Analysis of Total Geographical Land Use and Prediction of Crops using Various ML Models

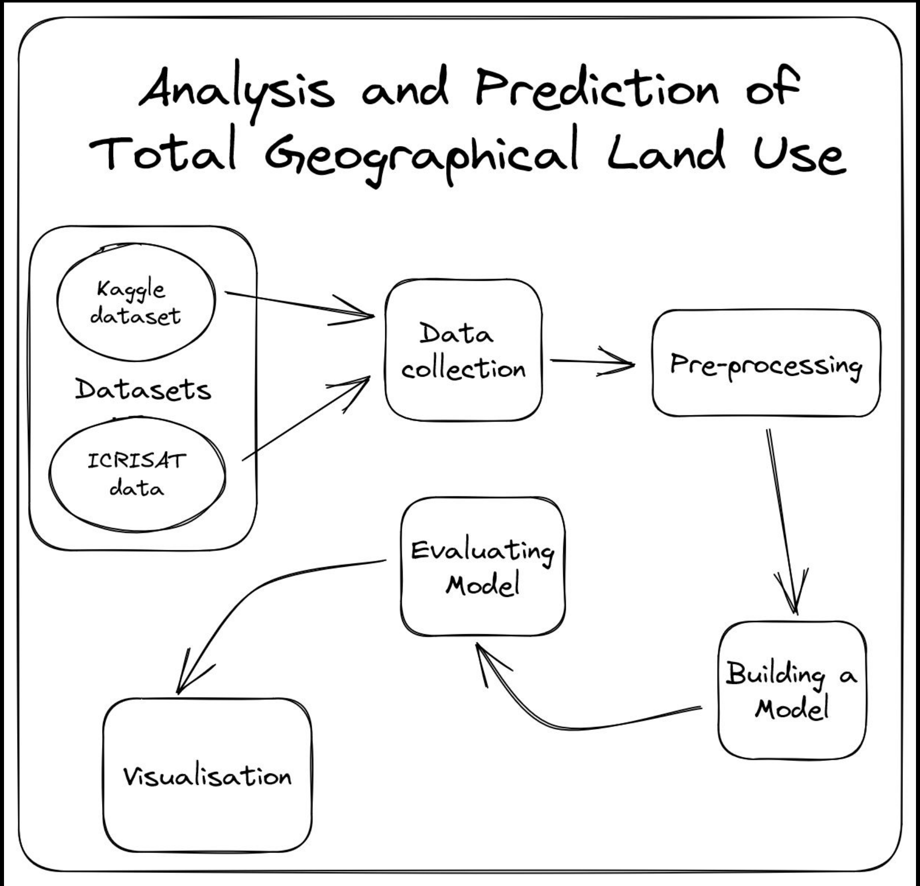
Review - 2

20BCE1043 (Vishal N)
20BCE1360 (Prathiba Narayan)
20BCE1317 (Jyothssena GS)

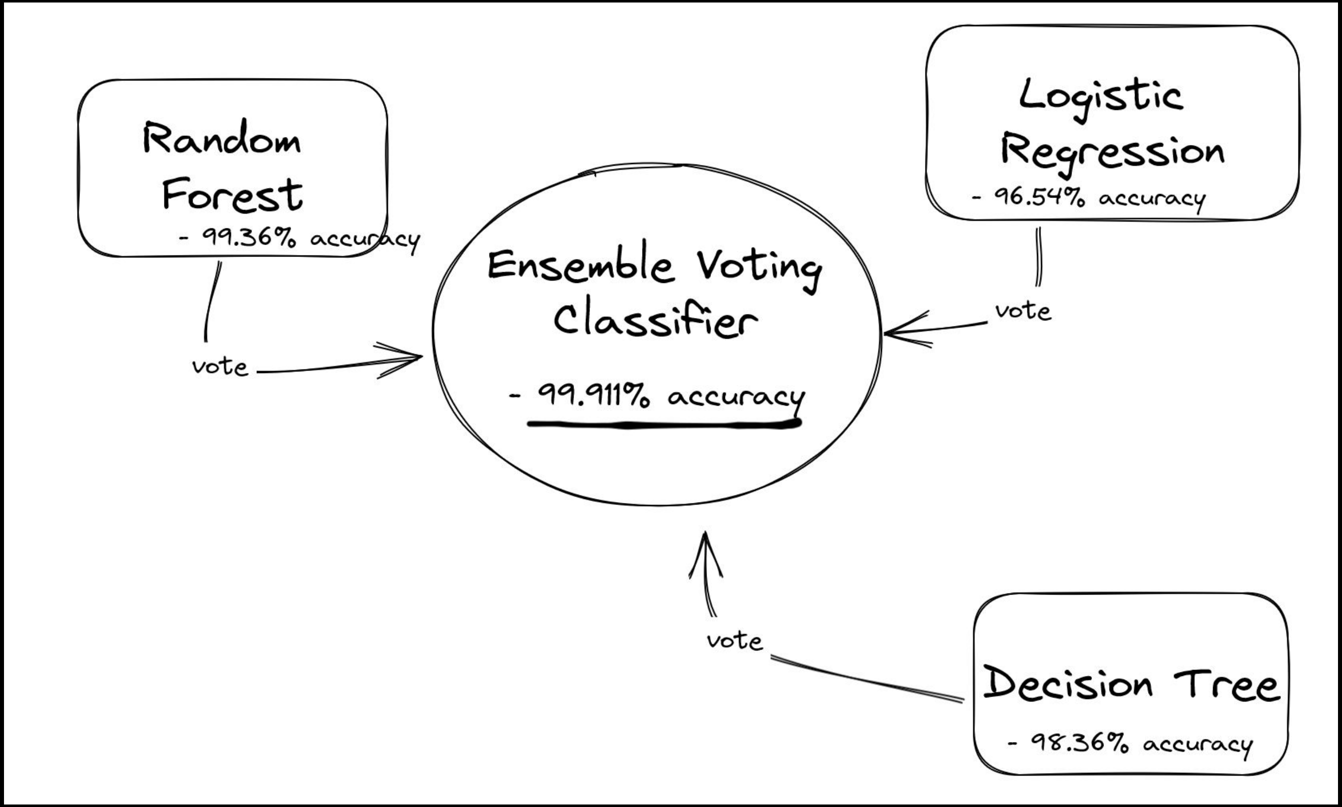
CLASSIFICATION MODELS

1. Support Vector Classifier
2. Decision Tree
3. Random Forest
4. Gradient Boosting
5. XGBoost
6. Linear Regression
7. Deep learning

ARCHITECTURE DIAGRAM



Ensemble Voting Classifier

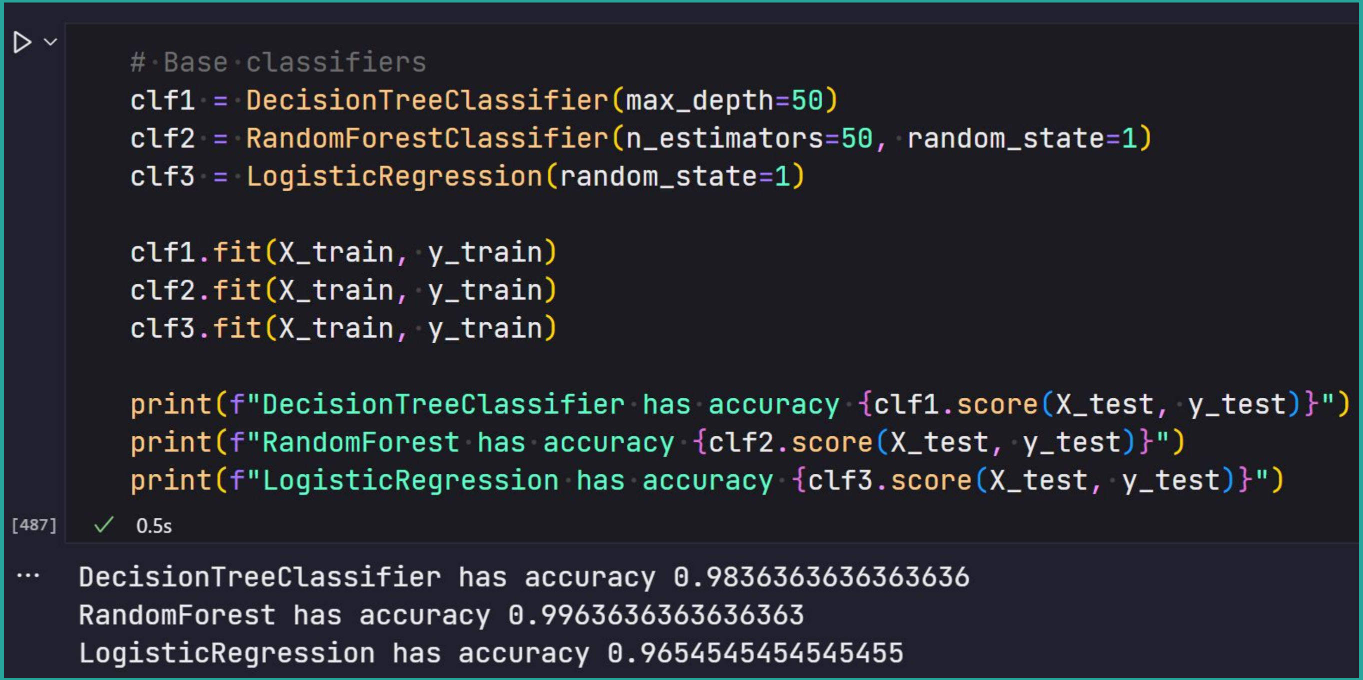


- Our model is capable of achieving an accuracy of > 99.91%.

MODEL RESULTS

MODEL	Accuracy
Decision Tree	98.36%
Random Forest	99.36%
Logistic Regression	96.54%
Support Vector Classifier	98.72%
XGBoost	99.45%

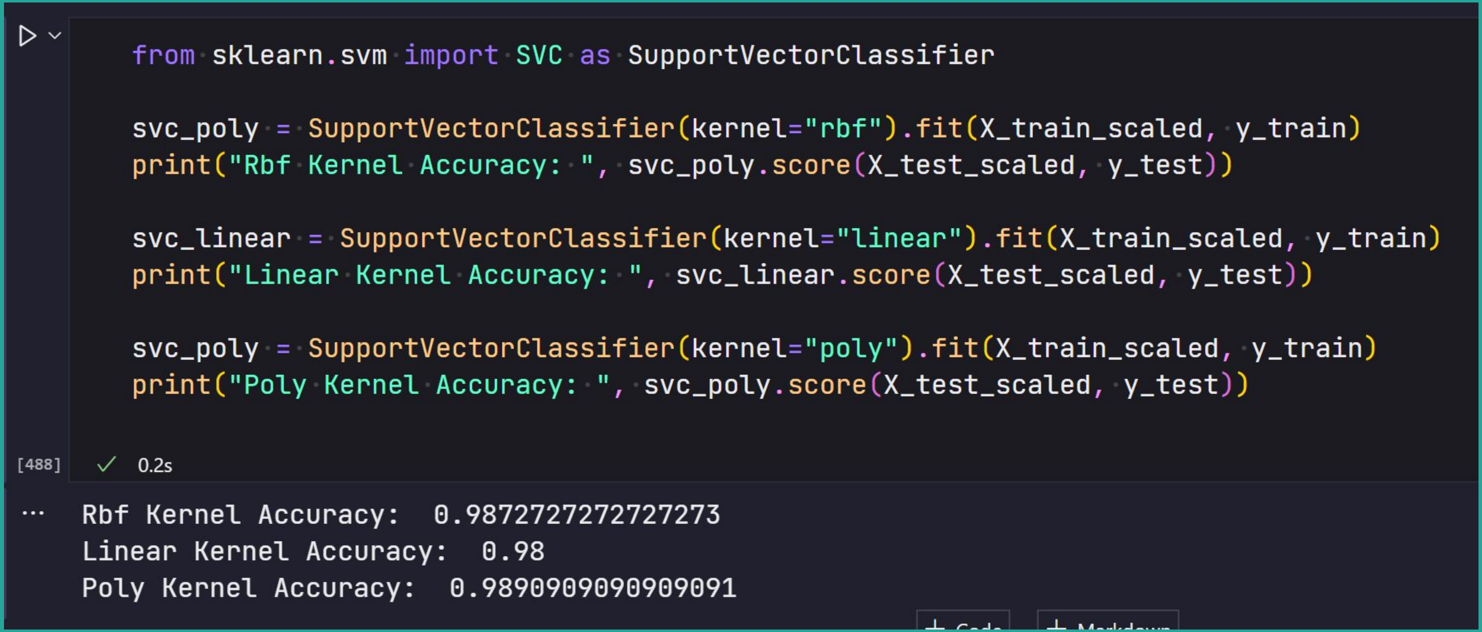
MODEL RESULTS



```python
# Base classifiers
clf1 = DecisionTreeClassifier(max_depth=50)
clf2 = RandomForestClassifier(n_estimators=50, random_state=1)
clf3 = LogisticRegression(random_state=1)

clf1.fit(X_train, y_train)
clf2.fit(X_train, y_train)
clf3.fit(X_train, y_train)

print(f"DecisionTreeClassifier has accuracy {clf1.score(X_test, y_test)}")
print(f"RandomForest has accuracy {clf2.score(X_test, y_test)}")
print(f"LogisticRegression has accuracy {clf3.score(X_test, y_test)}")
```

```
DecisionTreeClassifier has accuracy 0.9836363636363636
RandomForest has accuracy 0.9963636363636363
LogisticRegression has accuracy 0.9654545454545455
```

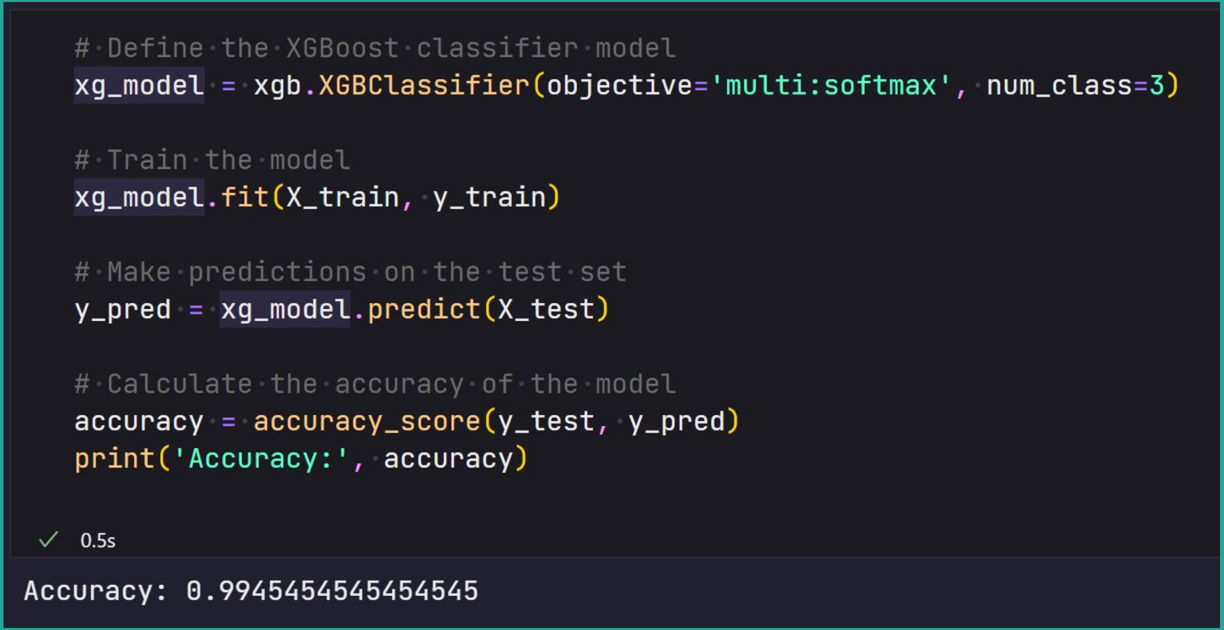
MODEL RESULTS



```
from sklearn.svm import SVC as SupportVectorClassifier

svc_poly = SupportVectorClassifier(kernel="rbf").fit(X_train_scaled, y_train)
print("Rbf Kernel Accuracy: ", svc_poly.score(X_test_scaled, y_test))

svc_linear = SupportVectorClassifier(kernel="linear").fit(X_train_scaled, y_train)
print("Linear Kernel Accuracy: ", svc_linear.score(X_test_scaled, y_test))

svc_poly = SupportVectorClassifier(kernel="poly").fit(X_train_scaled, y_train)
print("Poly Kernel Accuracy: ", svc_poly.score(X_test_scaled, y_test))
```

```
Rbf Kernel Accuracy:  0.9872727272727273
Linear Kernel Accuracy:  0.98
Poly Kernel Accuracy:  0.9890909090909091
```

MODEL RESULTS

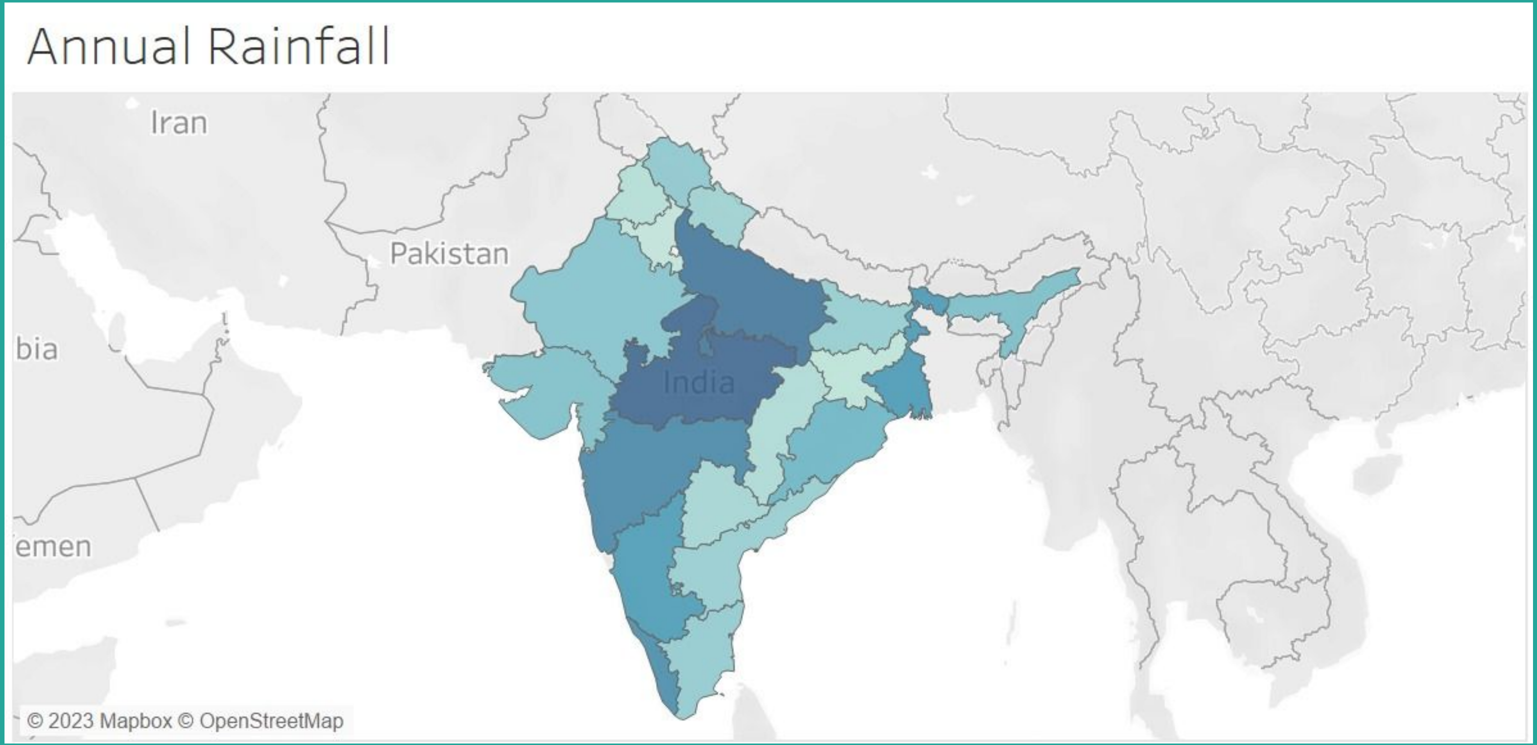


```
# Define the XGBoost classifier model
xg_model = xgb.XGBClassifier(objective='multi:softmax', num_class=3)

# Train the model
xg_model.fit(X_train, y_train)

# Make predictions on the test set
y_pred = xg_model.predict(X_test)

# Calculate the accuracy of the model
accuracy = accuracy_score(y_test, y_pred)
print('Accuracy:', accuracy)
```

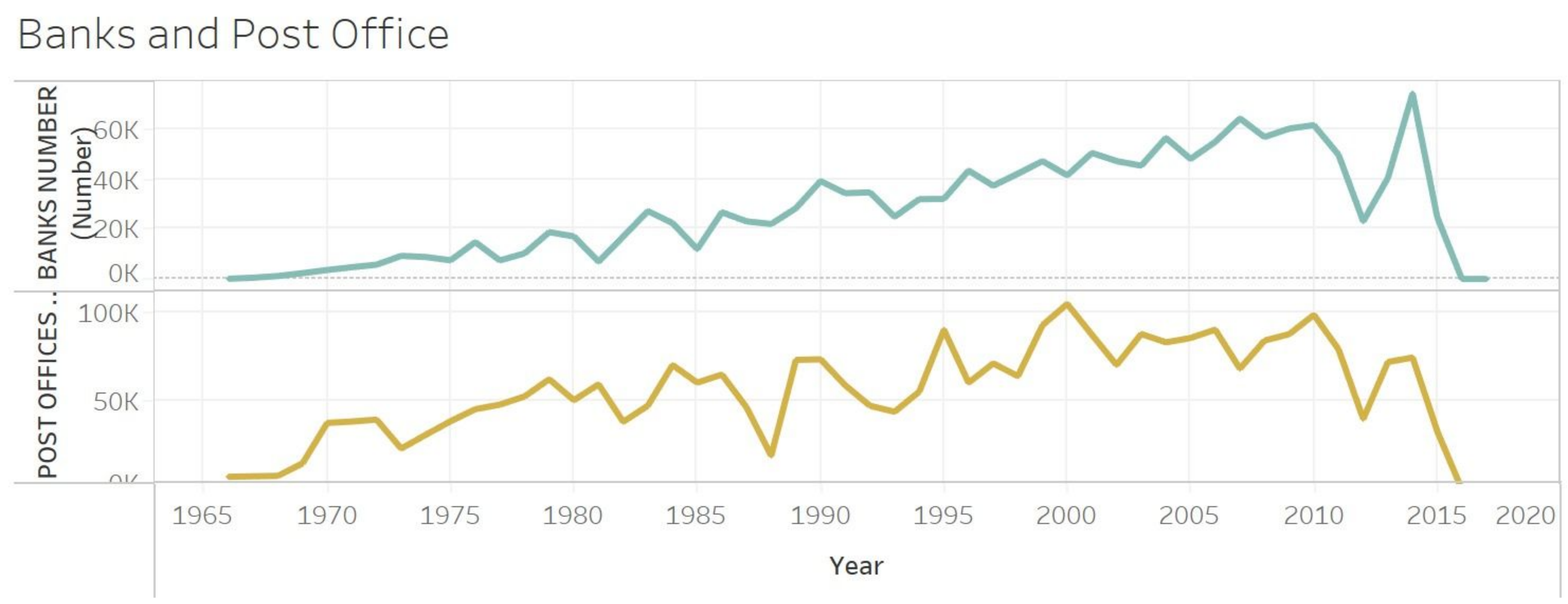
✓ 0.5s

```
Accuracy: 0.9945454545454545
```

VISUALIZATION

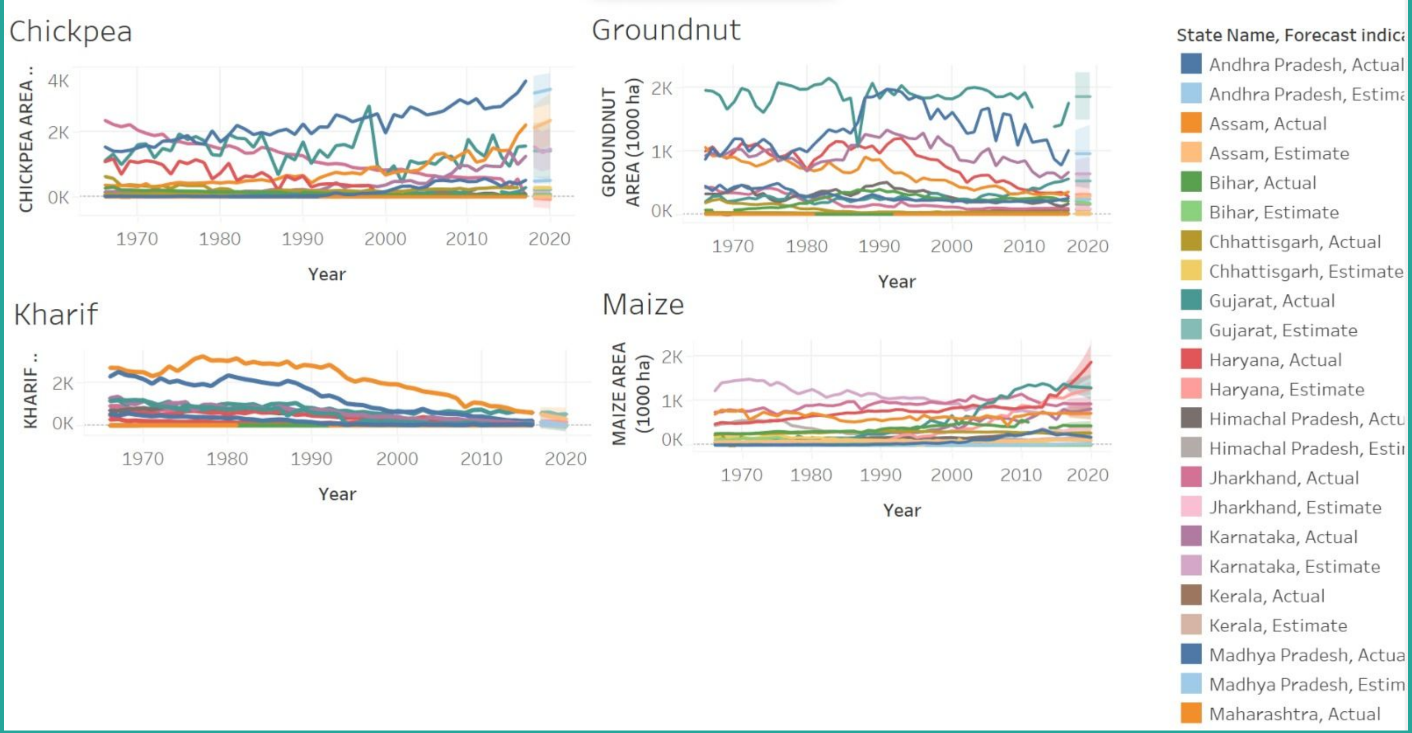


VISUALIZATION

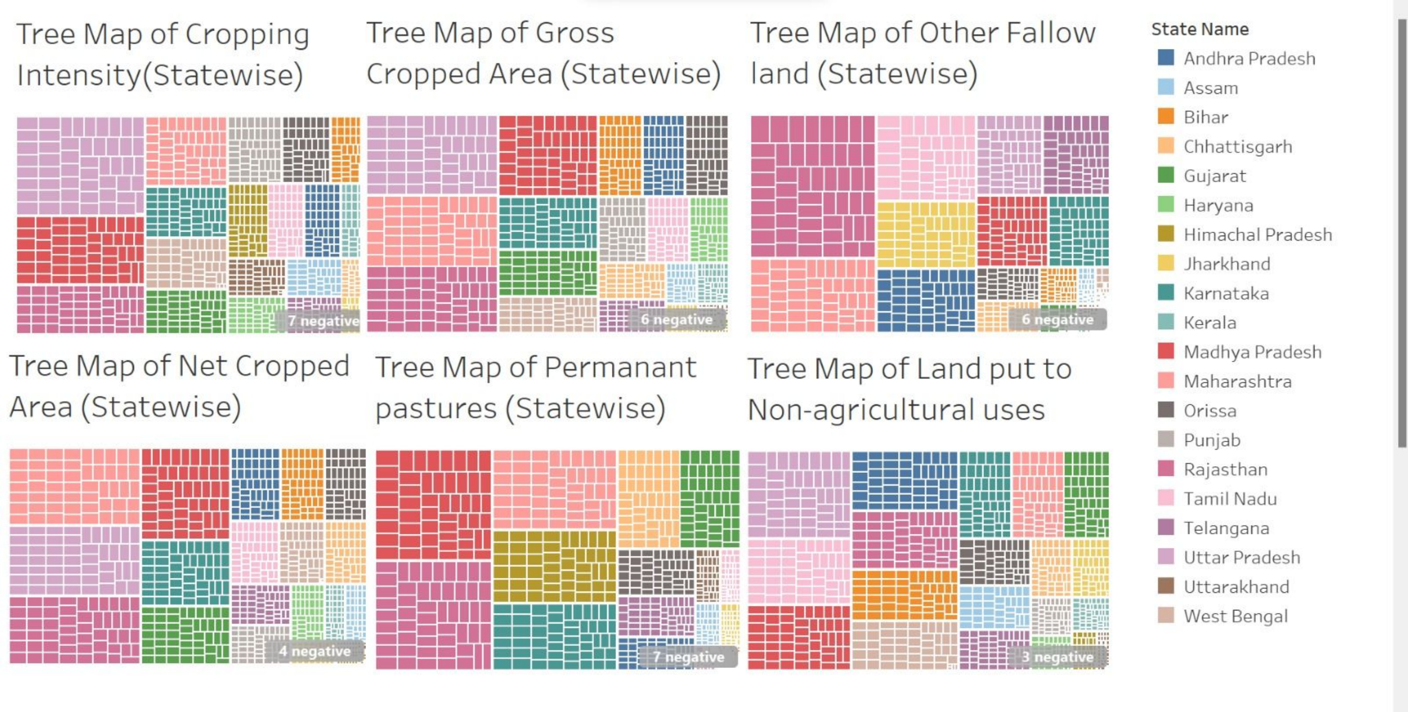


Banks and Post Office

VISUALIZATION



VISUALIZATION



Tree Map of Cropping Intensity(Statewise)

Tree Map of Gross Cropped Area (Statewise)

Tree Map of Other Fallow land (Statewise)

Tree Map of Net Cropped Area (Statewise)

Tree Map of Permanant pastures (Statewise)

Tree Map of Land put to Non-agricultural uses

Thank You