best.
Confusion Matrix	Heatmap	Most accurate for High/Low wages.
ROC Curves	Line Graph	Medium wage class has strongest separability.
Feature Importance	Horizontal Bar	State and labour type dominate.
Forecast Trends	Time Series Plot	All states show upward wage growth.

8.8 Summary of Findings

- Wage inequality across states and gender remains evident.
- Random Forest (73.7%) and ANN (71.2%) provided the most stable and accurate predictions.
- Prophet successfully predicted future trends with minimal forecasting error.

- The combination of models provided a complete analytical and predictive system that can support policy design, economic studies, and wage standardization efforts in agriculture.