Applications PanchangulaKrishnacha... Thunar ML_Assignment_update... 2025-11-25 09:36 Cloud User

ML_Assignment_updated.ipynb - ML_assign - Visual Studio Code

File Edit Selection View Go Run Terminal Help

ML_Assignment_updated.ipynb × Submission.csv

ML_Assignment_updated.ipynb > Updating the predictions to the submission.csv file > Q12. Why can't Linear Regression alone capture time-of-day effects effectively?

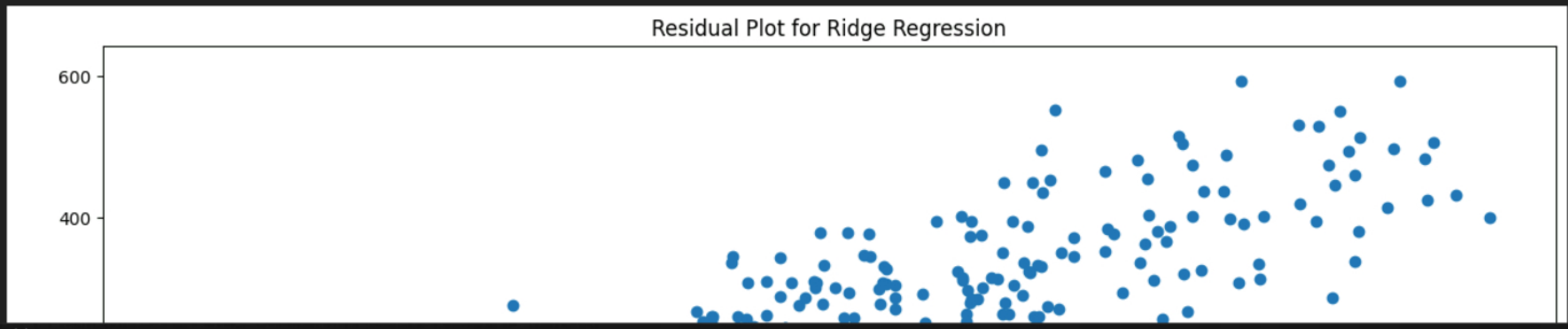
+ Code + Markdown | Run All Restart Clear All Outputs | Jupyter Variables Outline ··· base (Python 3.12.7)

Q8- Plotting the residual for the best model

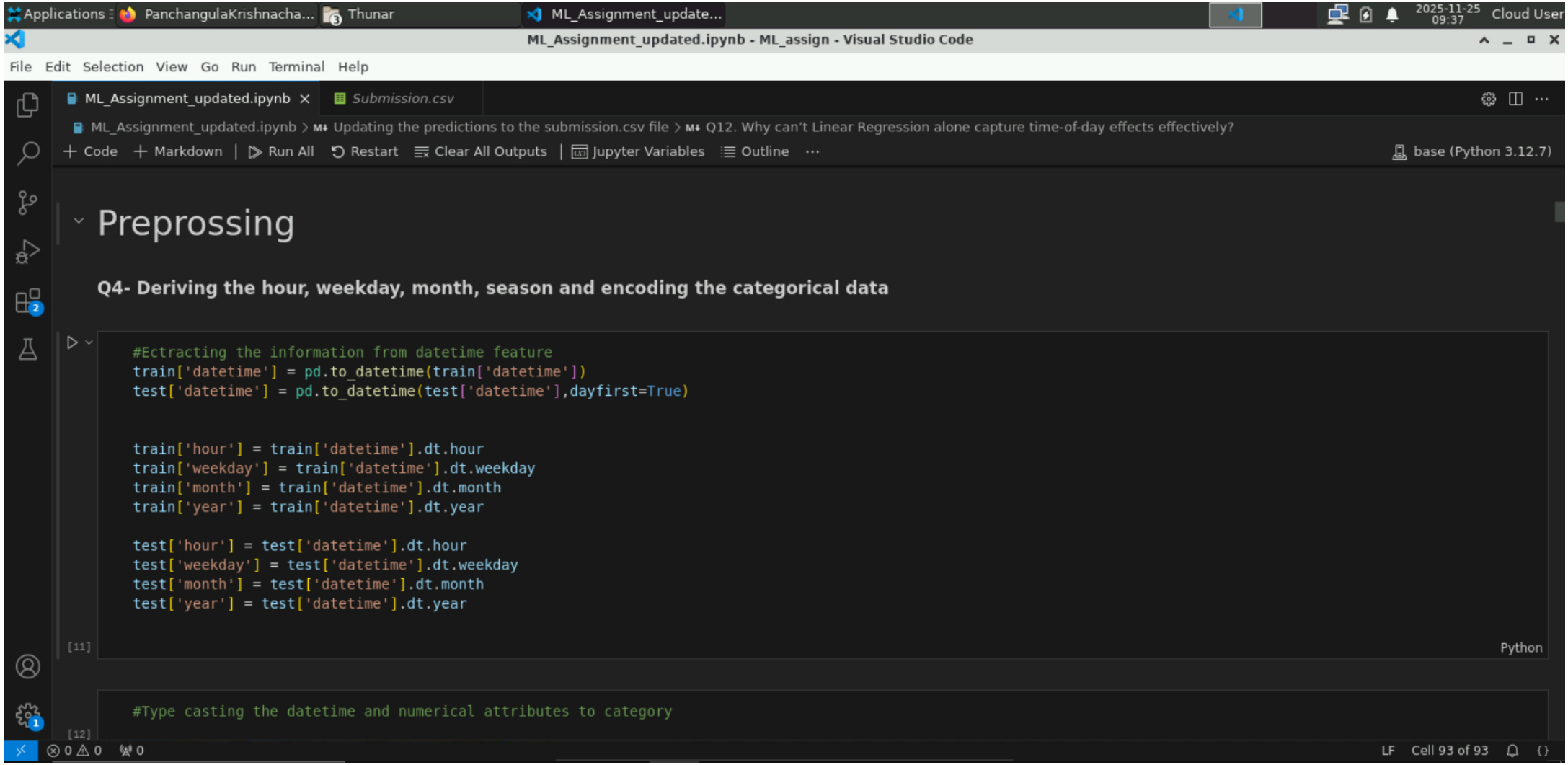
+ Code + Markdown

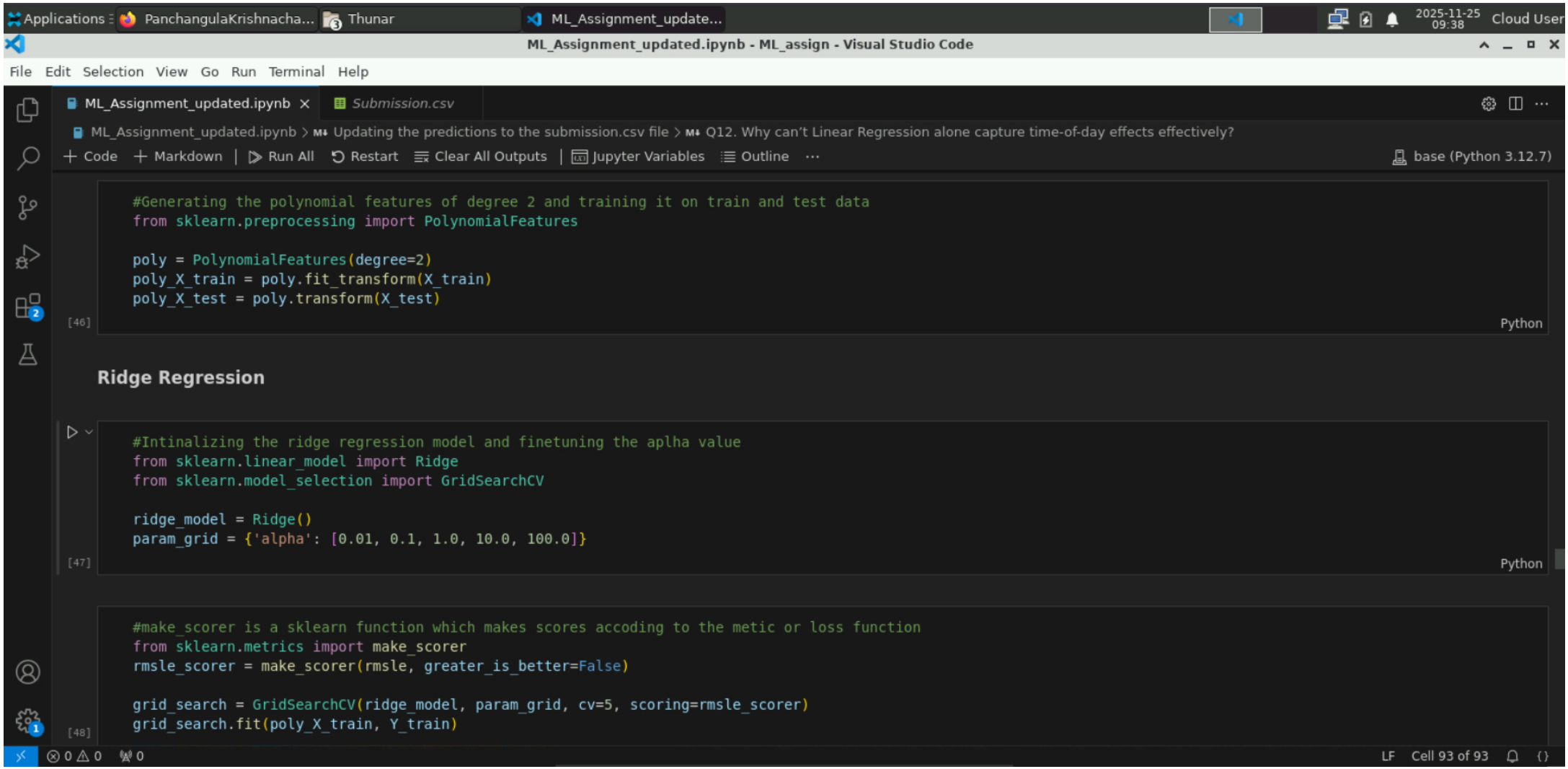
```
import matplotlib.pyplot as plt

#Residual plot for the best model
fig, ax = plt.subplots(figsize=(15,8))
ax.scatter(Y_test, Y_test - best_model_predictions)
ax.axhline(lw=2, color='green')
ax.set_xlabel('Observed Values')
ax.set_ylabel('Residuals')
ax.set_title(f'Residual Plot for {best_model_name}')
plt.show()
```

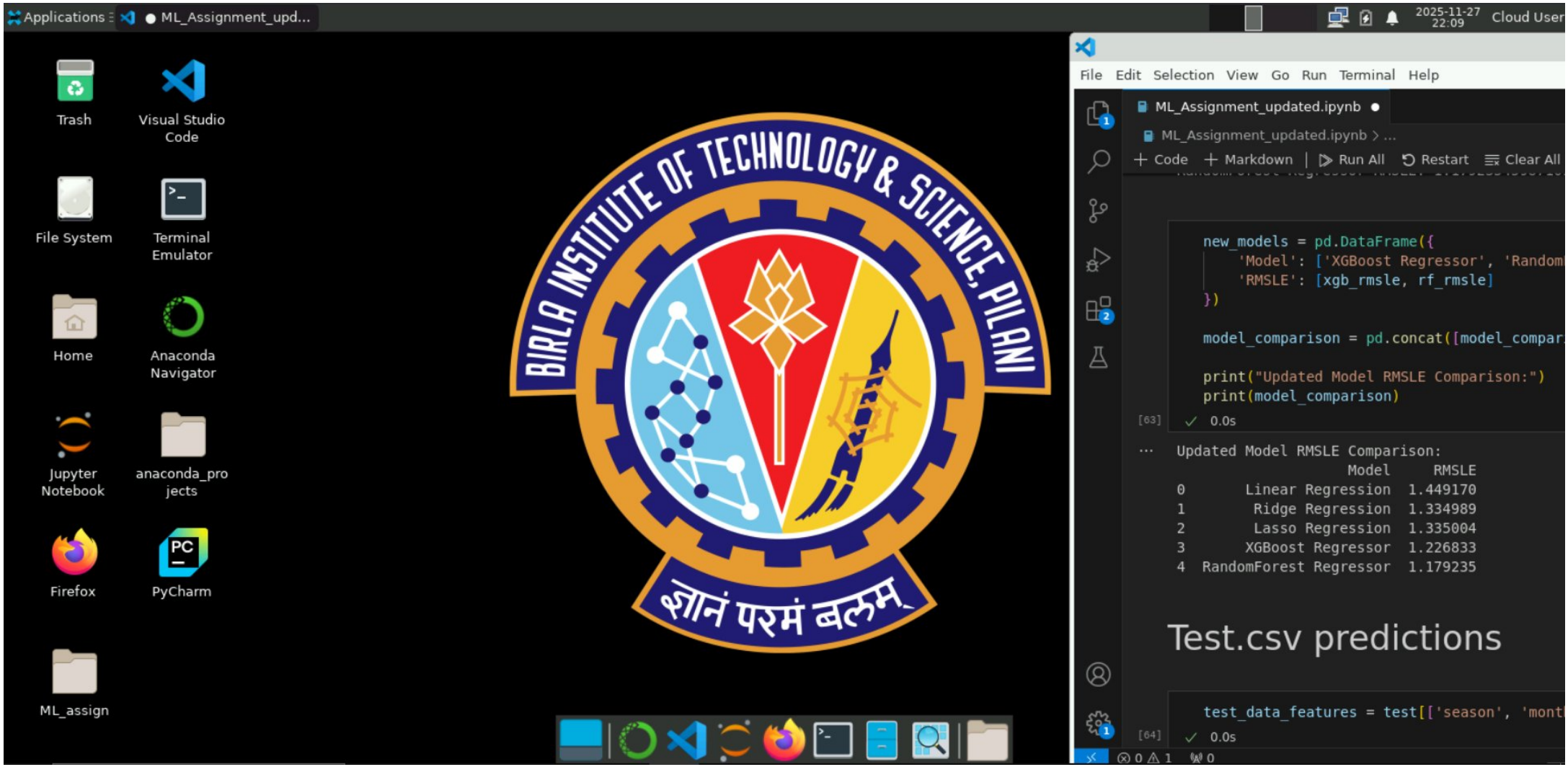
[59] Python



0 0 0 LF Cell 93 of 93


Applications
PanchangulaKrishnacha...
Thunar
ML_Assignment_update...
2025-11-25
09:37
Cloud User
ML_Assignment_updated.ipynb - ML_assign - Visual Studio Code
File Edit Selection View Go Run Terminal Help
ML_Assignment_updated.ipynb
Submission.csv
ML_Assignment_updated.ipynb > Updating the predictions to the submission.csv file > Q12. Why can't Linear Regression alone capture time-of-day effects effectively?
+ Code + Markdown | Run All Restart Clear All Outputs | Jupyter Variables Outline ...
base (Python 3.12.7)
Preprossing
Q4- Deriving the hour, weekday, month, season and encoding the categorical data
#Ectracting the information from datetime feature
train['datetime'] = pd.to_datetime(train['datetime'])
test['datetime'] = pd.to_datetime(test['datetime'],dayfirst=True)
train['hour'] = train['datetime'].dt.hour
train['weekday'] = train['datetime'].dt.weekday
train['month'] = train['datetime'].dt.month
train['year'] = train['datetime'].dt.year
test['hour'] = test['datetime'].dt.hour
test['weekday'] = test['datetime'].dt.weekday
test['month'] = test['datetime'].dt.month
test['year'] = test['datetime'].dt.year
[11]
Python
#Type casting the datetime and numerical attributes to category
[12]
0 0 0
LF Cell 93 of 93


Applications
PanchangulaKrishnacha...
Thunar
ML_Assignment_update...
2025-11-25
09:38
Cloud User
ML_Assignment_updated.ipynb - ML_assign - Visual Studio Code
File Edit Selection View Go Run Terminal Help
ML_Assignment_updated.ipynb
Submission.csv
ML_Assignment_updated.ipynb > Updating the predictions to the submission.csv file > Q12. Why can't Linear Regression alone capture time-of-day effects effectively?
Code Markdown Run All Restart Clear All Outputs Jupyter Variables Outline
base (Python 3.12.7)
#Generating the polynomial features of degree 2 and training it on train and test data
from sklearn.preprocessing import PolynomialFeatures
poly = PolynomialFeatures(degree=2)
poly_X_train = poly.fit_transform(X_train)
poly_X_test = poly.transform(X_test)
[46]
Python
Ridge Regression
#Intinalizing the ridge regression model and finetuning the aplha value
from sklearn.linear_model import Ridge
from sklearn.model_selection import GridSearchCV
ridge_model = Ridge()
param_grid = {'alpha': [0.01, 0.1, 1.0, 10.0, 100.0]}
[47]
Python
#make_scorer is a sklearn function which makes scores accoding to the metic or loss function
from sklearn.metrics import make_scorer
rmsle_scorer = make_scorer(rmsle, greater_is_better=False)
grid_search = GridSearchCV(ridge_model, param_grid, cv=5, scoring=rmsle_scorer)
grid_search.fit(poly_X_train, Y_train)
[48]
0 0 0
LF Cell 93 of 93


Applications
ML_Assignment_upd...
2025-11-27
22:09
Cloud User
Trash
Visual Studio Code
File System
Terminal Emulator
Home
Anaconda Navigator
Jupyter Notebook
anaconda_projects
Firefox
PyCharm
ML_assign
BIRLA INSTITUTE OF TECHNOLOGY & SCIENCE, PILANI
ज्ञानं परमं बलम्
File Edit Selection View Go Run Terminal Help
ML_Assignment_updated.ipynb
ML_Assignment_updated.ipynb > ...
+ Code + Markdown | Run All Restart Clear All
new_models = pd.DataFrame({
'Model': ['XGBoost Regressor', 'Random
'RMSLE': [xgb_rmsle, rf_rmsle]
})
model_comparison = pd.concat([model_compar
print("Updated Model RMSLE Comparison:")
print(model_comparison)
[63] ✓ 0.0s
Updated Model RMSLE Comparison:
Model RMSLE
0 Linear Regression 1.449170
1 Ridge Regression 1.334989
2 Lasso Regression 1.335004
3 XGBoost Regressor 1.226833
4 RandomForest Regressor 1.179235
Test.csv predictions
test_data_features = test[['season', 'monti
[64] ✓ 0.0s
⊗ 0 ⚠ 1 0