jupyter Economic_Feature_Taxonomy Last Checkpoint: 4 hours ago

File Edit View Run Kernel Settings Help

Trusted

Code

JupyterLab Python 3 (ipykernel)

[9]:
```
# 🧬 Economic Feature Label Taxonomy Builder

import random
import pandas as pd

# 📌 Linchpin concepts — critical economic indicators
linchpins = [
    "GDP_growth", "CPI_inflation", "unemployment_rate", "VIX_index", "10y_treasury_yield",
    "federal_funds_rate", "consumer_sentiment_index", "PMI_manufacturing", "housing_starts",
    "initial_jobless_claims", "ISM_nonmanufacturing_index", "S&P500_returns"
]

# 🧠 Academic Theory concepts
academic_theory = [
    "output_gap", "term_premium", "monetary_velocity", "natural_rate_of_unemployment",
    "total_factor_productivity", "okuns_law_residual", "fisher_effect_offset",
    "real_potential_GDP", "nairu_gap", "neutral_interest_rate"
]

# 🧱 Professional Signal concepts
professional_signal = [
    "credit_spread", "yield_curve_slope", "foreign_exchange_rate", "corporate_earnings_forecast",
    "market_liquidity", "high_yield_spread", "libor_ois_spread", "repo_market_stress",
    "bank_lending_survey", "margin_debt_growth"
]

# 🧪 Engineered / Model Outputs
engineered_signals = [
    "nowcast_gdp", "recession_model_score", "real_interest_rate_gap", "economic_surprise_index",
    "tail_risk_score", "volatility_term_structure", "machine_sentiment_factor", "fwd_earnings_surprise",
    "macro_risk_factor_1", "regime_switch_probability"
]

# 🔠 Layman Friendly Concepts
layman_friendly = [
    "oil_price", "gas_price", "housing_prices", "grocery_index", "retail_sales",
    "job_postings_online", "mortgage_rate", "used_car_prices", "electricity_cost_index", "minimum_wage_trend"
]

# Combine all and shuffle to simulate assignment
all_features = linchpins + academic_theory + professional_signal + engineered_signals + layman_friendly

# 🎯 Simulate a massive raw feature matrix of 250 variables
num_raw_features = 250

# If fewer labels than features, repeat labels with slight variation
if len(all_features) < num_raw_features:
    extra_needed = num_raw_features - len(all_features)
    repeated = [f"{name}_{i}" for i, name in enumerate(random.choices(all_features, k=extra_needed))]
    all_features += repeated

# Final shuffle
random.shuffle(all_features)

# Export to dataframe for downstream use
raw_feature_map = pd.DataFrame({
    "raw_feature_id": [f"feature_{i}" for i in range(num_raw_features)],
    "economic_label": all_features[:num_raw_features]
})

# Save to CSV
raw_feature_map.to_csv("raw_feature_label_mapping.csv", index=False)

# Preview
raw_feature_map.head(10)
```

[9]:

	raw_feature_id	economic_label
0	feature_0	minimum_wage_trend_43
1	feature_1	neutral_interest_rate_153
2	feature_2	market_liquidity
3	feature_3	initial_jobless_claims_11
4	feature_4	corporate_earnings_forecast_71
5	feature_5	foreign_exchange_rate_187
6	feature_6	federal_funds_rate_53
7	feature_7	real_potential_GDP_157
8	feature_8	retail_sales
9	feature_9	PMI_manufacturing_56

[ ]: