1. Data Gathering and Preprocessing Handling Missing Values Identify columns with missing values and decide on a strategy to handle them (imputation or removal). Columns with missing values: avgTemp, Cloud Cover, maxTemp, Precipitation, vapPressure, Wet Day Freq, minTemp, Production. For columns with few missing values, consider imputation (e.g., mean, median, or using machine learning models). For columns with significant missing values (minTemp and Production), consider more sophisticated methods or dropping them if necessary.

Encoding Categorical Data Convert categorical variables (State, City, Season, Crop) into numerical values using techniques like one-hot encoding or label encoding. Feature Scaling Normalize or standardize numerical features to ensure they are on a similar scale, which is particularly important for algorithms like LSTM.

2. Feature Engineering Create new features that could be relevant for weather forecasting, such as lagged values, rolling averages, seasonal trends, etc. Evaluate the importance of existing features and potentially remove less important ones to simplify the model.

3. Model Selection and Training Selecting Models Choose models suitable for time series forecasting and regression tasks, such as: Random Forest: Good for initial predictions and understanding feature importance. LSTM (Long Short-Term Memory): Suitable for capturing temporal dependencies in the data. Training Models Split the dataset into training and testing sets, ensuring the split respects the temporal order. Train chosen models on the training set, tuning them to find the best hyperparameters.

4. Model Evaluation Evaluation Metrics Use metrics such as Mean Absolute Error (MAE), Mean Squared Error (MSE), Root Mean Squared Error (RMSE), and R-squared to evaluate model performance.

5. Hyperparameter Tuning Perform hyperparameter tuning using techniques like Grid Search or Random Search to optimize model performance. yperparameter tuning in machine learning is the process of finding the optimal set of hyperparameters for a particular machine learning model. Here's a breakdown of the key concepts:

Hyperparameters: These are settings that control the learning process of a model. They are set before training the model and are not learned from the data itself. Examples include the number of trees in a Random Forest, the learning rate in an Artificial Neural Network, or the kernel size in a Support Vector Machine.

6. Documentation and Iteration Document each step of the process, including data preprocessing methods, feature engineering techniques, model parameters, and evaluation results. Iterate on the model based on feedback and evaluation results to improve accuracy and efficiency.

[ ]:

[16]:
```
import pandas as pd
data=pd.read_csv("D:\\intership\\Ignite Intern\\task 1 weather\\AHMEDABAD_WEATHER_DATASET.csv" )
data
```

[16]:

	State	City	Year	Season	Crop	Area	avgTemp	Cloud Cover	maxTemp	Precipitation	vapPressure	Rainfall	Wet Day Freq	minTemp	Production
0	Gujarat	Ahmedabad	1997	Kharif	Arhar/Tur	2900	31.239400	52.422800	37.931	176.3270	35.265400	540.90	12.68950	24.0370	2200.0
1	Gujarat	Ahmedabad	1997	Kharif	Bajra	41700	31.239400	52.422800	37.931	176.3270	35.265400	540.90	12.68950	24.0370	43700.0
2	Gujarat	Ahmedabad	1997	Kharif	Dry chillies	700	31.239400	52.422800	37.931	176.3270	35.265400	540.90	12.68950	24.0370	700.0
3	Gujarat	Ahmedabad	1997	Kharif	Groundnut	500	31.239400	52.422800	37.931	176.3270	35.265400	540.90	12.68950	24.0370	600.0
4	Gujarat	Ahmedabad	1997	Kharif	Jowar	42500	31.239400	52.422800	37.931	176.3270	35.265400	540.90	12.68950	24.0370	33500.0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
8431	Gujarat	Valsad	2012	Rabi	Gram	2000	21.120500	11.709750	29.562	0.9390	13.781375	16.48	0.57425	11.5775	2000.0
8432	Gujarat	Valsad	2012	Rabi	Other Rabi pulses	3300	21.120500	11.709750	29.562	0.9390	13.781375	16.48	0.57425	11.5775	2600.0
8433	Gujarat	Valsad	2012	Summer	Rice	600	26.240333	23.931333	33.135	3.2020	19.369500	355.32	2.45175	17.0880	1700.0
8434	Gujarat	Valsad	2012	Whole Year	Dry chillies	100	23.892250	31.071542	33.135	586.6035	18.901000	664.42	38.79010	11.5775	100.0
8435	Gujarat	Valsad	2012	Whole Year	Sugarcane	8600	23.892250	31.071542	33.135	586.6035	18.901000	664.42	38.79010	11.5775	65500.0

8436 rows × 15 columns

[ ]:

[17]:
```
data.shape
```

[17]:
```
(8436, 15)
```

[18]:
```
data.isna()
```

[18]:

	State	City	Year	Season	Crop	Area	avgTemp	Cloud Cover	maxTemp	Precipitation	vapPressure	Rainfall	Wet Day Freq	minTemp	Production
0	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False
1	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False
2	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False
3	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False
4	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
8431	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False
8432	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False
8433	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False
8434	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False
8435	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False

8436 rows × 15 columns

[ ]:

[19]:
```
data.describe()
```

[19]:

	Year	Area	avgTemp	Cloud Cover	maxTemp	Precipitation	vapPressure	Rainfall	Wet Day Freq	minTemp	Production
count	8436.000000	8436.000000	8434.000000	8429.000000	8429.000000	8429.000000	8429.000000	8436.000000	8429.000000	8208.000000	8.367000e+03
mean	2005.159673	18367.001422	26.456331	35.305001	35.081779	518.041159	23.634287	444.719386	21.776194	17.270197	6.266184e+04
std	4.586311	48911.276510	3.333164	16.718134	4.148326	430.227159	15.944184	392.677592	16.031058	4.490583	3.927533e+05
min	1997.000000	100.000000	17.334000	6.982500	25.607000	0.000000	0.071250	5.120000	0.000000	8.014000	0.000000e+00
25%	2001.000000	500.000000	23.211500	19.615667	32.515000	16.519000	18.150333	120.320000	2.130400	12.753000	6.000000e+02
50%	2005.000000	2300.000000	27.523333	35.725800	35.464000	558.413000	22.926333	410.300000	23.902800	18.198000	3.100000e+03
75%	2009.000000	11600.000000	29.254000	51.033800	38.414000	824.232000	26.272400	673.700000	33.869400	21.459000	2.120000e+04
max	2012.000000	499100.000000	32.023600	64.197400	42.552000	2260.642000	199.472600	1870.260000	65.977200	24.748000	1.175400e+07

[20]:
```
data.isna()
```

[20]:

	State	City	Year	Season	Crop	Area	avgTemp	Cloud Cover	maxTemp	Precipitation	vapPressure	Rainfall	Wet Day Freq	minTemp	Production
0	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False
1	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False
2	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False
3	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False
4	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
8431	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False
8432	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False
8433	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False
8434	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False
8435	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False

8436 rows × 15 columns

[21]:
```
data.isna().sum()
```

[21]:
```
State              0
City               0
Year               0
Season             0
Crop               0
Area               0
avgTemp            2
Cloud Cover        7
maxTemp            7
Precipitation      7
vapPressure        7
Rainfall           0
Wet Day Freq       7
minTemp          228
Production        69
dtype: int64
```

[22]:
```
data.dtypes
```

[22]:
```
State             object
City              object
Year               int64
Season            object
Crop              object
Area               int64
avgTemp          float64
Cloud Cover      float64
maxTemp          float64
Precipitation    float64
vapPressure      float64
Rainfall         float64
Wet Day Freq     float64
minTemp          float64
Production       float64
dtype: object
```

[23]:
```
data.index
```

[23]:
```
RangeIndex(start=0, stop=8436, step=1)
```

[24]:
```
data.index
```

[24]:
```
RangeIndex(start=0, stop=8436, step=1)
```

[25]:
```
data.dtypes
```

[25]:
```
State             object
City              object
Year               int64
Season            object
Crop              object
Area               int64
avgTemp          float64
Cloud Cover      float64
maxTemp          float64
Precipitation    float64
vapPressure      float64
Rainfall         float64
Wet Day Freq     float64
minTemp          float64
Production       float64
dtype: object
```

[26]:
```
missing_minTemp = data[data["minTemp"].isna()]
missing_minTemp
```

[26]:

	State	City	Year	Season	Crop	Area	avgTemp	Cloud Cover	maxTemp	Precipitation	vapPressure	Rainfall	Wet Day Freq	minTemp	Production
2903	Gujarat	Jamnagar	1997	Kharif	Arhar/Tur	6300	30.6548	52.0732	NaN	NaN	NaN	350.49	NaN	NaN	4700.0

3272	Gujarat	Junagadh	1997	Kharif	Arhar/Tur	3100	27.5992	49.0525	NaN	NaN	NaN	650.49	NaN	NaN	2300.0
3637	Gujarat	Kutch	1997	Kharif	Arhar/Tur	300	25.9744	NaN	NaN	NaN	NaN	350.49	NaN	NaN	200.0
5029	Gujarat	Navsari	2003	Kharif	Arhar/Tur	2900	22.2138	NaN	NaN	NaN	NaN	1229.23	NaN	NaN	2500.0
5920	Gujarat	Porbandar	2003	Kharif	Arhar/Tur	100	26.8060	NaN	NaN	NaN	NaN	529.23	NaN	NaN	100.0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
8267	Gujarat	Valsad	2003	Kharif	Moong(Green Gram)	300	24.8174	51.8866	30.67	883.406	22.9771	1629.23	44.1931	NaN	200.0
8268	Gujarat	Valsad	2003	Kharif	Ragi	9600	24.8174	51.8866	30.67	883.406	22.9771	1629.23	44.1931	NaN	10200.0
8269	Gujarat	Valsad	2003	Kharif	Rice	42700	24.8174	51.8866	30.67	883.406	22.9771	1629.23	44.1931	NaN	93300.0
8270	Gujarat	Valsad	2003	Kharif	Small millets	400	24.8174	51.8866	30.67	883.406	22.9771	1629.23	44.1931	NaN	100.0
8271	Gujarat	Valsad	2003	Kharif	Urad	4500	24.8174	51.8866	30.67	883.406	22.9771	1629.23	44.1931	NaN	3000.0

228 rows × 15 columns

[29]:
```
from sklearn.preprocessing import OneHotEncoder
# Encoding categorical data
categorical_features = ['State', 'City', 'Season', 'Crop']
categorical_transformer = OneHotEncoder(handle_unknown='ignore')
categorical_transformer
```

[29]:
```
OneHotEncoder
OneHotEncoder(handle_unknown='ignore')
```

[28]:
```
# Encoding categorical data
from sklearn.preprocessing import StandardScaler
numerical_features = ['Area', 'avgTemp', 'Cloud Cover', 'maxTemp', 'Precipitation', 'vapPressure', 'Rainfall', 'Wet Day Freq', 'minTemp']
numerical_transformer = StandardScaler()
numerical_transformer
```

[28]:
```
StandardScaler
StandardScaler()
```

[30]:
```
from sklearn.compose import ColumnTransformer
# Preprocessor
preprocessor = ColumnTransformer(transformers=[('num', numerical_transformer, numerical_features),('cat', categorical_transformer, categorical_features)])
preprocessor
```

[30]:
```
ColumnTransformer
num
StandardScaler
cat
OneHotEncoder
```

[33]:
```
from sklearn.ensemble import RandomForestRegressor
# Define the model
model = RandomForestRegressor()
model
```

[33]:
```
RandomForestRegressor
RandomForestRegressor()
```

[34]:
```
from sklearn.pipeline import Pipeline
from sklearn.model_selection import train_test_split
# Create and evaluate pipeline
pipeline = Pipeline(steps=[('preprocessor', preprocessor), ('model', model)])
X = data.drop('Production', axis=1)
y = data['Production']

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

[35]:
```
X_train
```

[35]:

	State	City	Year	Season	Crop	Area	avgTemp	Cloud Cover	maxTemp	Precipitation	vapPressure	Rainfall	Wet Day Freq	minTemp
4635	Gujarat	Mahesana	2005	Whole Year	Potato	2700	27.649833	28.365333	41.680	352.844	19.500667	245.02	23.0166	10.523
1794	Gujarat	Bhavnagar	1997	Whole Year	Sesamum	53400	27.573417	32.342500	36.248	293.325	24.047917	195.91	21.1150	13.902
3092	Gujarat	Jamnagar	2005	Kharif	Jowar	1100	29.450200	51.033800	36.047	322.856	29.881600	470.46	20.6786	23.014
3649	Gujarat	Kutch	1997	Whole Year	Cotton(lint)	50200	23.048667	23.057333	31.858	48.490	19.105833	135.21	7.5121	8.998
6499	Gujarat	Sabarkantha	1999	Whole Year	Dry chillies	200	27.224250	29.711167	41.564	679.492	18.256583	268.69	27.7155	10.673
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
5734	Gujarat	Patan	2003	Rabi	Wheat	22400	22.809750	10.098750	34.247	24.610	12.089500	11.15	1.7913	11.608
5191	Gujarat	Navsari	2010	Whole Year	Sugarcane	22000	21.126500	24.112417	29.618	934.914	16.823000	617.60	36.7637	11.215
5390	Gujarat	Panchmahal	2002	Kharif	Sesamum	2500	29.598400	53.193400	38.841	719.213	26.004000	800.22	32.8816	21.207
860	Gujarat	Anand	2009	Kharif	Other Kharif pulses	100	29.145400	54.180000	36.039	564.778	26.234400	706.23	26.7607	21.201
7270	Gujarat	Surendranagar	1998	Whole Year	Dry chillies	300	27.327750	30.146417	40.068	637.197	21.862500	178.35	27.4138	11.535

6748 rows × 14 columns

[36]:
```
X_test
```

[36]:

	State	City	Year	Season	Crop	Area	avgTemp	Cloud Cover	maxTemp	Precipitation	vapPressure	Rainfall	Wet Day Freq	minTemp

6377	Gujarat	Rajkot	2010	Summer	Groundnut	15000	28.364000	18.632333	36.016	9.544	24.412333	104.22	1.0817	18.614
8100	Gujarat	Vadodara	2011	Summer	Moong(Green Gram)	1000	30.147333	21.071000	38.611	18.409	19.265667	129.91	1.2751	19.230
8045	Gujarat	Vadodara	2010	Kharif	Groundnut	800	29.365000	51.604600	36.669	742.545	25.466800	800.48	32.5647	23.017
1127	Gujarat	Banas Kantha	2003	Kharif	Sesamum	21000	30.812800	44.594200	39.783	466.358	27.599000	570.41	16.3942	21.907
7249	Gujarat	Surendranagar	1997	Whole Year	Onion	200	28.073250	29.901833	39.238	141.746	22.596333	177.84	13.5464	11.185
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
668	Gujarat	Amreli	2010	Kharif	Tobacco	1200	23.062600	43.400200	28.386	550.136	24.299200	460.26	22.9239	18.745
1832	Gujarat	Bhavnagar	1999	Rabi	Rapeseed &Mustard	700	22.964000	11.977250	33.801	2.527	15.756750	12.30	1.3393	12.611
2181	Gujarat	Dangs	2001	Kharif	Arhar/Tur	3100	27.218800	60.436200	33.749	572.510	24.695600	1870.26	47.0704	21.255
5900	Gujarat	Patan	2012	Kharif	Arhar/Tur	500	30.307200	44.721000	38.936	250.533	27.650200	461.45	9.7448	22.383
6744	Gujarat	Sabarkantha	2010	Kharif	Arhar/Tur	20600	29.937600	45.727000	38.535	632.954	26.074800	705.48	24.4224	22.018

1688 rows × 14 columns

[37]:
```
y_train
```

[37]:
```
4635      71200.0
1794      30100.0
3092       1300.0
3649      82200.0
6499        200.0
           ...
5734      61000.0
5191    1584000.0
5390        900.0
860         100.0
7270        500.0
Name: Production, Length: 6748, dtype: float64
```

[38]:
```
y_test
```

[38]:
```
6377    26700.0
8100      600.0
8045     1600.0
1127    10700.0
7249     5900.0
          ...
668      2000.0
1832      600.0
2181     1900.0
5900      500.0
6744    13100.0
Name: Production, Length: 1688, dtype: float64
```

[*]:
```
from sklearn.metrics import mean_absolute_error, mean_squared_error
from sklearn.model_selection import GridSearchCV
pipeline.fit(X_train, y_train)
y_pred = pipeline.predict(X_test)

mae = mean_absolute_error(y_test, y_pred)
mse = mean_squared_error(y_test, y_pred)
rmse = mse ** 0.5
print(f'MAE: {mae}, MSE: {mse}, RMSE: {rmse}')

# Hyperparameter tuning
param_grid = {
    'model__n_estimators': [100, 200],
    'model__max_depth': [None, 10, 20],
    'model__min_samples_split': [2, 5, 10]
}

grid_search = GridSearchCV(pipeline, param_grid, cv=5)
grid_search.fit(X_train, y_train)
print(f'Best params: {grid_search.best_params_}')
```

```
MAE: 15691.868675592783, MSE: 3914796002.928196, RMSE: 62568.33067078101
```

[*]:

[ ]:

[ ]: