Crop Planner

Business Case

Background

In today's agriculture landscape, farmers face numerous challenges including unpredictable weather patterns and soil variability. Data-driven solutions are crucial for addressing these challenges.

Our project builds upon existing datasets on rainfall, climate, and fertilizer data, augmenting them to create a robust foundation for predictive modeling.

Project Definition

Our goal is to develop a predictive model that recommends the most suitable crops based on soil and environmental conditions.

By analyzing parameters such as nitrogen, phosphorous, potassium levels, temperature, humidity, pH, and rainfall, farmers will receive personalized recommendations for optimal crop selection, ultimately enhancing agricultural productivity and sustainability.

Project Implementation

Our implementation plan follows a structured timeline, with milestones and responsibilities clearly defined.

1. Project Kickoff	6. Documentation
2. Data Collection and Preparation	7. Deployment
3. Model Development	8. Training and Support
4. User Interface Design	9. Evaluation and Iteration
5. Integration and Testing	10. Conclusion and Handover

Business Requirements

Accuracy, usability, and scalability are paramount, ensuring the model is both effective and accessible to farmers.

Compliance with data regulations ensures ethical data management practices, building trust and reliability within the agricultural community.

Model and Deployment

After careful consideration, we've chosen Gaussian Naive Bayes as our preferred option. This algorithm is suitable for this task as it assumes that the features follow a Gaussian (normal) distribution, which is often a reasonable assumption for many real-world datasets. It's a simple and efficient algorithm which aligns with our project goals. Utilizing Streamlit, we'll develop a user-friendly interface and deploy it on Heroku for farmers to access crop planner seamlessly.

Evaluation

We've evaluated multiple approaches for model development, considering factors such as algorithm complexity, tool availability, and deployment feasibility.

Options range from machine learning algorithms to traditional statistical methods, each with its pros and cons. Our decision prioritizes accuracy, interpretability, and scalability

Benefits and Risks

The benefits of our crop planner project are manifold. Farmers gain access to tailored recommendations, improving crop yields and profitability. Resource efficiency reduces input costs and environmental impact. Ultimately, the project enhances food security and rural livelihoods.

While the project offers significant benefits, risks must be acknowledged. Data quality issues, model accuracy, and adoption barriers pose challenges. Mitigation strategies include rigorous testing, stakeholder engagement, and ongoing monitoring to ensure project success.

Financial Analysis

- Cost Estimation: Evaluate project expenses including data collection, model development, and software infrastructure.
- Revenue Projection: Estimate revenue gains from increased crop yields facilitated by the crop planner system.
- Return on Investment (ROI) Calculation: Determine the project's ROI by comparing projected revenue with initial investment.
- Funding Options: Explore grants, partnerships, and subsidies to secure financial resources.
- Financial Risk Assessment: Identify and mitigate risks such as cost overruns and revenue uncertainty.

Resources Required

- Personnel: Assemble a team of data scientists, developers, and agricultural experts.
- Technology Infrastructure: Ensure access to cloud resources and development tools.
- Data Resources: Utilize comprehensive datasets for model training and validation.
- Training and Capacity Building: Develop programs to educate users on the crop planner system.
- Stakeholder Engagement: Foster collaboration with farmers, organizations, and government agencies.

Methodology

About Dataset

Data fields:

- N - ratio of Nitrogen content in soil
- P - ratio of Phosphorus content in soil
- K - ratio of Potassium content in soil
- temperature - temperature in degree Celsius
- humidity - relative humidity in %
- ph - ph value of the soil
- rainfall - rainfall in mm
- Target Variables

The Machine Learning Canvas (v0.4)

Designed for: Designed by: Date: Iteration:

Decisions

How are predictions used to make decisions that provide the proposed value to the end-user?

Predictions generated by the model empower farmers to make data-driven decisions about crop selection.

ML task

Input, output to predict, type of problem.

The model takes input data related to soil and environmental conditions and predicting the recommended crop type.

Value Propositions

What are we trying to do for the end-user(s) of the predictive system? What objectives are we serving?

Our objective is to boost agricultural productivity by optimizing crop selection, thereby maximizing crop yields and overall farm profitability. Concurrently, we aim to promote sustainability by encouraging the adoption of environmentally friendly and resource-efficient crop varieties, minimizing environmental impact and fostering long-term agricultural sustainability.

Data Sources

Which raw data sources can we use (internal and external)?

We sourced a comprehensive dataset from Kaggle containing information on soil composition, climate conditions, and crop types.

Collecting Data

How do we get new data to learn from (inputs and outputs)?

Continuously gather data from external sources such as meteorological stations, soil surveys, and satellite imagery providers.

Making Predictions

When do we make predictions on new inputs? How long do we have to featurize a new input and make a prediction?

Once the relevant soil composition, climate conditions, and other input parameters are collected or provided, the model can immediately generate a prediction for the most suitable crop type to grow

Offline Evaluation

Methods and metrics to evaluate the system before deployment.

Classification metrics such as precision, recall, and F1-score to assess the model's performance across multiple crop classes.

Features

Input representations extracted from raw data sources.

Building Models

When do we create/update models with new training data? How long do we have to featurize training inputs and create a model?

The creation or updating of models with new training data involves featurizing training inputs and training the model, followed by evaluation to ensure its performance meets the desired standards.

Live Evaluation and Monitoring

Methods and metrics to evaluate the system after deployment, and to quantify value creation.

For live evaluation and monitoring post-deployment on Heroku via Streamlit, we track real-time metrics like prediction accuracy, latency, and throughput, alongside error logging

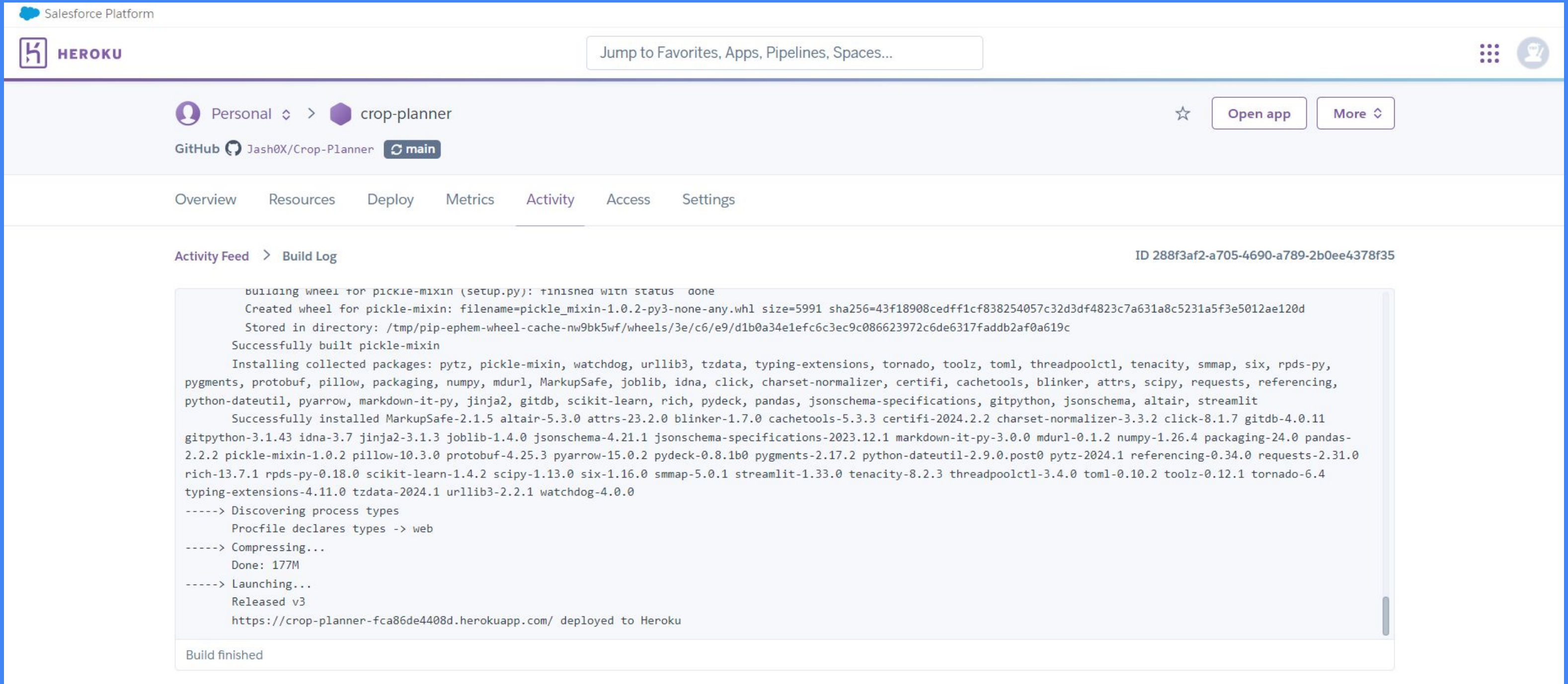
Code to train the model

```python
import pandas as pd
import numpy as np
import pickle
from sklearn.model_selection import train_test_split
from sklearn.naive_bayes import GaussianNB
from sklearn.metrics import confusion_matrix, classification_report
from sklearn.preprocessing import StandardScaler

def preprocess_data(df):
    for col in df.columns:
        try:
            df[col] = pd.to_numeric(df[col])
        except ValueError:
            pass

    df.dropna(inplace=True)

def read_in_and_split_data(data, target):
    X = data.drop(target, axis=1)
    y = data[target]
    X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=45)
    return X_train, X_test, y_train, y_test

def train_model(X_train, y_train):
    model = GaussianNB()
    model.fit(X_train, y_train)
    return model

def save_model(model, filename):
    with open(filename, 'wb') as file:
        pickle.dump(model, file)

df = pd.read_csv('Crop_recommendation.csv')

# Preprocess data
preprocess_data(df)

# Split Data to Training and Validation set
target ='label'
X_train, X_test, y_train, y_test = read_in_and_split_data(df, target)

# Train model
model = train_model(X_train, y_train)

# Save model
save_model(model, 'model.pkl')
```

Code for Streamlit

```python
def main():
    html_temp = """
    <div>
    <h1 style="color:MEDIUMSEAGREEN;text-align:left;"> Crop Planner</h1>
    </div>
    """
    st.markdown(html_temp, unsafe_allow_html=True)

    st.write("""
    ### About:
    By analyzing parameters such as nitrogen, phosphorous, potassium levels, temperature, humidity, pH, and rainfall, you w
    """)

    st.write("""
    Complete all the parameters and our model will give you the most suitable crop to grow in your farm land.
    """)

    N = st.number_input("Nitrogen", 1,10000)
    P = st.number_input("Phosphorus", 1,10000)
    K = st.number_input("Potassium", 1,10000)
    temp = st.number_input("Temperature",0.0,100000.0)
    humidity = st.number_input("Humidity in %", 0.0,100000.0)
    ph = st.number_input("pH", 0.0,100000.0)
    rainfall = st.number_input("Rainfall in mm",0.0,100000.0)

    feature_list = [N, P, K, temp, humidity, ph, rainfall]
    single_pred = np.array(feature_list).reshape(1,-1)

    if st.button('Predict'):
        loaded_model = load_model('model.pkl')
        prediction = loaded_model.predict(single_pred)
        st.write('''
        ## Recommendation:
        ''')
        st.success(f"{prediction.item().title()} is the most suitable crop to be grown based on the parameters provided.")


    hide_menu_style = """
    <style>
    #MainMenu {visibility: hidden;}
    </style>
    """

    st.markdown(hide_menu_style, unsafe_allow_html=True)

if __name__ == '__main__':
```

Heroku deployment



DEMO


Thank You