Assignment 1

Assignment Problem:

The assignment involves analyzing Airbnb data for Copenhagen using Python and various data science libraries. It includes data cleaning, exploration, visualization, and the application of machine learning algorithms to classify rental properties.

Solution Steps:

Data Cleaning: Unwanted columns are dropped, and rows with zero reviews are removed. The 'neighbourhood_cleansed' values are fixed.

Currency Conversion: Prices are converted to DKK.

Visualization: Various visualizations are created, including a word cloud for house names, a map of listings, boxplots for neighborhood prices, and bar charts for top hosts.

Descriptive Analysis: Statistics are calculated for room types and prices in neighborhoods.

Machine Learning: A Naïve Bayes and k-Nearest Neighbor model are implemented to classify rental properties into "expensive" and "affordable."

Algorithms Used:

Naïve Bayes: Used for classification based on self-chosen features.

k-Nearest Neighbor (k-NN): Another classification model applied to the data.

Performance:

The Naïve Bayes model is evaluated for accuracy, and a classification report is generated.

The k-NN model is also evaluated, providing accuracy and a classification report.

Reflection on Learning Outcome:

The assignment has provided valuable learning experiences in data analysis, preprocessing, and visualization. Specific skills include data cleaning, geospatial visualization using libraries like folium and geopandas, and addressing data issues such as symbol uniformity. The use of machine learning algorithms for classification, particularly Naïve Bayes and k-NN, enhances the understanding of predictive modeling.

Assignment 2

Assignment Problem:

The assignment tackles a classification problem using the exoplanet dataset.

The goal is to predict the disposition status of exoplanets (FALSE POSITIVE, CANDIDATE, CONFIRMED) based on various features.

Solution Approach:

Data Acquisition and Preparation:

The dataset is loaded and explored using pandas.

Columns are renamed for better clarity.

Missing values are analyzed, and the top features with missing values are visualized.

Feature Engineering and Transformation:

Potential outliers in numeric features are identified.

Columns with 100% missing values are dropped.

Irrelevant columns (IDs, names) are removed.

Numeric features are log-transformed.

Target Variable Encoding:

The target variables, 'ExoplanetArchiveDisposition' and 'DispositionUsingKeplerData,' are encoded into numerical values (0, 1, 2).

Algorithms Used:

k-Nearest Neighbors (k-NN):

The script uses scikit-learn's KNeighborsClassifier.

Grid search is performed for hyperparameter tuning.

Model evaluation involves training-test splits and visualization of accuracy vs. neighbors.

Gaussian Naive Bayes:

The script employs scikit-learn's GaussianNB.

The model is trained and evaluated using accuracy, precision, recall, and F1-score.

Performance of the Final Model:

Both k-NN and Naive Bayes models are trained and evaluated.

Evaluation metrics such as accuracy, precision, recall, and F1-score are calculated.

Confusion matrix and ROC curves are visualized for model performance assessment.

Learning Outcome Reflection:

The script demonstrates a comprehensive understanding of the machine learning workflow, including data exploration, cleaning, feature engineering, model training, and evaluation.

The use of different algorithms highlights flexibility in model selection based on the problem at hand.

The analysis provides insights into data patterns, outliers, and the impact of feature transformations on model performance.

Assignment 3

Assignment Problem:

The assignment focuses on optimizing methods for generating long-term corrected (LTC) wind data for Vestas, a global wind energy company.

Two types of time series data are provided: mast data, representing wind conditions at a specific location, and meso data, which provides a longer historical context but covers a broader area.

Solution Approach:

Data Acquisition:

The script handles realistic wind industry data, specifically from Vestas' wind mast and meso datasets.

Utilizes netCDF format for mast datasets and processes CSV data for meso datasets.

Data Preparation:

Addresses challenges of large datasets by implementing data preprocessing tasks.

Manages missing values, outliers, and scales data appropriately for regression modeling.

Feature Engineering:

Selects relevant wind speed and wind direction signals, considering altitude, coverage of all seasons, and interpolation for meso data.

Converts mast data from Danish time to UTC time, aligning it with meso data.

Resamples mast data to match the hourly frequency of meso data, handling wind speed and wind direction signals individually.

Exploratory Analysis:

Conducts exploratory analysis using tables and graphs, describing features of interest and performing correlation analysis.

Fits Weibull distributions to mast, resampled mast, and meso data.

Algorithms Used:

Employs Linear Regression for modeling wind speed based on wind direction for both mast and meso datasets.

Utilizes cross-validation techniques to assess model performance.

Model Performance:

For the mast group, the Mean Squared Error (MSE) and R-squared (R^2) metrics are used for evaluation.

Similarly, the meso group undergoes regression analysis, with MSE and R^2 as performance metrics.

Reflection on Learning Outcome:

The assignment presents a real-world machine learning task, emphasizing the importance of handling large datasets and applying regression techniques for wind data correction.

Provides insights into the complexities of wind data preprocessing, exploratory analysis, and model evaluation in a practical industry context.

Encourages critical thinking about the trade-offs between regression and neural networks, considering time and cost implications.

Assignment 4

Assignment Problem: The assignment aimed to analyze candidate tests from Danish television networks DR and TV2, examining responses on a scale of five for various political parties. The goal was to understand the political landscape, crucial questions, party positions, age distribution, candidate confidence, response differences, party classification, and clustering of elected candidates.

Solution Approach:

1. **Data Acquisition and Preparation:**
    - Datasets from both networks were loaded using pandas, and missing data was addressed.
    - Data normalization and standardization were performed for PCA and clustering.
2. **PCA and Dimensionality Reduction:**
    - Principal Component Analysis (PCA) was applied to visualize the political landscape in two dimensions.
    - The results were plotted using party colors to represent different political affiliations.
3. **Crucial Questions Analysis:**
    - The most concerning questions were identified by counting extreme responses.
    - Top questions for DR and TV2 were presented, shedding light on pivotal issues.
4. **Average Positions of Parties:**
    - Average positions of parties concerning selected questions were calculated and visualized.
5. **Age Analysis:**
    - The average age of candidates grouped by parties was explored for TV2.
6. **Confident Candidates Identification:**
    - Candidates with the highest proportion of "strongly agree" or "strongly disagree" responses were identified.
7. **Response Differences between Candidates:**
    - Internal disagreements within parties were analyzed, revealing parties with the most differences.
8. **Classification Models:**
    - Decision Tree, Random Forest, and Gradient Boosted Tree algorithms were utilized to predict party affiliations.
    - Model accuracies were evaluated.

9. **Clustering Analysis:**
    - K-Means, Hierarchical, and DBSCAN clustering techniques were employed to explore potential party clusters.
    - Optimal cluster numbers were discussed.
10. **Political Landscape Overview:**
    - The political landscape of elected candidates was examined, emphasizing members with significant agreement or disagreement.

Reflection: This assignment provided a holistic exploration of political data, requiring a diverse set of analytical techniques. The solution involved comprehensive data preprocessing, visualization, and modeling, showcasing the application of PCA, clustering, and classification algorithms. The analysis deepened understanding regarding the political landscape, candidate behaviors, and internal party dynamics. The iterative process of data exploration and algorithmic application enhanced skills in data analysis, interpretation, and communication of results.

Assignment 5

Assignment Problem:

The task is to perform sentiment analysis on the IMDb dataset using natural language processing techniques. The dataset, comprising movie reviews and corresponding sentiment labels, is initially loaded into a pandas dataframe.

Solution Approach:

Data Acquisition: Reviews and corresponding labels were loaded from 'reviews.txt' and 'labels.txt', respectively.

Data Splitting: The dataset was split into training, validation, and test sets using the train_test_split function from scikit-learn.

Text Representation: The reviews were transformed into Bag-of-Words (BoW) representations using a CountVectorizer with a maximum of 10,000 most frequent words.

Neural Network Training: A neural network with a single hidden layer was defined and trained using TensorFlow and Keras. The model was compiled with the Adam optimizer and binary crossentropy loss.

Performance Evaluation: The model's performance was evaluated on the test set, achieving a test accuracy of approximately 87.56%.

Algorithms Used:

CountVectorizer: Used for text representation, converting reviews into BoW representations.

Neural Network: Implemented with a single hidden layer and sigmoid activation for sentiment classification.

Performance:

The final model, a neural network trained on BoW representations, achieved a test accuracy of 87.56%, demonstrating its effectiveness in sentiment classification.

Reflection on Learning Outcome:

This assignment provided a hands-on experience in natural language processing and sentiment analysis. Key takeaways include data preprocessing, feature extraction, and the application of neural networks for text classification. The exploration of BoW representations and the iterative process of tuning

hyperparameters contributed to a deeper understanding of machine learning techniques. Additionally, the ability to interpret and visualize the impact of individual words on predictions enhanced insights into the workings of the model. Overall, the assignment facilitated practical learning and problem-solving in the context of sentiment analysis.

Assignment 6

Assignment Problem:

The assignment involves leveraging the Google Speech Command Dataset, a subset of which comprises 105,829 one-second audio files with utterances of four common words ("Yes," "No," "Stop," and "Go"). The goal is to perform speech recognition using machine learning techniques.

Solution:

(a) Data Preparation and Convolutional Neural Network (CNN) Training:

Loaded audio spectrograms and class labels from XSound.npy and YSound.npy, respectively.

Split the data into training, validation, and test sets.

Reshaped data to fit the CNN model.

Trained a CNN with hyperparameter tuning for optimal performance.

(b) CNN Performance Analysis:

Achieved a test accuracy of approximately 94.6%.

Utilized a confusion matrix and classification report for detailed model evaluation.

(c) Long Short-Term Memory (LSTM) Model Training:

Implemented an LSTM model, known for capturing temporal dependencies in sequential data.

Trained the LSTM model and evaluated its performance on the test set.

(d) LSTM Performance Analysis:

Obtained a test accuracy of around 95.2%.

Visualized the confusion matrix and generated a classification report for comprehensive evaluation.

Reflection on Learning Outcome:

Both CNNs and LSTMs were effective in audio classification, with CNNs capturing spatial patterns and LSTMs excelling in modeling temporal dependencies.

Considerations for model selection involve the nature of the audio data, the need for capturing long-term dependencies, and the dataset size.

LSTMs proved valuable for sequential audio data, while CNNs excelled in capturing local patterns in spectrograms.

The assignment provided practical insights into choosing appropriate models for audio classification tasks and understanding their strengths and limitations.

Assignment 7 (Final project)

Assignment 7 (Final group project)

Assignment Problem:

The ultimate objective of this final project is to develop a model capable of predicting flight delays and estimating the likelihood of delays based on departure and arrival times and airports. The comprehensive solution involves a multi-step process encompassing data collection, analysis, feature engineering, model training, algorithm selection, and performance evaluation.

Solution Steps:

1. Data Collection: Gather the necessary data for analysis.
2. Data Analysis: Examine the dataset, identifying correlations between columns.
3. Feature Engineering: Select relevant columns, handle missing values, and address outliers. Represent airports as binary values.
4. Data Splitting and Model Training: Divide processed data into training and test sets and proceed with model training.
5. Algorithm Evaluation: Explore various algorithms using methods like confusion matrices and AUC-ROC curves.
6. Model Refinement: Iteratively fine-tune the model for enhanced performance.
7. Predictive Testing: Utilize the trained model to make predictions about flight delays.

Algorithms Used:

1. Random Forest: Introduces randomness, preventing overfitting, with fast training speed and the ability to handle high-dimensional data. It accommodates both discrete and continuous data without requiring normalization.

2. Gradient Boosting Machine (GBM): Demonstrates high prediction accuracy, captures nonlinear relationships well, and is robust to noise and outliers. Combining multiple models reduces the risk of overfitting.

3. Neural Network: Possesses strong nonlinear fitting ability, excelling in capturing complex relationships. Through dynamic tracking and deep learning, it reflects prediction results realistically. Continuous iteration and structure adjustment ensure stability, eliminating human subjective factors for authentic and objective results.

Performance:

1. Random Forest: Swift with relatively high accuracy.
2. GBM: Faster with higher accuracy than Random Forest. Iterative refinement enhances comprehensiveness and accuracy.
3. Neural Network: Slower but exhibits relatively high accuracy. Adjustments further enhance performance.

Reflection on Learning Outcome:

This project, initiated from scratch and independently completed, amalgamates knowledge acquired throughout the learning journey. From data analysis to feature engineering, model training, and iterative refinement, the experience uncovered challenges, led to the discovery of intriguing insights, and enriched understanding. The process not only showcased the application of acquired skills but also facilitated continuous learning in problem-solving contexts.