Data Science Internship Projects - Solved Version

Project 1: Real-Time Stock Market Trend Predictor & Visual Analytics Dashboard

This project builds a real-time dashboard that fetches live stock data, computes indicators, and predicts trends using ML models like ARIMA, Prophet, or XGBoost. The dashboard provides visual insights and actionable trading signals such as Buy, Hold, or Sell recommendations.

Steps:

1. Collect real-time stock data using *yfinance* API.
2. Perform data preprocessing and compute indicators such as Moving Averages, RSI, MACD, and volatility.
3. Apply models like Prophet for time-series forecasting or XGBoost for classification.
4. Evaluate using MAE, RMSE (forecast) and accuracy, precision, recall (classification).
5. Display insights in a Streamlit dashboard with interactive charts using Plotly.

Key Python Code:

```
import yfinance as yf data = yf.download('AAPL', period='30d', interval='1h')
data['MA_10'] = data['Close'].rolling(10).mean() data['MA_50'] =
data['Close'].rolling(50).mean()
from prophet import Prophet df = data.reset_index()[['Datetime','Close']].rename(col
umns={'Datetime':'ds','Close':'y'}) model = Prophet() model.fit(df) future =
model.make_future_dataframe(periods=24, freq='H') forecast = model.predict(future)
# XGBoost for trend classification from xgboost import XGBClassifier clf =
XGBClassifier() clf.fit(X_train, y_train) predictions = clf.predict(X_test)
```

Deliverables:

• Streamlit dashboard displaying real-time stock data.
• Predictive model output showing Buy/Sell/Hold signals.
• Performance metrics: MAE, RMSE, Accuracy, F1-score.
• Documentation explaining approach, evaluation, and limitations.

Project 2: Smart Healthcare Data Analyzer for Disease Pattern Detection

This project analyzes healthcare data to identify disease patterns, perform clustering, and predict disease risk using ML classification models. The system visualizes correlations and predictive accuracy through a dashboard.

Steps:

1. Load healthcare dataset (e.g., Diabetes dataset) and handle missing values.
2. Normalize features using StandardScaler and impute missing data.
3. Perform EDA using Seaborn visualizations (correlation heatmaps, histograms).
4. Apply KMeans clustering to detect hidden patient groups.
5. Train classification models like RandomForest to predict disease outcomes.
6. Evaluate performance using Accuracy, ROC-AUC, and Confusion Matrix.
7. Deploy on a Streamlit dashboard for visualization.

Key Python Code:

```
import pandas as pd from sklearn.preprocessing import StandardScaler from
sklearn.impute import SimpleImputer df = pd.read_csv('health_data.csv') imputer =
SimpleImputer(strategy='median') df = pd.DataFrame(imputer.fit_transform(df),
columns=df.columns) scaler = StandardScaler() df[df.columns] =
scaler.fit_transform(df)
from sklearn.ensemble import RandomForestClassifier from sklearn.model_selection
import train_test_split from sklearn.metrics import classification_report X_train,
X_test, y_train, y_test = train_test_split(X, y, test_size=0.2) clf =
RandomForestClassifier() clf.fit(X_train, y_train) preds = clf.predict(X_test)
print(classification_report(y_test, preds))
```

Deliverables:

• Cleaned and processed healthcare dataset.
• EDA visuals: correlation heatmap, cluster scatter plot.
• Predictive model: RandomForest or Logistic Regression.
• Streamlit app for analysis and prediction.
• Ethical note: ensures no personally identifiable data used.

Conclusion:

These projects demonstrate data science proficiency through real-time data handling, visualization, machine learning, and ethical analysis. The first project emphasizes financial forecasting, while the second focuses on healthcare insights for decision support.