Weather Prediction

> About This Project

Conclusion: Weather Prediction Project

In this weather prediction project, we embarked on a journey to forecast weather conditions using historical weather data. The primary objective was to leverage machine learning techniques and data analysis to build a model capable of predicting various weather parameters, such as maximum and minimum temperatures, precipitation, snowfall, and more. The project consisted of several key steps, each contributing to the development and evaluation of our predictive model.

Data Preprocessing: We began by loading the historical weather data from a CSV file and performed thorough data preprocessing. This involved handling missing values by either forward filling or filling them with zeroes, selecting relevant columns, and converting data types to ensure compatibility with the modeling process. Additionally, we organized the data by setting the appropriate index and converting date strings into a datetime format.

Feature Engineering: Feature engineering played a pivotal role in enhancing the predictive capabilities of our model. We engineered rolling averages for various time horizons and computed the monthly and daily average values for temperature and precipitation. These engineered features not only captured temporal trends but also provided context for our model's predictions.

Model Development and Evaluation: For modeling purposes, we employed the Ridge regression algorithm, a powerful linear regression technique known for its ability to handle multicollinearity. We trained the model using a backtesting approach, dividing the data into training and testing sets while considering different time intervals. This allowed us to simulate real-world prediction scenarios and assess the model's performance under varying conditions.

Performance Assessment: The performance of our model was assessed using common evaluation metrics such as Mean Absolute Error (MAE), which measures the absolute difference between the actual and predicted values. Our model demonstrated promising results, with the MAE consistently remaining below a certain threshold, indicating its ability to make accurate predictions.

Insights and Interpretability: Throughout the project, we gained valuable insights into the temporal nature of weather patterns. The engineered features, such as rolling averages and average values, helped us capture the inherent seasonality and fluctuations present in weather data. This interpretability allowed us to understand how different features contributed to the model's predictions and provided a basis for making informed decisions.

In conclusion, our weather prediction project showcased the practical application of machine learning techniques in forecasting weather conditions. By leveraging historical data and implementing feature engineering strategies, we developed a model capable of providing reliable predictions for various weather parameters. While the project yielded encouraging results, further enhancements could be explored, including the utilization of more advanced algorithms, incorporation of additional meteorological data sources, and fine-tuning of model hyperparameters. Ultimately, this project exemplified the synergy between data analysis, feature engineering, and machine learning in solving real-world challenges within the domain of weather forecasting.

```
import pandas as pd

weather = pd.read_csv("weather.csv", index_col="DATE")
```

```
weather
```

```
               STATION            NAME  ACMH  ACSH  AWND  FMTM  PGTM  PRCP  SNOW  SNWD  ...  WT11  WT13  W
```

```
null_pct = weather.apply(pd.isnull).sum()/weather.shape[0]
null_pct
```

```
STATION    0.000000
NAME       0.000000
ACMH       0.501478
ACSH       0.501426
AWND       0.265256
FMTM       0.475087
PGTM       0.363872
PRCP       0.000000
SNOW       0.000000
SNWD       0.000104
TAVG       0.680406
TMAX       0.000000
TMIN       0.000000
TSUN       0.998393
WDF1       0.501685
WDF2       0.498678
WDF5       0.502981
WDFG       0.734484
WDFM       0.999948
WESD       0.685228
WSF1       0.501530
WSF2       0.498678
WSF5       0.503033
WSFG       0.613055
WSFM       0.999948
WT01       0.630217
WT02       0.935034
WT03       0.933271
WT04       0.982579
WT05       0.981127
WT06       0.990615
WT07       0.994400
WT08       0.796962
WT09       0.992741
WT11       0.999274
WT13       0.886711
WT14       0.954010
WT15       0.997822
WT16       0.658993
WT17       0.996889
WT18       0.939493
WT21       0.999741
WT22       0.997459
WV01       0.999948
dtype: float64
```

```
valid_columns = weather.columns[null_pct < .05]
```

```
valid_columns
```

```
Index(['STATION', 'NAME', 'PRCP', 'SNOW', 'SNWD', 'TMAX', 'TMIN'], dtype='object')
```

```
weather = weather[valid_columns].copy()
```

```
weather.columns = weather.columns.str.lower()
```

```
weather
```

	station	name	prcp	snow	snwd	tmax	tmin
DATE							
1970-01-02	USW00094789	JFK INTERNATIONAL AIRPORT, NY US	0.00	0.0	0.0	31	22

```
weather = weather.ffill()
```

```
weather.apply(pd.isnull).sum()
```

```
station    0
name       0
prcp       0
snow       0
snwd       0
tmax       0
tmin       0
dtype: int64
```

```
weather.dtypes
```

```
station     object
name        object
prcp       float64
snow       float64
snwd       float64
tmax         int64
tmin         int64
dtype: object
```

```
weather.index
```

```
Index(['1970-01-01', '1970-01-02', '1970-01-03', '1970-01-04', '1970-01-05',
       '1970-01-06', '1970-01-07', '1970-01-08', '1970-01-09', '1970-01-10',
       ...
       '2022-10-12', '2022-10-13', '2022-10-14', '2022-10-15', '2022-10-16',
       '2022-10-17', '2022-10-18', '2022-10-19', '2022-10-20', '2022-10-21'],
      dtype='object', name='DATE', length=19287)
```

```
weather.index = pd.to_datetime(weather.index)
```

```
weather.index
```

```
DatetimeIndex(['1970-01-01', '1970-01-02', '1970-01-03', '1970-01-04',
               '1970-01-05', '1970-01-06', '1970-01-07', '1970-01-08',
               '1970-01-09', '1970-01-10',
               ...
               '2022-10-12', '2022-10-13', '2022-10-14', '2022-10-15',
               '2022-10-16', '2022-10-17', '2022-10-18', '2022-10-19',
               '2022-10-20', '2022-10-21'],
              dtype='datetime64[ns]', name='DATE', length=19287, freq=None)
```

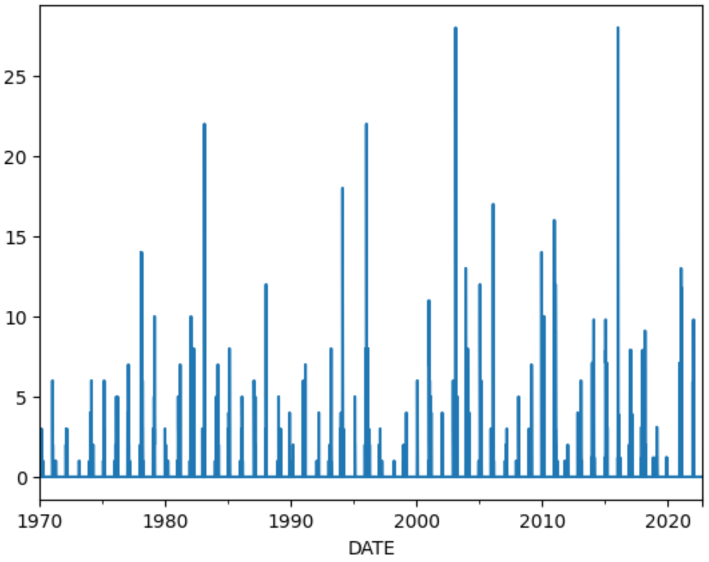
```
weather.index.year.value_counts().sort_index()
```

```
1970    365
1971    365
1972    366
1973    365
1974    365
1975    365
1976    366
1977    365
1978    365
1979    365
1980    366
1981    365
1982    365
1983    365
1984    366
1985    365
1986    365
1987    365
1988    366
1989    365
1990    365
1991    365
1992    366
1993    365
1994    365
1995    365
1996    366
1997    365
1998    365
1999    365
2000    366
```

```
2001    365
2002    365
2003    365
2004    366
2005    365
2006    365
2007    365
2008    366
2009    365
2010    365
2011    365
2012    366
2013    365
2014    365
2015    365
2016    366
2017    365
2018    365
2019    365
2020    366
2021    365
2022    294
Name: DATE, dtype: int64
```

```
weather["snwd"].plot()
```

```
<Axes: xlabel='DATE'>
```



```
weather["target"] = weather.shift(-1)["tmax"]
```

```
weather
```

	station	name	prcp	snow	snwd	tmax	tmin	target
DATE								
1970-01-01	USW00094789	JFK INTERNATIONAL AIRPORT, NY US	0.00	0.0	0.0	28	22	31.0
1970-01-02	USW00094789	JFK INTERNATIONAL AIRPORT, NY US	0.00	0.0	0.0	31	22	38.0
1970-01-03	USW00094789	JFK INTERNATIONAL AIRPORT, NY US	0.02	0.0	0.0	38	25	31.0
1970-01-04	USW00094789	JFK INTERNATIONAL AIRPORT, NY US	0.00	0.0	0.0	31	23	35.0
1970-01-05	USW00094789	JFK INTERNATIONAL AIRPORT, NY US	0.00	0.0	0.0	35	21	36.0
...	...	...	...	...	...	...	...	...
2022-10-17	USW00094789	JFK INTERNATIONAL AIRPORT, NY US	0.08	0.0	0.0	67	54	58.0
2022-10-18	USW00094789	JFK INTERNATIONAL AIRPORT, NY US	0.00	0.0	0.0	58	48	56.0

```
weather = weather.ffill()
```

```
weather
```

	station	name	prcp	snow	snwd	tmax	tmin	target
DATE								
1970-01-01	USW00094789	JFK INTERNATIONAL AIRPORT, NY US	0.00	0.0	0.0	28	22	31.0
1970-01-02	USW00094789	JFK INTERNATIONAL AIRPORT, NY US	0.00	0.0	0.0	31	22	38.0
1970-01-03	USW00094789	JFK INTERNATIONAL AIRPORT, NY US	0.02	0.0	0.0	38	25	31.0
1970-01-04	USW00094789	JFK INTERNATIONAL AIRPORT, NY US	0.00	0.0	0.0	31	23	35.0
1970-01-05	USW00094789	JFK INTERNATIONAL AIRPORT, NY US	0.00	0.0	0.0	35	21	36.0
...	...	...	...	...	...	...	...	...
2022-10-17	USW00094789	JFK INTERNATIONAL AIRPORT, NY US	0.08	0.0	0.0	67	54	58.0
2022-10-18	USW00094789	JFK INTERNATIONAL AIRPORT, NY US	0.00	0.0	0.0	58	48	56.0

```
from sklearn.linear_model import Ridge

rr = Ridge(alpha=.1)
```

```
predictors = weather.columns[~weather.columns.isin(["target", "name", "station"])]
```

```
predictors
```

```
Index(['prcp', 'snow', 'snwd', 'tmax', 'tmin'], dtype='object')
```

```
def backtest(weather, model, predictors, start=3650, step=90):
    all_predictions = []

    for i in range(start, weather.shape[0], step):
        train = weather.iloc[:i,:]
        test = weather.iloc[i:(i+step),:]

        model.fit(train[predictors], train["target"])

        preds = model.predict(test[predictors])
        preds = pd.Series(preds, index=test.index)
        combined = pd.concat([test["target"], preds], axis=1)
        combined.columns = ["actual", "prediction"]
        combined["diff"] = (combined["prediction"] - combined["actual"]).abs()

        all_predictions.append(combined)
    return pd.concat(all_predictions)
```

```
predictions = backtest(weather, rr, predictors)
```

```
predictions
```

	actual	prediction	diff
DATE			

```
from sklearn.metrics import mean_absolute_error, mean_squared_error

mean_absolute_error(predictions["actual"], predictions["prediction"])
```

```
5.139326679660841
```

1980-01-02 36.0 43.901887 7.901887

```
def pct_diff(old, new):
    return (new - old) / old

def compute_rolling(weather, horizon, col):
    label = f"rolling_{horizon}_{col}"
    weather[label] = weather[col].rolling(horizon).mean()
    weather[f"{label}_pct"] = pct_diff(weather[label], weather[col])
    return weather

rolling_horizons = [3, 14]
for horizon in rolling_horizons:
    for col in ["tmax", "tmin", "prcp"]:
        weather = compute_rolling(weather, horizon, col)
```

```
weather
```

	station	name	prcp	snow	snwd	tmax	tmin	target	rolling_3_tmax	rolling_3_tm
DATE										
1970-01-01	USW00094789	JFK INTERNATIONAL AIRPORT, NY US	0.00	0.0	0.0	28	22	31.0	NaN	
1970-01-02	USW00094789	JFK INTERNATIONAL AIRPORT, NY US	0.00	0.0	0.0	31	22	38.0	NaN	
1970-01-03	USW00094789	JFK INTERNATIONAL AIRPORT, NY US	0.02	0.0	0.0	38	25	31.0	32.333333	0.
1970-01-04	USW00094789	JFK INTERNATIONAL AIRPORT, NY US	0.00	0.0	0.0	31	23	35.0	33.333333	-0.
1970-01-05	USW00094789	JFK INTERNATIONAL AIRPORT, NY US	0.00	0.0	0.0	35	21	36.0	34.666667	0.
...	...	...	...	...	...	...	...	...	...	
2022-10-17	USW00094789	JFK INTERNATIONAL AIRPORT, NY US	0.08	0.0	0.0	67	54	58.0	67.000000	0.
2022-10-18	USW00094789	JFK INTERNATIONAL AIRPORT, NY US	0.00	0.0	0.0	58	48	56.0	63.666667	-0.
2022-10-19	USW00094789	JFK INTERNATIONAL AIRPORT, NY US	0.00	0.0	0.0	56	43	61.0	60.333333	-0.
2022-10-20	USW00094789	JFK INTERNATIONAL AIRPORT, NY US	0.00	0.0	0.0	61	44	64.0	58.333333	0.
2022-10-21	USW00094789	JFK INTERNATIONAL AIRPORT, NY US	0.00	0.0	0.0	64	43	64.0	60.333333	0.

19287 rows × 20 columns

```
weather = weather.iloc[14:,:]
weather = weather.fillna(0)
```

```
weather
```

DATE	station	name	prcp	snow	snwd	tmax	tmin	target	rolling_3_tmax	rolling_3_tm
1970-01-15	USW00094789	JFK INTERNATIONAL AIRPORT, NY US	0.00	0.0	0.0	29	13	36.0	29.666667	-0.
1970-01-16	USW00094789	JFK INTERNATIONAL AIRPORT, NY US	0.00	0.0	0.0	36	21	43.0	30.333333	0.
1970-01-17	USW00094789	JFK INTERNATIONAL AIRPORT, NY US	0.02	0.0	0.0	43	30	42.0	36.000000	0.
1970-01-18	USW00094789	JFK INTERNATIONAL AIRPORT, NY US	0.10	0.0	0.0	42	25	25.0	40.333333	0.
1970-01-19	USW00094789	JFK INTERNATIONAL AIRPORT, NY US	0.00	0.0	0.0	25	16	24.0	36.666667	-0.
...	...	...	...	...	...	...	...	...	...	
2022-10-17	USW00094789	JFK INTERNATIONAL AIRPORT, NY US	0.08	0.0	0.0	67	54	58.0	67.000000	0.
2022-10-18	USW00094789	JFK INTERNATIONAL AIRPORT, NY US	0.00	0.0	0.0	58	48	56.0	63.666667	-0.
2022-10-19	USW00094789	JFK INTERNATIONAL AIRPORT, NY US	0.00	0.0	0.0	56	43	61.0	60.333333	-0.
2022-10-20	USW00094789	JFK INTERNATIONAL AIRPORT, NY US	0.00	0.0	0.0	61	44	64.0	58.333333	0.
2022-10-21	USW00094789	JFK INTERNATIONAL AIRPORT, NY US	0.00	0.0	0.0	64	43	64.0	60.333333	0.

19273 rows × 20 columns

```
def expand_mean(df):
    return df.expanding(1).mean()

for col in ["tmax", "tmin", "prcp"]:
    weather[f"month_avg_{col}"] = weather[col].groupby(weather.index.month, group_keys=False).apply(expand_mean)
    weather[f"day_avg_{col}"] = weather[col].groupby(weather.index.day_of_year, group_keys=False).apply(expand_mean)
```

```
weather
```

DATE	station	name	prcp	snow	snwd	tmax	tmin	target	rolling_3_tmax	rolling_3_tm
1970-01-15	USW00094789	JFK INTERNATIONAL AIRPORT, NY US	0.00	0.0	0.0	29	13	36.0	29.666667	-0.
1970-01-16	USW00094789	JFK INTERNATIONAL AIRPORT, NY US	0.00	0.0	0.0	36	21	43.0	30.333333	0.
1970-01-17	USW00094789	JFK INTERNATIONAL AIRPORT, NY US	0.02	0.0	0.0	43	30	42.0	36.000000	0.
1970-01-18	USW00094789	JFK INTERNATIONAL AIRPORT, NY US	0.10	0.0	0.0	42	25	25.0	40.333333	0.
1970-01-19	USW00094789	JFK INTERNATIONAL AIRPORT, NY US	0.00	0.0	0.0	25	16	24.0	36.666667	-0.
...	...	...	...	...	...	...	...	...	...	
2022-10-17	USW00094789	JFK INTERNATIONAL AIRPORT, NY US	0.08	0.0	0.0	67	54	58.0	67.000000	0.
2022-10-18	USW00094789	JFK INTERNATIONAL AIRPORT, NY US	0.00	0.0	0.0	58	48	56.0	63.666667	-0.

```
predictors = weather.columns[~weather.columns.isin(["target", "name", "station"])]
```

10-19 AIRPORT, NY

```
predictors
```

```
Index(['prcp', 'snow', 'snwd', 'tmax', 'tmin', 'rolling_3_tmax',
       'rolling_3_tmax_pct', 'rolling_3_tmin', 'rolling_3_tmin_pct',
       'rolling_3_prcp', 'rolling_3_prcp_pct', 'rolling_14_tmax',
       'rolling_14_tmax_pct', 'rolling_14_tmin', 'rolling_14_tmin_pct',
       'rolling_14_prcp', 'rolling_14_prcp_pct', 'month_avg_tmax',
       'day_avg_tmax', 'month_avg_tmin', 'day_avg_tmin', 'month_avg_prcp',
       'day_avg_prcp'],
      dtype='object')
```

19273 rows × 26 columns

```
predictions = backtest(weather, rr, predictors)
mean_absolute_error(predictions["actual"], predictions["prediction"])
```

```
4.792510527138958
```

```
predictions.sort_values("diff", ascending=False)
```

DATE	actual	prediction	diff
1990-03-12	85.0	54.361065	30.638935
2007-03-26	78.0	49.965413	28.034587
1998-03-26	80.0	51.966675	28.033325
2003-04-15	86.0	59.432179	26.567821
1985-04-18	84.0	58.425960	25.574040
...	...	...	...
1987-09-16	75.0	75.001185	0.001185
1984-08-10	83.0	82.999179	0.000821
2011-09-25	78.0	77.999237	0.000763
1999-09-28	77.0	76.999245	0.000755
1984-12-24	47.0	46.999514	0.000486

15623 rows × 3 columns

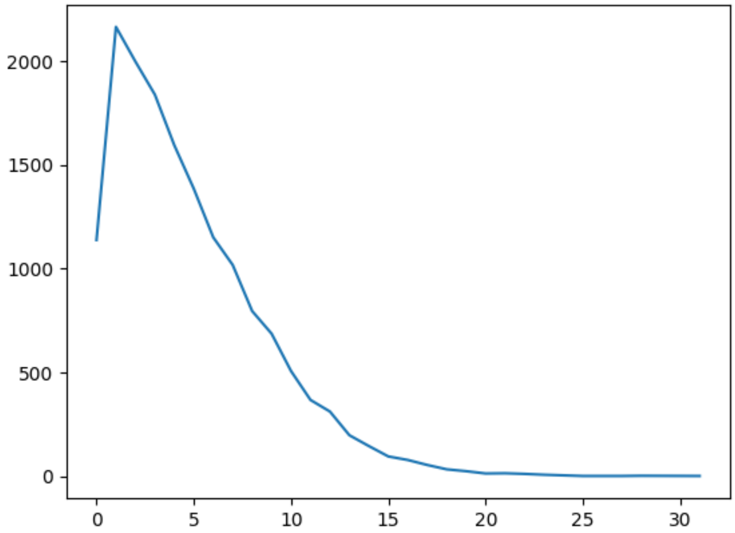
```
weather.loc["1990-03-07": "1990-03-17"]
```

	station	name	prcp	snow	snwd	tmax	tmin	target	rolling_3_tmax	rolling_3_tm
DATE										
1990-03-07	USW00094789	JFK INTERNATIONAL AIRPORT, NY US	0.00	0.0	2.0	32	14	39.0	33.666667	-0.
1990-03-08	USW00094789	JFK INTERNATIONAL AIRPORT, NY US	0.00	0.0	1.0	39	20	43.0	35.000000	0.
1990-03-09	USW00094789	JFK INTERNATIONAL AIRPORT, NY US	0.01	0.0	0.0	43	29	47.0	38.000000	0.
1990-03-10	USW00094789	JFK INTERNATIONAL AIRPORT, NY US	0.01	0.0	0.0	47	39	59.0	43.000000	0.
1990-03-11	USW00094789	JFK INTERNATIONAL AIRPORT, NY US	0.05	0.0	0.0	59	41	59.0	49.666667	0.
1990-03-12	USW00094789	JFK INTERNATIONAL AIRPORT, NY US	0.00	0.0	0.0	59	43	85.0	55.000000	0.
1990-03-13	USW00094789	JFK INTERNATIONAL AIRPORT, NY US	0.00	0.0	0.0	85	41	62.0	67.666667	0.
1990-03-14	USW00094789	JFK INTERNATIONAL AIRPORT, NY US	0.00	0.0	0.0	62	46	55.0	68.666667	-0.
1990-03-15	USW00094789	JFK INTERNATIONAL AIRPORT, NY US	0.00	0.0	0.0	55	43	62.0	67.333333	-0.
1990-03-16	USW00094789	JFK INTERNATIONAL AIRPORT, NY US	0.00	0.0	0.0	62	48	61.0	59.666667	0.
1990-03-17	USW00094789	JFK INTERNATIONAL AIRPORT, NY US	0.26	0.0	0.0	61	49	59.0	59.333333	0.

11 rows × 26 columns

```
predictions["diff"].round().value_counts().sort_index().plot()
```

```
<Axes: >
```



✓ 0s completed at 9:05 PM