	age	workclass	fnlwgt	education	educational-num	marital-status	occupation	relationship	race	gender	capital-gain	capital-loss	hours-per-week	native-country	income
0	25	Private	226802	11th	7	Never-married	Machine-op-inspct	Own-child	Black	Male	0	0	40	United-States	<=50K
1	38	Private	89814	HS-grad	9	Married-civ-spouse	Farming-fishing	Husband	White	Male	0	0	50	United-States	<=50K
2	28	Local-gov	336951	Assoc-acdm	12	Married-civ-spouse	Protective-serv	Husband	White	Male	0	0	40	United-States	>50K
3	44	Private	160323	Some-college	10	Married-civ-spouse	Machine-op-inspct	Husband	Black	Male	7688	0	40	United-States	>50K
4	18	?	103497	Some-college	10	Never-married	?	Own-child	White	Female	0	0	30	United-States	<=50K
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
48837	27	Private	257302	Assoc-acdm	12	Married-civ-spouse	Tech-support	Wife	White	Female	0	0	38	United-States	<=50K
48838	40	Private	154374	HS-grad	9	Married-civ-spouse	Machine-op-inspct	Husband	White	Male	0	0	40	United-States	>50K
48839	58	Private	151910	HS-grad	9	Widowed	Adm-clerical	Unmarried	White	Female	0	0	40	United-States	<=50K
48840	22	Private	201490	HS-grad	9	Never-married	Adm-clerical	Own-child	White	Male	0	0	20	United-States	<=50K
48841	52	Self-emp-inc	287927	HS-grad	9	Married-civ-spouse	Exec-managerial	Wife	White	Female	15024	0	40	United-States	>50K

48842 rows × 15 columns

```
[ ] data.isnull().sum()  # it retrive all the count the null value is available
```

	0
age	0
workclass	0
fnlwgt	0
education	0
educational-num	0
marital-status	0
occupation	0
relationship	0
race	0
gender	0
capital-gain	0
capital-loss	0
hours-per-week	0
native-country	0
income	0

dtype: int64

```
data
```

	age	workclass	fnlwgt	educational-num	marital-status	occupation	relationship	race	gender	capital-gain	capital-loss	hours-per-week	native-country	income
0	25	2	226802	7	4	5	3	2	1	0	0	40	37	<=50K
1	38	2	89814	9	2	3	0	4	1	0	0	50	37	<=50K
2	28	1	336951	12	2	10	0	4	1	0	0	40	37	>50K
3	44	2	160323	10	2	5	0	2	1	7688	0	40	37	>50K
4	18	6	103497	10	4	6	3	4	0	0	0	30	37	<=50K
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
48837	27	2	257302	12	2	12	5	4	0	0	0	38	37	<=50K
48838	40	2	154374	9	2	5	0	4	1	0	0	40	37	>50K
48839	58	2	151910	9	6	0	4	4	0	0	0	40	37	<=50K
48840	22	2	201490	9	4	0	3	4	1	0	0	20	37	<=50K
48841	52	3	287927	9	2	2	5	4	0	15024	0	40	37	>50K

47603 rows × 14 columns

```
[ ] X = data.drop(columns =["income"]) # input
    y = data["income"]  # outut
    X
```

	age	workclass	fnlwgt	educational-num	marital-status	occupation	relationship	race	gender	capital-gain	capital-loss	hours-per-week	native-country
0	25	2	226802	7	4	5	3	2	1	0	0	40	37
1	38	2	89814	9	2	3	0	4	1	0	0	50	37
2	28	1	336951	12	2	10	0	4	1	0	0	40	37
3	44	2	160323	10	2	5	0	2	1	7688	0	40	37
4	18	6	103497	10	4	6	3	4	0	0	0	30	37
...	...	...	...	...	...	...	...	...	...	...	...	...	...
48837	27	2	257302	12	2	12	5	4	0	0	0	38	37
48838	40	2	154374	9	2	5	0	4	1	0	0	40	37
48839	58	2	151910	9	6	0	4	4	0	0	0	40	37
48840	22	2	201490	9	4	0	3	4	1	0	0	20	37
48841	52	3	287927	9	2	2	5	4	0	15024	0	40	37

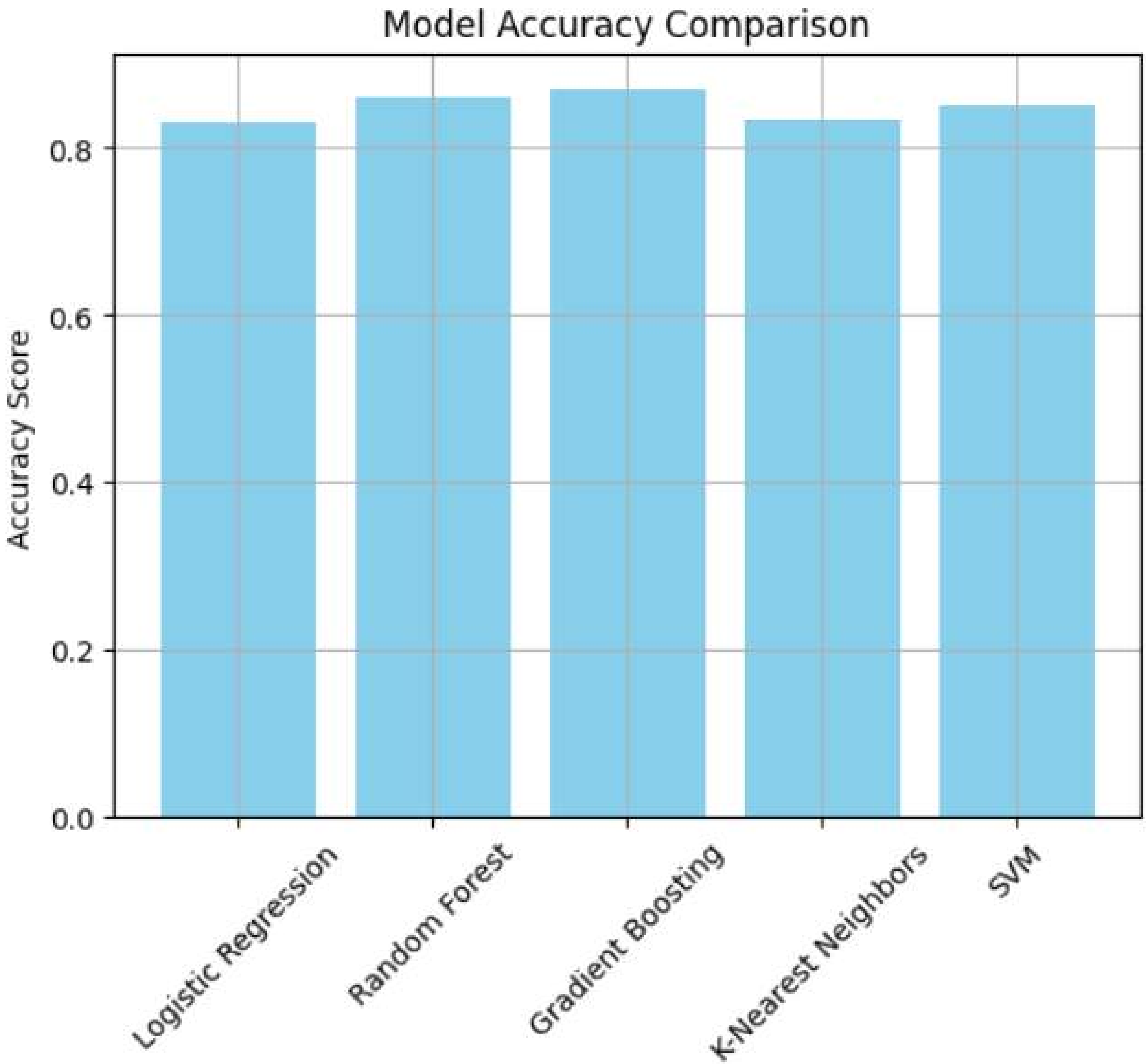
47603 rows × 13 columns

```
Logistic Regression Accuracy :0.830795
              precision    recall  f1-score   support

       <=50K       0.85      0.95      0.89      7236
        >50K       0.73      0.46      0.57      2285

    accuracy                           0.83      9521
   macro avg       0.79      0.71      0.73      9521
weighted avg       0.82      0.83      0.82      9521

Random Forest Accuracy :0.860939
              precision    recall  f1-score   support

       <=50K       0.89      0.93      0.91      7236
        >50K       0.75      0.64      0.69      2285

    accuracy                           0.86      9521
   macro avg       0.82      0.78      0.80      9521
weighted avg       0.86      0.86      0.86      9521

Gradient Boosting Accuracy :0.869446
              precision    recall  f1-score   support

       <=50K       0.89      0.95      0.92      7236
        >50K       0.79      0.62      0.69      2285

    accuracy                           0.87      9521
   macro avg       0.84      0.78      0.81      9521
weighted avg       0.86      0.87      0.86      9521

K-Nearest Neighbors Accuracy :0.832581
```

	precision	recall	f1-score	support
<=50K	0.88	0.91	0.89	7236
>50K	0.67	0.60	0.63	2285
accuracy			0.83	9521
macro avg	0.77	0.75	0.76	9521
weighted avg	0.83	0.83	0.83	9521

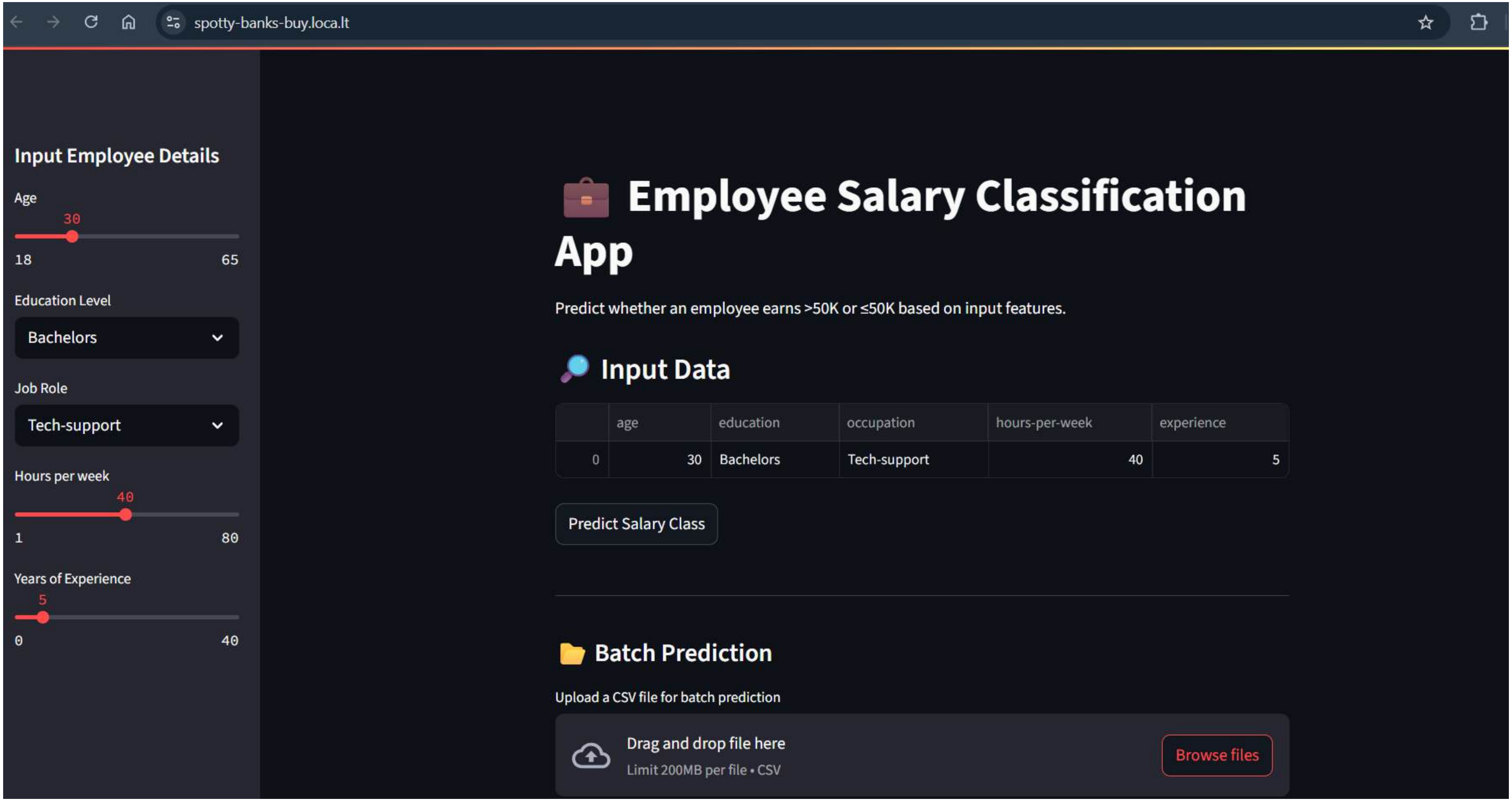
SVM Accuracy :0.851381

	precision	recall	f1-score	support
<=50K	0.87	0.95	0.91	7236
>50K	0.76	0.55	0.64	2285
accuracy			0.85	9521
macro avg	0.82	0.75	0.77	9521
weighted avg	0.84	0.85	0.84	9521


Model Accuracy Comparison
Accuracy Score
0.0
0.2
0.4
0.6
0.8
Logistic Regression
Random Forest
Gradient Boosting
K-Nearest Neighbors
SVM

```
LogisticRegression: 0.8308
RandomForest: 0.8599
KNN: 0.8326
SVM: 0.8514
GradientBoosting: 0.8694

✅ Best model: GradientBoosting with accuracy 0.8694
✅ Saved best model as best_model.pkl
```

```
warnings.warn(
──────────────────── Traceback (most recent call last) ────────────────────
/usr/local/lib/python3.11/dist-packages/streamlit/runtime/scriptrunner/exec_code.py:
128 in exec_func_with_error_handling

/usr/local/lib/python3.11/dist-packages/streamlit/runtime/scriptrunner/script_runner
.py:669 in code_to_exec

/content/app.py:55 in <module>

  52 if uploaded_file is not None:
  53 │   batch_data = pd.read_csv(uploaded_file)
  54 │   st.write("Uploaded data preview:", batch_data.head())
❱ 55 │   batch_preds = model.predict(batch_data)
  56 │   batch_data['PredictedClass'] = batch_preds
  57 │   st.write("✅ Predictions:")
  58 │   st.write(batch_data.head())

/usr/local/lib/python3.11/dist-packages/sklearn/ensemble/_gb.py:1623 in predict

  1620 │   │   y : ndarray of shape (n_samples,)
  1621 │   │   │   The predicted values.
  1622 │   │   """
❱ 1623 │   │   raw_predictions = self.decision_function(X)
  1624 │   │   if raw_predictions.ndim == 1:  # decision_function already squeezed i
  1625 │   │   │   encoded_classes = (raw_predictions >= 0).astype(int)
  1626 │   │   else:

/usr/local/lib/python3.11/dist-packages/sklearn/ensemble/_gb.py:1576 in
decision_function

  1573 │   │   │   :term:`classes_`. Regression and binary classification produce an
  1574 │   │   │   array of shape (n_samples,).
  1575 │   │   """
```


spotty-banks-buy.loca.lt
Input Employee Details
Age
30
18
65
Education Level
Bachelors
Job Role
Tech-support
Hours per week
40
1
80
Years of Experience
5
0
40
Employee Salary Classification App
Predict whether an employee earns >50K or ≤50K based on input features.
Input Data
age
education
occupation
hours-per-week
experience
0
30
Bachelors
Tech-support
40
5
Predict Salary Class
Batch Prediction
Upload a CSV file for batch prediction
Drag and drop file here
Limit 200MB per file • CSV
Browse files